Studies in Computational Intelligence 275

Dilip Kumar Pratihar
Lakhmi C. Jain (Eds.)

Intelligent Autonomous Systems

Foundations and Applications

Springer

Dilip Kumar Pratihar and Lakhmi C. Jain (Eds.)

Intelligent Autonomous Systems

Studies in Computational Intelligence, Volume 275

Editor-in-Chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 255. Catarina Silva and Bernardete Ribeiro
Inductive Inference for Large Scale Text Classification, 2009
ISBN 978-3-642-04532-5

Vol. 256. Patricia Melin, Janusz Kacprzyk, and Witold Pedrycz (Eds.)
Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition, 2009
ISBN 978-3-642-04515-8

Vol. 257. Oscar Castillo, Witold Pedrycz, and Janusz Kacprzyk (Eds.)
Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control, 2009
ISBN 978-3-642-04513-4

Vol. 258. Leonardo Franco, David A. Elizondo, and José M. Jerez (Eds.)
Constructive Neural Networks, 2009
ISBN 978-3-642-04511-0

Vol. 259. Kasthurirangan Gopalakrishnan, Halil Ceylan, and Nii O. Attoh-Okine (Eds.)
Intelligent and Soft Computing in Infrastructure Systems Engineering, 2009
ISBN 978-3-642-04585-1

Vol. 260. Edward Szczerbicki and Ngoc Thanh Nguyen (Eds.)
Smart Information and Knowledge Management, 2009
ISBN 978-3-642-04583-7

Vol. 261. Nadia Nedjah, Leandro dos Santos Coelho, and Luiza de Macedo de Mourelle (Eds.)
Multi-Objective Swarm Intelligent Systems, 2009
ISBN 978-3-642-05164-7

Vol. 262. Jacek Koronacki, Zbigniew W. Ras, Slawomir T. Wierzchon, and Janusz Kacprzyk (Eds.)
Advances in Machine Learning I, 2009
ISBN 978-3-642-05176-0

Vol. 263. Jacek Koronacki, Zbigniew W. Ras, Slawomir T. Wierzchon, and Janusz Kacprzyk (Eds.)
Advances in Machine Learning II, 2009
ISBN 978-3-642-05178-4

Vol. 264. Olivier Sigaud and Jan Peters (Eds.)
From Motor Learning to Interaction Learning in Robots, 2009
ISBN 978-3-642-05180-7

Vol. 265. Zbigniew W. Ras and Li-Shiang Tsay (Eds.)
Advances in Intelligent Information Systems, 2009
ISBN 978-3-642-05182-1

Vol. 266. Akitoshi Hanazawa, Tsutom Miki, and Keiichi Horio (Eds.)
Brain-Inspired Information Technology, 2009
ISBN 978-3-642-04024-5

Vol. 267. Ivan Zelinka, Sergej Celikovský, Hendrik Richter, and Guanrong Chen (Eds.)
Evolutionary Algorithms and Chaotic Systems, 2009
ISBN 978-3-642-10706-1

Vol. 268. Johann M.Ph. Schumann and Yan Liu (Eds.)
Applications of Neural Networks in High Assurance Systems, 2009
ISBN 978-3-642-10689-7

Vol. 269. Francisco Fernández de de Vega and Erick Cantú-Paz (Eds.)
Parallel and Distributed Computational Intelligence, 2009
ISBN 978-3-642-10674-3

Vol. 270. Zong Woo Geem
Recent Advances In Harmony Search Algorithm, 2009
ISBN 978-3-642-04316-1

Vol. 271. Janusz Kacprzyk, Frederick E. Petry, and Adnan Yazici (Eds.)
Uncertainty Approaches for Spatial Data Modeling and Processing, 2009
ISBN 978-3-642-10662-0

Vol. 272. Carlos A. Coello Coello, Clarisse Dhaenens, and Laetitia Jourdan (Eds.)
Advances in Multi-Objective Nature Inspired Computing, 2009
ISBN 978-3-642-11217-1

Vol. 273. Fatos Xhafa, Santi Caballé, Ajith Abraham, Thanasis Daradoumis, and Angel Alejandro Juan Perez (Eds.)
Computational Intelligence for Technology Enhanced Learning, 2010
ISBN 978-3-642-11223-2

Vol. 274. Zbigniew W. Raś and Alicja Wieczorkowska (Eds.)
Advances in Music Information Retrieval, 2010
ISBN 978-3-642-11673-5

Vol. 275. Dilip Kumar Pratihar and Lakhmi C. Jain (Eds.)
Intelligent Autonomous Systems, 2010
ISBN 978-3-642-11675-9

Dilip Kumar Pratihar and Lakhmi C. Jain (Eds.)

Intelligent Autonomous Systems

Foundations and Applications

Springer

Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology
Kharagpur
India

Prof. Lakhmi C. Jain
School of Electrical and
Information Engineering
University of South Australia
Adelaide
Mawson Lakes Campus
South Australia
Australia

ISBN 978-3-642-11675-9 e-ISBN 978-3-642-11676-6

DOI 10.1007/978-3-642-11676-6

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: 2009943992

Typeset & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Foreword

Intelligent Autonomous Systems (IAS) are the physical embodiment of machine intelligence providing a core concept for integrating various advanced technologies with pattern recognition and learning. The basic philosophy of IAS research is to explore and understand the nature of intelligence in problems of perception, reasoning, learning and control in order to develop and implement the theory to engineered realization. In other words, the objective is to formulate various methodologies for the development of robots which can operate autonomously and exhibit intelligent behavior by making appropriate decisions to perform the right task at the right time. Since IAS basically deals with the integration of machines, computing, sensing, and software to create intelligent systems capable of interacting with the complexities of the real world, advanced topics like soft computing, artificial life, evolutionary biology, and cognitive psychology have great promise in improving its intelligence and performance.

Because of the inter-disciplinary character, the subject has several challenging issues for research, design and development covering a number of disciplines. These issues are further concerned with the development of both technology and methodology apart from various operations.

The present research monograph titled “Intelligent Autonomous Systems: Foundations and Applications", edited by two renowned researchers, Professor Dilip K. Pratihar of IIT, Kharagpur, India and Professor Lakhmi C. Jain, University of South Australia, Australia, provides a fairly representative cross-section of the activities that is going on all over the world in this area. The volume comprising ten chapters begins with explaining the basic definition and elements of IAS, and then covers different problems in topics like adaptive motion planning, stair-climbing and obstacle avoidance, ensemble learning, and attack graph in designing a robot with various functions. The most salient feature is the use of soft computing tools for optimization, learning, reasoning and uncertainty analysis in such a design for real life applications. It is shown, as an example, how synergistic integration of the merits of genetic algorithms, neural networks and fuzzy logic can be leveraged for dynamically balanced biped robots moving on uneven terrains.

Besides, some general issues like security management with scalable representation of the attack Graphs and the development of intelligent tools with Hidden Markov Model are studied.

I believe the chapters would help in understanding not only the basic issues and characteristic features of Intelligent Autonomous Systems, but also the application of different technologies (classical and modern) being used in this area so that it can be extended to other domains. Professors Pratihar and Jain deserve congratulations for bringing out the nice piece of comprehensive collection.

Professor Dr. Sankar K. Pal
Fellow IEEE
Director
Indian Statistical Institute
Kolkata
India

Preface

If history is a useful guide, we can state that humans have always used machines to reduce the hard labour otherwise required to provide society's needs. Since the middle of last century, researchers have been trying to utilise intelligence in the machines. The recent advances in artificial intelligence and the availability of cheap computing power are at the heart of the present day success of intelligent autonomous systems.

To fulfil the increasing demands of systems in a highly dynamic world, today's systems need to be intelligent and autonomous. An intelligent autonomous system can take its decisions independently and as the situation requires in an ever changing environment. In order to develop such a system, the principle of artificial, computational intelligence needs to be merged with the original system. It is a difficult task and to achieve it, training must be provided to the system. This may be either off-line or on-line in order to build a suitable knowledge-base. A proper combination of off-line and on-line training can also yield an efficient and intelligent autonomous system.

It is of interest to observe today's machines and engineering systems coming to the market with built in knowledge bases. Although these systems are already in operation, there is probability of further improvement in performance through more advanced training. A large number of training tools have been developed by researchers. Besides the traditional tools, biologically-inspired and knowledge-based tools have proved their effectiveness in the training and learning processes. In the near future, a large number of intelligent autonomous systems will evolve using the principle of soft computing. This book has ten chapters and the Appendix. It deals with the principles and the applications of intelligent autonomous systems.

We thank the authors and reviewers for their contributions. Thanks are also due to the editorial team of the Springer-Verlag Company for their helpful assistance during the book's preparation.

D.K. Pratihar, India
L.C. Jain, Australia

Contents

Chapter 1

Towards Intelligent Autonomous Systems 1
Dilip Kumar Pratihar, Lakhmi C. Jain

Chapter 2

General Aspects of Intelligent Autonomous Systems 5
Wolfgang Bibel

Chapter 3

Design and Development of Intelligent Autonomous Robots 29
Nirmal Baran Hui, Dilip Kumar Pratihar

Chapter 4

Gait Planning of Biped Robots Using Soft Computing: An Attempt to Incorporate Intelligence 57
Pandu Ranga Vundavilli, Dilip Kumar Pratihar

Chapter 5

On Design and Development of an Intelligent Mobile Robotic Vehicle for Stair-Case Navigation 87
Ranjit Ray, Bikash Bepari, Subhasis Bhaumik

Chapter 6

Ensemble Learning for Multi-source Information Fusion 123
Jörg Beyer, Kai Heesche, Werner Hauptmann, Clemens Otte, Rudolf Kruse

Chapter 7

Towards Developing Intelligent Autonomous Systems in Psychiatry: Its Present State and Future Possibilities 143
Subhagata Chattopadhyay, Dilip Kumar Pratihar

Chapter 8

Condition Monitoring of Internal Combustion Engine Using EMD and HMM 167
Sandeep Kumar Yadav, Prem Kumar Kalra

Chapter 9

An Intelligent Approach for Security Management of an Enterprise Network Using Planner 187
Nirnay Ghosh, S.K. Ghosh

Chapter 10

High Dimensional Neural Networks and Applications 215
B.K. Tripathi, P.K. Kalra

Author Index 235

Appendix 237

1

Towards Intelligent Autonomous Systems

Dilip Kumar Pratihar and Lakhmi C. Jain

Abstract. A great deal of research effort is being expended at present to develop intelligent autonomous systems. A brief overview of intelligent autonomous system is given in this introductory chapter. A summary of each chapter of this book is also included. An introduction to both the theoretical foundations and some applications of various intelligent autonomous systems are included.

1 Introduction

Intelligence is not an easy idea to define. Intelligent system should ideally perform "like a human" in a limited sense. Intelligent system should to some extent be able to learn to perform given tasks and make decisions. The term: intelligence has been defined in various ways. A popular definition of intelligence is given by Albus [1]. This is as follows: "… intelligence is defined as an ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral sub-goals that supports the system's ultimate goal." In an engineering system, intelligence is provided in the form of Artificial Intelligence. This is in order to learn and to reason.

A system is called autonomous, when it can independently perform the assigned task without continuous human guidance. It is important to note that all intelligent systems may not be autonomous.

An intelligent autonomous system will be capable of making decisions independently on-line, depending on the demands under varying conditions. It is not an easy task to design and develop such a system. It involves the following qualities:

1. The definition and representation of intelligence;
2. The incorporation of intelligence into a system;
3. The testing and validation of the system's performance.

Dilip Kumar Pratihar
Indian Institute of Technology Kharagpur,
India

Lakhmi C. Jain
University of South Australia
South Australia
Australia

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 1–4.
springerlink.com

A number of conventional and intelligent techniques which include Knowledge-based Systems [2-3], Fuzzy Systems [4-6], Evolutionary Computing [7], Neural Systems [8] and Intelligent Agents [9]. In addition, the fusion of these paradigms [10] has been reported in order to fuse the intelligence into the machines. The researchers have reported that the application of these paradigms into areas, such as Healthcare [11-14], Decision Making [15-16], Education [17-23] for example.

This book deals with the principle of Intelligent Autonomous Systems and with their applications in several fields.

2 Chapters Included in the Book

This book has ten chapters including the present chapter called *"Towards Intelligent Autonomous Systems"*. We also included an Appendix, which lists various resources for the use of readers.

Chapter 2 starts with the definition of Intelligent Autonomous Systems or *agents.* Motivations behind design and development of agents are outlined here. It illustrates the state of the art of this field of research both conceptually and includes representative examples. This chapter outlines the future scope of research and also notes five challenges in this field.

Chapter 3 outlines the issues related to the design and development of intelligent autonomous robots. One of the most important concerns when developing an intelligent autonomous robot is the provision of an adaptive motion planning scheme. Conventional Potential Field Methods and Soft Computing-Based Approaches have been developed, and their performances are studied by means of Computer Simulations and by Practical Experiments.

In Chapter 4, adaptive gait planners are developed for dynamically balanced biped robots when they are moving on uneven terrains. These systems are based on Neural Network- and Fuzzy Logic-Based approaches. The knowledge bases used in the two approaches are optimized using an off-line genetic algorithm. The performances of the approaches developed have been tested using computer simulations.

Chapter 5 focuses on the study of several features, such as stair-climbing mechanisms and autonomously transporting materials. Issues related to design and development of intelligent robotic vehicle for stair- navigation have been discussed. The following functional attributes of the robot are investigated: (i) The motion on plane surface; (ii) The navigation on a staircase; (iii) The autonomous navigation and control including features such as obstacle avoidance and stair-detection are included. A vision-assisted system has been developed for stair-case detection and a sonar data-based two layered fuzzy logic technique has been incorporated for obstacle avoidance. The performance of the system developed has been tested by its use on a real robot.

Chapter 6 deals with a new ensemble learning method. The aim here is to jointly use Data-Driven and Knowledge-Based Sub-models using this combined approach, the advantages of both the models, which are the Robustness and Physical Transparency of the Knowledge-Based Model, and the approximation skills of

Data-Driven learning, have been utilized. The performance when using this approach has been tested by the use on a real-world problem.

In Chapter 7, the issues related to the development of an intelligent autonomous system in psychiatry are considered. It develops a meta-analysis of current researches on the application of various knowledge engineering techniques and intelligent decision support systems in psychiatry. In addition to providing a critical review of the present state of research, issues related to the design, development and the practical implementation of intelligent autonomous systems in day-to-day psychiatry practice are considered.

Chapter 8 provides an acoustics-based condition monitoring system for an Internal Combustion (IC) engine using both Empirical Mode Decomposition (EMD) and Hidden Markov Model (HMM). EMD is a new time-frequency method of analysis for nonlinear and non-stationary signals. The aim is to decompose a complicated signal into a number of Intrinsic Mode Functions (IMFs). Considering these IMFs as feature vectors, HMM is then used to classify the IC engine acoustic signals. Experimental results support the use of the proposed method as a useful tool for all intelligent autonomous systems to mention the conditions and to provide fault diagnosis of an IC engine.

Chapter 9 proposes the use of an *attack graph*. This is to provide intelligent approach for security management of a computer network. An attack graph is utilized to correlate the multi-stage, multi-host attacks used for representing various attack scenarios. This chapter considers a novel approach called *planner*. This has been developed using an artificial intelligence technique. It provides scalable representation of the attack graphs. The performance of the approach developed has been tested using a case study.

Chapter 10 explains the principle of high dimensional neural networks when used as the candidate tools for design and development of intelligent systems. An overview of the existing methodologies in high dimensional neural computation is given. This chapter deals with the applications of complex domain neural network-based intelligent systems in a number of real-world problems.

3 Conclusion

This chapter introduces to the reader intelligent autonomous systems. The book deals with the principles of intelligent autonomous systems and their application in various fields. This includes Robotics, Medical Diagnosis, monitoring the health of IC engines, Computer Network Security, Face Detection for example. An important concern of society today is the safety and the living environment of humans. In order to satisfy these concerns in a highly dynamic environment, systems need to be autonomous and to be intelligent. An attempt has been made in this book to meet the need and the principles of intelligent autonomous systems. Some applications of intelligent autonomous systems when used in various fields have been outlined. However, much research is yet to be done before all requirements are satisfied.

References

[1] Albus, J.: Outline for a Theory of Intelligence. IEEE Trans. on Systems, Man and Cybernetics 21(3), 473–509 (1991)
[2] Jain, L.C. (ed.): Soft Computing Techniques in Knowledge-Based Intelligent Engineering Systems. Springer, Germany (1997)
[3] Jain, L.C., Johnson, R.P., Takefuji, Y., Zadeh, L.A. (eds.): Knowledge-Based Intelligent Techniques in Industry. CRC Press, USA (1998)
[4] Dumitrescu, D., Lazzerini, B., Jain, L.C.: Fuzzy Sets and their Applications to Clustering and Training. CRC Press, USA (2000)
[5] Sato, M., Sato, Y., Jain, L.C.: Fuzzy Clustering Models and Applications. Springer, Germany (1997)
[6] Sato, M., Jain, L.C.: Innovations in Fuzzy Clustering. Springer, Heidelberg (2006)
[7] Dumitrescu, D., Lazzerini, B., Jain, L.C., Dumitrescu, A.: Evolutionary Computing and Applications. CRC Press, USA (2000)
[8] Jain, L.C., Vemuri, R. (eds.): Industrial Applications of Neural Networks (1998) ISBN 0-8493-9802-9
[9] Resconi, G., Jain, L.C.: Intelligent Agents: Theory and Applications. Springer, Heidelberg (2004)
[10] Jain, L.C., Jain, R.K. (eds.): Hybrid Intelligent Engineering Systems. World Scientific Publishing Company, Singapore (1997)
[11] Sordo, M., Vaidya, S., Jain, L.C. (eds.): Advanced Computational Intelligence Paradigms in Healthcare, vol. 3. Springer, Heidelberg (2008)
[12] Yoshida, H., Jain, A., Ichalkaranje, A., Jain, L.C. (eds.): Advanced Computational Intelligence Paradigms in Healthcare, vol. 1. Springer, Heidelberg (2007)
[13] Teodorescu, H.N., Kandel, A., Jain, L.C. (eds.): Fuzzy and Neuro-fuzzy Systems in Medicine. CRC Press, USA (1998)
[14] Schmitt, M., Teodorescu, H.N., Jain, A., Jain, A., Jain, S., Jain, L.C. (eds.): Computational Intelligence Processing in Medical Diagnosis. Springer, Heidelberg (2002)
[15] Phillips-Wren, G., Ichalkaranje, N., Jain, L.C. (eds.): Intelligent Decision Making-An AI-Based Approach. Springer, Heidelberg (2008)
[16] Phillips-Wren, G., Jain, L.C. (eds.): Intelligent Decision Support Systems in Agent-Mediated Environments. IOS Press, Amsterdam (2005)
[17] Jain, L.C., Tedman, R., Tedman, D. (eds.): Evolution of Teaching and Learning in Intelligent Environment. Springer, Heidelberg (2007)
[18] Ghaoui, C., Bannore, V., Jain, M., Jain, L.C. (eds.): Knowledge-Based Virtual Education. Springer, Heidelberg (2005)
[19] Jain, L.C., Howlett, R.J., Ichalkaranje, N., Tonfoni, G. (eds.): Virtual Environments for Teaching and Learning. World Scientific Publishing Company, Singapore (2002)
[20] Jain, L.C. (ed.): Innovative Teaching and Learning. Springer, Germany (2000)
[21] Jain, L.C., Garud, G.N.: CAD and CAL Tools for Electrical Engineering Students. Journal of the Institution of Engineers 75, 52–54 (1994)
[22] Jain, L.C., Bowden, B.S.: Development of Expert System Course for Electronic Students. International Journal of Electrical Engineering Education 31(1), 34–45 (1994)
[23] Rowland, J.G., Jain, L.C.: Artificial Intelligence Languages in Engineering Education. International Journal of Electrical Engineering Education 31(2), 138–143 (1994)

2

General Aspects of Intelligent Autonomous Systems

Wolfgang Bibel

Abstract. The chapter motivates the study of intelligent autonomous systems (or agents) and illustrates the state of the art in this area both conceptually as well as with representative examples (like Stanley, the vehicle robot). It outlines promising areas of research that will not only lead to more intelligent systems in the near future, but also sets up five challenges, which in the long run would result in truly intelligent assistants and collaborators for human beings to help overcoming the enormous problems of this world.

1 Introduction

The industrial revolution transformed the society from a manual-labour-based economy towards machine-based manufacturing, starting in the late eighteenth century. It is characterized by the processes of mechanisation, which took hold in agriculture, production, and transportation. Later, information technology (IT) has added the feature of automation to the mechanisms which are employed in a great variety. In this way, the mechanisms have become smoother, easier to control and handle by human beings. They also have become more complex by several orders of magnitude, however.

Human beings, by nature, are not especially capable for coping with mechanisms of such a high complexity. Like other animals, they have evolved to collaborate with agents, which although extremely complex when seen as a mechanism, are relatively easier to communicate with because they behave in an autonomous and familiar way. In other words, collaboration with familiar autonomous systems by nature is far easier for the human beings than controlling complex automated systems. This is one of the deeper reasons why we currently can observe a new *trend from automation towards autonomy*.

Wolfgang Bibel
Darmstadt University of Technology, Darmstadt, Germany.
Also affiliated with the University of British Columbia
e-mail: Bibel@gmx.net

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 5–27.
springerlink.com

Examples of this trend are numerous. Clocks switch from winter to summer time autonomously. Subway systems in some cities (Tokyo, Nürnberg, etc.) are running without any human drivers and the systems are apparently working better without human beings in the loop. Autonomous logistic cargo systems are installed in harbors like that of Hamburg, Germany. Cars remind their owners of necessary maintenance routines. Wheel chairs are autonomously avoiding hazardous drives and obstacles. Many systems in the International Space Station or in other space objects are doing their work autonomously. Especially, the internet is full of autonomous agents carrying out numerous tasks, such as routing messages, downloading patches, datamining web-pages, infecting computers by viruses, and so forth. In the envisaged *internet of things*, the trend will increase even much further.

The above few examples demonstrate the wide variety of autonomous systems already in use. A broad article on intelligent autonomous systems can, therefore, address only very general issues concerning this field of research and its technology. The present contribution will, therefore, not go into any detail of specific autonomous systems and the techniques involved in their development; rather we focus on general aspects and trends in this area in order to provide with a broader view of the subject.

The chapter starts in the next section with some conceptual preliminaries and a more detailed discussion on the motivations behind intelligent autonomous systems. We also justify, why we prefer to speak of intelligent autonomous agents (IAAs) rather than of systems. In Section 3, we give a brief overview of the state of the art in IAA technology and briefly describe Stanley, the race-winning autonomous vehicle, as a representative for this state. Thereby, we distinguish different types of IAAs and describe which of these characterize the current state of affairs. Section 4 deals with a brief outline of technology in this field, which is currently in the state of development and will advance the capabilities of IAAs in the near future. Section 5 outlines the author's long-range technological perspectives of IAAs. In particular, we present five challenges which we consider crucial for advancing the field in the future. Thereby, we once again remind of the promising logical approach to a future IAA technology. Section 6 presents some ideas of how to design and develop IAAs and where to install the features materializing their intelligence. The final section ends the chapter with some conclusions.

Throughout the entire text, we lay great emphasis on the close connections between artificial IAAs on the one side and of natural agents, such as animals and human beings on the other. Developing the former and studying the latter has to go hand in hand in order to achieve the grand goal of truly intelligent autonomous agents.

2 Preliminaries and Motivations

In the present section, we start with clarifying the concepts involved in the area of intelligent autonomous systems and provide with some motivations for developing them.

2.1 Agents

The term *system* is used in this chapter in a very general sense comprising programs, machines, agents in an environment, organizations, and so forth. Since, we are focusing on intelligent systems in this chapter, we prefer speaking of *agents* to systems in our context. To a large extent, these two terms are synonymous, although intelligence is more often attributed to agents than to systems.

The characteristic of an agent is its ability to act within an environment, which may comprise the agent's body. Typically, an agent acts after receiving some inputs. This may be simple program inputs or, more generally, perceptions gathered through sensors from the environment. Similarly, actions may just consist of program output or, more generally again, in changes imposed on the environment through the agent's actuators.

The action of agents typically is related to the agent's state as well as to the perceptions made. This relation, sometimes termed decision in the following, is realized in our understanding of agents through both their physical and functional designs as well as computational means.

2.2 Autonomy and Intelligence

Biological agents, such as polar bears or birds, but also trees, are born into an environment and endowed by nature with a variety of capacities. In the case of animals of the illustrated kind, they are nurtured by their parents for some time in which they learn from their own experiences and are taught how to use these capacities for survival. After the period of nursery, they are released to become autonomous agents in their environment. Autonomy, in this sense, is the ability to master on its own the tasks securing survival within a given environment.[1]

The technical term of *autonomy* in the case of artificial agents is coined in analogy with the sense just illustrated; it refers to the agent's ability of mastering a certain set of tasks within a given environment and without recourse to the agent's designer in order to achieve certain goals or desires.

It is a matter of the particular nature of the environment involved whether or not agents need intelligence to master their tasks. There is of course a vast variety of different task environments. To structure this complex and huge space, the book [24, p.40ff] has distinguished the following six different dimensions of environments: *fully* vs. *partially observable*, *deterministic* vs. *stochastic*, *episodic* vs. *sequential*, *static* vs. *dynamic*, *discrete* vs. *continuous*, *single* vs. *multi-agent*.

We refer to the mentioned book for details of these dimensions. Here, we just add a couple of clarifying comments for those dimensions, which might not be immediately understandable to the reader. Observability is determined by the performance of the agent's sensors in regard to the nature of the environment. An episodic task environment allows the division of the agent's experiences into atomic episodes, so

[1] This description of autonomy includes the case of human beings, whose autonomous behavior is nicely modeled in the book [14].

that a later episode becomes completely independent of the perceptions and actions (or decisions) in an earlier one. In contrast to that, in a sequential task environment the choice of an action could affect all future decisions.

Environments feature other distinguishing characteristics than those used in these six dimensions. For instance, environments might be either *digital* or *physical*; agents in a digital environment are also called *software agents*. This characteristic hence defines a seventh dimension, which is useful in our context. In other words, the structure defined by these dimensions is neither unique nor exhaustive. But, it helps to focus the discussion on important aspects of the environments.

The environment is an essential aspect in understanding and designing autonomous agents. Also in nature, agents are intrinsically related to a special kind of environment (their habitat). For instance, some species of birds and one kind of polar bears are endangered by the current climatic change, because they lack the intelligence to change their habits fast enough to adapt to the ensuing rapid changes of their habitats, especially the consequences in the conditions of food supply. In other words, any species of agents, whether biological or artificial, is situated in and fitted to a particular range of environments only. Autonomy is never an absolute one, but one relative to the envisioned environment. Designing an agent, thus, requires as well a clear conception of its *habitat*, i.e., the range of environments in which the agent is meant to operate.

Intelligence in a rather broad understanding "is that quality that enables an entity to function appropriately and with foresight in its environment" [21, Preface]. A higher degree of intelligence allows the agent to adapt to a wider range of environments. This is why, humanity has conquered and subdued the world and with it all other species. This particular power explains our interest in implanting as much intelligence into agents as possible. However, also for intelligent agents, there are limits to this ability for adaptation to changes in their environment. This is because of the fact that any non-digital agent is embedded into its environment by physical constraints. Overcoming these constraints would require a redesign of the agent rather than being achievable by more intelligence.

2.3 Motivation for Intelligent Autonomous Agents

The interest in IAAs stems from two main motivations, namely (i) achieving a better understanding of human beings (and other animals) and (ii) improving current technology.

The first motivation aims at explaining the behavior of animals upto the higher human cognitive activities, such as complex problem solving, planning, decision making, attention, and so forth upto consciousness. It has of course a tradition of already more than two thousand years, in which philosophers, psychologists and others tried to discover the principles which govern human beings, the prime examples of IAAs. It is only since the last century that these endeavours have started to be pursued on a solid scientific basis rather than on the shaky grounds of philosophical speculation. Especially, the computational paradigm introduced by the disciplines of Artificial Intelligence and Cognitive Science, jointly also termed Intellectics by the author [4], has laid a solid methodological basis for research into this direction, which is scientific in the strict sense of the natural sciences (like physics, biology etc.).

The second motivation, namely improving current technology through IAAs, has become even more important as systems penetrating and governing our societies have become ever more complex, as already pointed out in the previous introductory section.

Let us illustrate this growing need for IAAs with the omnipresent personal computers (PCs). Computer scientists have designed these machines in a functional way, where the functions are implemented within the hardware or by software in an algorithmic way. To cope with the growing complexity of the functionality of PCs, the level of functional abstraction has been raised over the decades as far as one could go (e.g., by way of designing high-level programming languages). However, this methodology could not solve the problems of the casual users in dealing with this growing complexity. To solve certain tasks (like installing some nonstandard software), the user has to plunge into the lower-level functional labyrinth of the system which most users are simply unable to do. This inacceptable situation generated the desire for IAAs, e.g., in the form of intelligent autonomous installation assistants which can carry out such a task without any technical guidance from the user while respecting his personal requests, illustrating again the trend from automation to autonomy noted in the Introduction.

What we just illustrated for the popular case of PCs is a general phenomenon, which can be experienced in all facettes of our societies. To mention a couple of other examples, recall first the worldwide financial crisis of the year 2008; the financial system has become so complex that nearly no one was able to realize the dangers involved with certain financial products (a fact which some contemporaries have exploited to their big advantage); or think of the entire legal system, or of the tax system, as just a part of it, which (at least in Germany, the worldmaster in producing tax regulations) has become so complex that it hosts a number of lawyers, tax advisors and public servants [6]. It is, thus, natural to hope for an IAA in the form of a future intelligent autonomous personal tax advisor in order to cope with this growing burden and hopefully reduce the opportunities for unscrupulous and awkward customers.

Another area of potential applications of IAAs is the control of parts of our environment. Just think of your house and the functions of its various devices (electricity, heating, sewage and many more), which need to be controlled or maintained at certain intervals. The same applies to the factories and facilities of companies, organizations, and even states (weather stations, satellites, traffic control stations, monitoring of the water quality of rivers, lakes and the sea, and so forth). Last but not the least, industry has a vital interest in integrating IAAs in their manufacturing, production and distribution processes to reduce labor costs and improve product quality.

These are just a few examples out of a great many more possible and useful applications of IAAs. Because of this huge potential of using IAAs in a great variety of applications, they have become a hot topic of great importance in current research.

Of course, we should not expect IAAs to bring heaven on the earth, rather they certainly bring along with them also a number of risks. For instance, while they help to cope with system complexity, their use will actually increase the degree of complexity in this world. Let us think of the PC installation assistants mentioned above; if these do not work as expected and desired, it might be even more complicated to get their behavior changed than in the case of standard computer systems without IAAs involved. Because of this fact, it is absolutely necessary that IAAs are designed in a responsible and technically flexible manner, which allows them to be controlled by a relatively easier human interaction. This, in turn, calls for a better understanding of what *easy human interaction*, in fact, means, which is nothing else than saying that we need a better understanding of human beings, the first of our two general motivations behind carrying out research in IAAs at the outset of the present subsection.

Other problems with IAAs are related to ethical, legal, political, and psychological issues to name some of them. For instance, who should be responsible if an IAA would cause damage to some person or property? The introduction of IAAs in practice must, therefore, be accompanied by transparent legal and political regulations.

3 Representative Examples of the State of the Art

So far we have viewed IAAs as agents, which pursue certain goals or desires in a certain environment by solving appropriate tasks. We now want to go a step deeper into details by considering important functions in IAAs' design. To cover the most general case we think in this section of an IAA as some physical entity acting in a physical environment. Let us name the entity *Flocke*. The reader might think of Flocke as some future version of Honda's robot: ASIMO or, in stark contrast, of some human beings. Digital IAAs are regarded as a particularly important, but nevertheless special case of this general setting due to which we will not treat it separately.

Flocke is designed or born with certain built-in goals, such as fulfilling the needs of a PC user or surviving in some environment. All functions of Flocke will team up in a possibly complex way to achieve the set goals. For an easy understanding, we imagine, however, that these functions are organised in a hierarchical manner. At the lowest level, we think of the functions which keep the components of Flocke going as needed; at the highest level (if existent in Flocke) we might have reasoning, planning, and so forth; at intermediate levels there are the functions which link those at the extreme levels, e.g., process the sensor data to become useful at the highest level.

The *components* of Flocke may involve interaction devices like wheels or legs, grippers and other manipulators or limbs, driving devices like motors, gears, belts, or muscles, energy sources like batteries or some complex digestion apparatus, sensor devices like cameras or eyes, etc., and last but not the least some (possibly complex) information processing devices or a (central and peripheral) nervous system. The set of components of Flocke along with its structure is called *architecture* of the agent. The *agent* itself is run by some kind of *program* realised on this architecture, so that the agent is characterized by its architecture along with its program.

Let us now describe, how for various types of agents their programs will activate their functions at the various levels. Following [24, p.46ff], we will, thereby, consider the agent types of simple reflex agents, model-based reflex agents, goal-based agents, and utility-based agents, and use these for structuring the subsequent text.

3.1 Simple Reflex Agents

We begin with the *simple reflex agents*. An agent of this particular kind selects its current action on the basis of its current state, which, in turn, is determined exclusively by the previous state and the percepts just received. In other words, the percepts functionally determine the set of actions to be taken in a direct way, so that Flocke will respond automatically with an immediate reflex to them. Biologists speak of *fixed action patterns* in this context. The feedback controller, well-known in engineering applications, is a simple reflex agent of this kind.

The underlying function may be realized in various ways, e.g., by way of a set of condition-action rules, of an (augmented) finite state machine (or automaton), or by way of some analogue device. It may easily be hardwired in order to speed up the reaction time. In fact, the earlies simple reflex agents were realized in the form of some mechanical devices. Think of the regulation of the water level in a modern flush toilet as an example, which uses a principle invented in 270 B.C.E. by a Greek inventor and barber, Ktesibios of Alexandria [21, Sect.2.3.1]. In the case of a finite state machine, this may consist of several layers in such a way that lower layers never rely on the existence of higher layers (a structure known as *subsumption architecture*). The result in any case is an agent, which interfaces directly to the world through perception and action without internal changes other than state transitions.

Simple reflex agents, although they feature a rather limited intelligence, are nevertheless surprisingly powerful. Especially, they are extremely robust because of the simplicity of their (sometimes feedback-driven) model-free controller and their architecture. In nature, insects and other biological organisms operate to a large extent in such a way that sometimes they are stupidly reacting to some percept (e.g., think of flies attracted by a hot light source within darkness). But, in fact a substantial part of the behavior of all animals (including human beings) is controlled in such a way. The behavior of an IAA resulting from the interplay of such a simple controller and an environment (possibly complex) is often referred to as *emergent behavior*.

It was Rodney Brooks, who in the mid-1980s emphasized the importance of such a circuit-based approach in realizing autonomous agents (see eg. [9]). He went too far, however, when he argued that *good oldfashioned AI* (or GOFAI) has foundered on the issue of representation, as we will see in Section 5.2. Nevertheless, reactive robot control in the sense of Brooks (and of Braitenberg [8] with a similar approach) has become an important part of the IAA technology. An example of a robot built along these lines is the influential hexapod robot by Brooks named Genghis [24, Fig.25.22], which features six legs and is able to walk through rough terrain, reminding of the walking behavior of an ant.

3.2 Model-Based Reflex Agents

Simple reflex agents lack any explicit model of the world around them, which is their main distinction in comparison with *model-based reflex agents*. Representing and using such a model potentially improves the flexibility and intelligence of an agent considerably. Flocke, if realized as a model-based agent, would be able to interpret new percepts in the light of its stored model of its own state and of the environment and for instance, react to surprises in more intelligent ways (than do *stupid* flies). Most importantly, the model may be adapted, as Flocke gathers experiences about its environment over its lifetime and this way improve its reactions to sensor data [13].

Of course, one could argue that also a simple reflex agent carries some model of the world somehow coded within its reflex function; or the other way around, that a model-based reflex agent might with appropriate technology be realized as well as a simple reflex agent. In other words, these two types of agents cannot easily be distinguished on functional grounds. In the practice of building IAAs, however, the distinction is very helpful and commonly used.

Among the kinds of such models, we mention the following ones. *Forward models* enable an agent to predict the effects of its actions on its own motor system and its environment. Conversely, *inverse models* are used to deduce and output the necessary motor commands to achieve or maintain a desired goal. Both these internal models have been hypothesised to be used in human motor control [27]. Further models, such as maps will be mentioned in passing in the following.

The uncertainty in prediction of the effects of actions can be represented in various ways. Some agents employ fuzzy systems in this context (see eg. [23]). Others are based on probability theory. For instance, a popular approach consists in using a Bayesian network [24, p.492ff]. More generally, probabilistic robotics has become an extremely successful methodology [25] which was spectacularly demonstrated by the autonomous vehicle: Stanley winning the DARPA Grand Challenge in 2005 [26]. Since Stanley nicely demonstrates the power and flexibility of model-based reflex agents, we will describe it in some more detail here.

The challenge consisted in traversing unrehearsed off-road terrain. More specifically, it required the 23 participating vehicles to autonomously navigate a 132-mile long course through the Mojave desert in Nevada, USA, for which Stanley needed a little less than seven hours (precisely 6:53:58). The route was defined by some 3000 waypoints (precisely 2935 – in a format containing longitudes, latitudes, corridor widths, along with associated speed limits) and handed over shortly before the race on a CD-ROM.

Stanley features among its sensors five laser range finders, a color camera, two RADAR sensors, seven antenna (GPS etc.), wheel velocity, inertial measurement units, accelerometers, and gyroscopes for measuring orientation, and of course the usual car devices, such as throttle, break and steering. From the route definition and the data received through its sensors, Stanley estimates its current state represented by a total of 15 variables, assesses the surface condition and builds several maps (laser, vision, radar), which model the environment state. From all this information, special algorithms compute the input for steering, throttle and brake control which is translated into appropriate actions of the vehicle devices. In this way, the system could achieve to drive safely close to the center of the route at a relatively high speed.

The challenge in this task was to cope with problems, such as GPS outages, slipping, skidding or fishtailing of the vehicle, errors in pose estimation, avoiding obstacles ahead, recognizing phantom obstacles (like shades), tolerating varying lighting conditions, dust on the lens of the camera, etc., choosing terrain-adequate velocities, smoothing the trajectories, and so forth. Probabilistic methods played a central role in meeting this challenge (e.g., for obstacle detection). Similarly, important for the success of the project was the intensive use of learning algorithms (for parameter optimization, learning Gaussians, speed control, etc.) before and during the race. Of course, the project could and did also build on a long history of research on road finding (see eg. [12]). "From a broad perspective, Stanley's software mirrors common methodology in autonomous vehicle control. However, many of the individual modules relied on state-of-the-art artificial intelligence techniques" [26, p.691].

While Stanley is certainly a great success representating the frontier in IAA technology, it is still far from the intelligence exhibited by animals, such as lizards, birds, let alone mammals. Nevertheless, the technology used in Stanley and similar systems can be, and is already, employed in a great variety of useful applications. In this sense, it represents the state of the art. In the following two sections, we discuss other types of agents, which go beyond the two types discussed so far, and we will mention some of the challenges lying ahead in this respect.

4 IAA Technology in Progress

If we compare Stanley with living organisms and their capabilities, we find a number of striking differences. First of all, the mechanical parts are rather different in terms of function, design and material from analogue parts in animals. Unfortunately, we have not invented artificial muscles yet, but rely on materials like steel or aluminium. We will not go into any details of these mechanical aspects except for pointing out their importance and adding the following remark on robot dynamics.

Bipedal locomotion, as exhibited by humanoids, offers a great advantage in terms of mobility in a great variety of terrains. Honda's ASIMO is a step into this direction for technology demonstration. But, one would wish to design a walking IAA in a way, which utilizes the robot dynamics in order to minimize energy consumption. This has led to the idea of passive dynamic walking, which was realized in a number of very energy-efficient walking machines [11]. Analoguous research efforts are invested into other types of actuators, such as those for dexterous actuation.

Secondly, sensor technology will have to go a long way until it matches animal capabilities. Especially proprioception (measuring the robot's own state) seems still too crude as compared with animals. Again, we cannot go into any details in this direction.

Thirdly, human-robot interaction again is a wide field including modalities like speech, eye, gaze, facial expressions, gestures, body language, etc. Since we as human beings are used to these communicative expressions, it would be comfortable to have them at our disposal also in communications with IAAs [3]. Again, we will not go into any details of this active area.

Instead, we will now focus on aspects which enhance the autonomy and intelligence of IAAs based on their internal information processing capabilities. As a first observation, we note that a robot like Stanley is autonomous only locally. Once it is started and equipped with the necessary input data then it does what it is expected to do but it has not really any idea of its own goals. This distinguishes it from *goal-based* and *utility-based agents*. These two more advanced types of IAAs feature an explicit representation of their overall goals on the one hand and of a measure for comparing different states in terms of their utility for the agent on the other. Once these representations are at the disposal of IAAs, they can start with deliberations *before acting*, the most important feature which distinguishes human beings from most of the animals.

Artificial Intelligence (AI)[2] is the discipline, which among other topics has studied the nature of such deliberations and has developed a great variety of methods to perform such deliberations in a computational way. Among those are methods for problem solving, especially through search, constraint satisfaction, genetic programming, etc., knowledge representation and reasoning, planning, dealing with uncertainty, decision making, learning, collaboration, negotiation, coordination, conflict handling, social behavior, and many others. Of course, this is not the place to describe any of these methods in details, since this can be found in textbooks such as [24]. We rather discuss the potential lying in these *deliberative* methods for the future capabilities of IAAs and the challenges faced before realizing this potential. In the present section, our emphasis is on the short and medium range possibilities.

4.1 AI Methods in Store for IAAs

In order to demonstrate that these AI possibilities are real rather than fiction, we begin with shortly describing an already existing technology. "On October 24, 1998 NASA launched Deep Space 1 (DS1), a spacecraft whose mission was to evaluate the space-worthiness of twelve advanced technologies. One of these technologies was Remote Agent (RA), a robotic system for planning and executing spacecraft actions.... one of the most unique characteristics of RA, and a main difference with traditional spacecraft commanding, is that ground operators can communicate with

[2] The discipline is also named Computational Intelligence (CI). Some associate with CI the part of AI which focuses on soft programming techniques such as fuzzy logic or evolutionary programming.

RA using goals (e.g., *During the next week take pictures of the following asteroids and thrust 90% of the time*) rather than with detailed sequences of timed commands. RA determines a plan of action that achieves those goals and carries out that plan by issuing commands to the spacecraft" [21, Sect.32.2.1]. In other words, goal-based agents do exist already also in IAAs and do useful work in situations, in which it could not easily be carried out by human beings.

An IAA, such as Stanley would gain considerably if it featured more knowledge about its environment. For instance, if it would have some knowledge about the light coming from the sun and the shades cast through trees at the side of the route, it would have avoided its stupid and risky manœuvres around such shades which endangered its success during the race. Stanley was simply lucky that nothing unusual (fallen trees, a broken down car, etc.) blocked its path, since in such a situation it would have immediately run out of ideas how to rescue the situation. In other words, IAAs like Stanley will not substitute human drivers unless it will be furnished with a lot more AI. At present it, in contrast to RA, just simulates that part of human behavior, which does not involve any noticible deliberations (like walking around or driving a car – without putting much attention to the processes involved in this kind of behavior, possibly even drawing at the same time full conscious attention to a conversation with some companion).

One might argue about the desirability of substituting human private car drivers by IAAs (as indirectly suggested by Stanley's development), especially since it is obvious that our earth cannot tolerate a further increase in individual traffic with all its consequences. A far more desirable goal is to improve the public transportation system for economical as well as ecological reasons. Traffic jams worldwide cause the loss of gigantic amounts of money. In the European Union alone, these costs are estimated at 135 billions of Euros per year. With a more efficient public transportation system, this huge waste of money could be avoided, let alone the enormous benefits for saving energy and the environment.

The challenge is to evolve the system to one that can compete with individual driving in terms of traffic time and comfort for the passengers. At present, cars are used because luggage can be put into the trunk at the point of departure and taken out at the point of arrival (while it has to be lugged around within the public system), travel time (from door to door) typically is still far shorter, and costs often are not really higher than that with the available public system. This is so, because the public system is so badly designed and run. It totally ignores the *last mile* (say from the bus station to home), offers time-wasting connections between different transportation modes (bus, train, plane, etc.), does not comfortably support luggage transportation, has no flexible and comfortable payment and information systems, and so forth.[3]

True, a system overcoming such weaknesses faces enormous challenges to be overcome due to the complexity of the anticipated system, which can no more be coped with easily by human actors alone. It is, therefore, here, where IAAs enhanced

[3] There might also be an evolutionary reason for the preference of individual transportation means by people, of the kind as discussed in [10]. As we argue in Section 5 IAAs should help in overcoming such psychological shortcomings carried over from long-passed ages.

by existing deliberative AI methods could excel. Such an enhanced IAA could plan the optimal route for an individual passenger and for his or her luggage from door to door, and adapt the routes and travel times of transportation vehicles appropriately to minimize average passengers' travelling times (instead of the fixed and hence stupid timetables currently used even for peripheral routes). The hundreds of billions of Euros currently (2009) invested by a number of countries (USA, Germany, France, etc.) to save big car companies from bankruptcy would, therefore, far better be used for purposes of the kind just described. This is for the simple reason that, for instance, a bus with 50 passengers wastes only a tiny fraction of the energy, space, material, etc. otherwise required for upto 50 cars, so that the endangered environment would benefit grossly.

What we illustrated for a future public transportation system applies to numerous other areas and applications where deliberative AI methods in connection with goal-based and utility-based agents technology would offer truly remarkable improvements. We just remind the reader of the possibilities listed already in Section 2.3 (financial system, legal system, maintenance of technical systems like computers, household appliances, weather stations, etc.). Additional examples are service robots, recommender systems for customers, decision support systems, which consider the user's preferences, decision support systems for medical applications, scheduling systems in production applications, automated trading systems, business management systems, and so forth. The methods needed to realize all these IAA possibilities are available. Some of the methods are not yet known widely enough to steer large investments into this kind of IAA technology. Some are used already in the rapidly growing sector of computer games, in which human users interact with IAAs in the form of artificial characters in a simulated world (e.g., Black and White, F.E.A.R., etc.).

We end this section by pointing to one particularly intriguing AI method, called metacognitive loop (MCL), successfully tested in highly varied domains [2]. MCL addresses the brittleness of current systems, which are doing well in a certain domain but completely fail, if some unexpected problem occurs. It gives autonomous systems the ability to notice, assess, and repair such problems. Central to the MCL approach is "building systems that don't simply do things but know what they are supposed to be achieving ... and ... are also self-diagnosing and self-repairing" [2, p.68]. This is already a first tiny step towards a kind of consciousness, which we will be asking for in the next section. At any rate, knowing its own goals seems to be a prerequisite for an intelligent autonomous agent in a stricter sense of this term.

5 Longer-Term Perspectives

Recall that we pointed out in Section 2.3 two different motivations for the study of IAAs, namely understanding and technology improvement. The two are so densely interwoven that their joint exploration would generate considerable synergy while, at present, they are actually explored separately with occasional interactions only. Such a truly joint exploration would be possible within a common discipline, if it existed. This is the reason why the author has proposed for decades to establish the discipline of Intellectics as the union of AI, Cognitive Science, Cognitive Neuroscience, and others [4]. In the present section, we will speculate about the future of IAAs and argue that such a common disciplinary study would speed up both the understanding and development of IAAs.

As to the second motivation of technology improvement, we have listed in Section 2.3 already a number of attractive special applications of IAAs. One might, and should, however, first step back and clarify what the general aims behind IAAs are before thinking of particular applications. Let me, therefore, start with a rather general perspective into the future.

Natural IAAs have evolved over millions, in fact even billions, of years [10]. Evolution represents an ingenious mechanism, which eventually leads to extremely well designed IAAs within their environments. Unfortunately, the mechanism is a rather slow one, taking hundreds or thousands of generations of a species for relatively small adaptations. It is certainly too slow for coping with the rapid changes of the environments, which are caused by human activities (e.g., climate change, changes in food supply, social realities, etc.).

Hence, the general vision is that technically designed IAAs help humanity to avoid as much as possible superfluous and disadvantageous changes, such as the climate change, but also to assist human beings in enhancing their capabilities. Namely, human beings have adapted habits which fit perfectly for the life-styles of the stone-age but cause numerous serious problems within our modern world [10]. Solving these problems requires higher-level mechanisms, which do not prevail naturally in us, but could be designed artificially.

Under these general perspectives which amounts to nothing less than an enhancement of evolution, we describe five of the challenges, which we are facing as we go about designing IAAs of the kind serving such general aims. One of these challenges involves the logical approach to AI, which amounts to such a higher-level and rational mechanism now urgently needed generally and in many particular applications. This particular approach is, therefore, briefly described as well, also in order to clarify some existing myths about it.

5.1 Five Challenges

In [5, eg. Sect.1.6], the author has pointed out the importance of consciousness also for the human low-level apparatus for enabling balance, locomotion, etc. Nature has apparently invested a lot into the feedback from this apparatus to the brain, which by our current proprioceptive sensor measurements realized in IAA technology are only very poorly simulated. But, what is this kind of self-consciousness, which apparently contributes to animals' elegant and efficient locomotion abilities? Had we a better understanding of it, then – and perhaps even only then – we could enhance IAAs' locomotion and actuation capabilities to the level of animals. That is why, neuroscientific, cognitive, AI and IAA research must go hand in hand to *uncover* *the mystery of consciousness and its role in IAAs*, certainly a long-range goal and perhaps the greatest challenge ever in science.

From self-observation, it is also apparent to us that our human problem solving capabilities are enabled to a considerable degree by the conscious deliberations by which we activate the salient knowledge and improvise the possible solutions to a given problem in our imagination. Similarly, an IAA would need the ability to bring its knowledge to bear on self-experienced models of potential solutions. So again, understanding consciousness might be a prerequisite for any true artificial intelligence.

In Section 3, we distinguished (at least) three different functional levels in our imagined IAA Flocke. A robot like Stanley realizes only the lowest and intermediary level. On the other hand, there are robots and systems with functions at the highest level of deliberation. In fact one of the first robots, Shakey, featured already deliberative functions like planning, reasoning, and learning, and was the very first one to do so [21, Ch.12]. However, in current IAA technology there is a wide gap between the high and the low level, which is in stark contrast with human functionality. For instance, think of the human way of running downhill in rough terrain, of playing tennis, or of rehearsing a piece of music on some instrument, etc. In all these cases, there seems to be a close collaboration of the high and low functional levels in our brains in a way, we do not currently understand.

We do, at present, not know how to intertwine the two levels in a beneficial way, especially how to feed the low level on the basis of results obtained on the high level. That is why, robots like Stanley are relatively smart in the locomotive behavior but otherwise completely stupid; inversely, robots with deliberative capabilies typically are poor in their locomotive or actuative behavior, at any rate they are unable to take full advantage of the deliberative capabilities for improving the low-level behavior. In this weakness in *bridging the gulf between the high and the low level functions in IAAs* lies another great challenge for future research, and again progress can only be achieved in this direction, if understanding the respective human functions go hand in hand with technological improvements. The AI community is well aware of this particular challenge and has taken steps in meeting it, e.g., under the topic of hybrid agent architectures [24, p.971f].

The third challenge concerns software agents only (introduced in Section 2.2), i.e., no embodiment is needed to be considered here. Let us start by noting that Information Technology (IT), today, is a substantial part of all science and engineering. Take the engineering of a car as an example. Its design today is exclusively carried out with (CAD and other) IT systems, its anticipated behavior is studied in computer simulations, its production including a just-in-time supply chain is controlled by computer systems, and the same is true for the distribution and marketing of the products. To some extent the systems involved are intelligent, to some extent they are autonomous. Still, one would not readily associate the concept of IAAs with the kind of systems involved in today's car industry except for the robots installed in the production line. But, clearly we are on our way into the direction of engaging IAAs in every section of this and other industries. They will act as assistants in the engineers' daily work, supporting a consistent specification of the product, helping in designing the product in line with its specification, pointing out alternatives or oversights in the design, and so forth. In fact, some of our products have become so complex that their design can only be accomplished in collaboration with an IAA of this kind, because the complexity surmounts human capabilities.

So, as already noted in Section 2.3, IAAs are needed for many of these tasks because they have become too complex for human beings to manage them unaided. But, now imagine how much more complex the task will be to develop such an IAA, e.g., a *designer IAA*, which should help to solve a complex task like designing cars. So, if we need an IAA for designing products like a car, how much more would we need an IAA to design the IAA for designing a car. Obviously, we face here an infinite regress. Hence, the challenge is whether we can find a way out of this regress.

The most obvious approach to cope with this challenge is to engage the most brilliant minds on earth in the design of IAAs, which is certainly a good idea to start with. But, even the most brilliant minds have only limited creative capacity, which may not suffice to design the ultimate designer IAA, say. Therefore, the next idea might be to think of a system which is used to improve its own performance (which reminds of Till Eulenspiegel who pulled himself out of a swamp with his hands at his plait of hair). There are, in fact, already AI approaches of this kind in use.

One approach uses the technique of Genetic Programming (GP), a search method based on simulating the processes of evolution and a relative of Genetic Algorithms (GA). In a series of books, its inventor, John Koza, demonstrates that designs like those of an optimal antenna system or of optical lenses may evolve from dozens or hundreds of generations [19, 20, 17, 18]. In these applications, GP evolves LISP programs. It starts with a random collection of programs containing some basic LISP functions and constants thought to be important for solving the task at hand and applies biologically inspired operations like random mutations, cross-over and others to result in new generations of functions, which might solve the problem at hand. In [22], a GA is used for evolving fuzzy rules for solving the path generation problem of multiple robots with a fuzzy logic controller.

GP is a special technique from a wider area, which is known as metaheuristics or, more specifically, stochastic local search [16]. These and other search techniques are especially promising in relatively narrow domains, which give rise to a relatively homogeneous search space. In so far one is tempted to put the class of domains, in which they excel in analogy with the domains of low level behavior of IAAs. For those, we have just pointed out in the preceding challenge the gulf, which separates it from the high level of deliberation, that is, we identify here a similar gulf between (relatively low-level) metaheuristics techniques on one side and high-level cognitive functions on the other. In other words, it is these high-level functions which remain isolated in our current IAA technology from all the remaining functions. So, the first part of the third challenge asks for *bridging the gulf between the high-level cognitive functions and the functions realizable by metaheuristics*. This amounts to integrate the results of deliberations into the mechanism of these search methods which has not been tried so far.

The second part of this challenge or rather, for convenience, the fourth challenge, amounts to the development in IAAs of the high-level cognitive functions like reasoning, planning, learning, conceptualisation, abstraction, metacognition, and so forth. We have made impressive progress in the last decades in this respect, but are still far from a performance level comparable with that of human beings. Also, we are far from a deeper understanding how these functions are realized in human beings.

It is this challenge, which has been the focus of the logical approach to AI for decades. This approach has two different aspects. One is to implant into IAAs true intelligence comparable with that of human beings. The other is that logic-oriented scientists have tried and still try to overcome the infinite regress mentioned before by representing the problems faced in the quest for an artificial intelligence in some logical formalism and solve the resulting logical problem with deductive means. In other words, the fourth challenge again splits into two parts, so that we may again number them separately and refer to the fifth challenge as the *logical approach to realizing AI*, while the fourth is restricted to *implanting truly high-level intelligence functions into IAAs*.

In the remainder of this section, we will plead once again for this exciting approach, which is still promising if used wisely and which addresses both challenges at the same time. In particular, we will describe in more details, how the high-level intelligent functions would interact with the low-level functions dominated by perception and control, an issue also addressed by Challenge 2.

In summary, we have thus advanced the following five related challenges:

1. Uncover the mystery of consciousness and its role in IAAs.
2. Bridge the gulf between the high-level and the low-level functions in IAAs.
3. Bridge the gulf between the high-level cognitive functions and the functions realizable by metaheuristics.
4. Implant truly high-level intelligence functions into IAAs.
5. Develop the logical approach to realizing AI.

5.2 *The Logical Approach to IAAs*

All of the scientific knowledge is communicated and passed on to next generations in some languages (natural, artificial, mathematical or formal). From this body of knowledge along with new experiences and hypotheses, new knowledge is generated by scientists, engineers, and others, again represented in some languages. So, scientific discovery may be represented as a process, which transforms a body of language constructs to another one. The same is true for any other kind of problem solving. Logic, on the other hand, is a form of abstracted language along with rules which simulate at least some important aspects of human reasoning.

Given the two facts just described, the idea is natural to try to figure out whether the language transformations involved in the scientific discovery or problem solving processes could not be represented as logical transformations. If this could be done, then we would be able to speed up the discovery process by automating logic, and thus, the process of discovery and problem solving. This is the dream driving the logical approach already since Gottfried Wilhelm Leibniz in the 17th century.

To reiterate in other words, the logical approach tries to avoid the infinite regress by developing basically one single IAA in the form of a general logical problem solver (GLPS) for supporting scientific discovery as well as the development of any other specialized IAAs – including those which would solve our fourth challenge. That is, we are talking here of a meta-approach: rather than developing myriads of specialized IAAs, the goal is to work out a single one which would serve as a tool to figure out any of the others. This, it is believed, could overcome the infinite regress by putting all efforts into solving one confined problem.

The logical approach has already been extremely successful along its route so far. It has led to many facets in AI technology. For instance, the widely used technology of knowledge systems, logical programming languages like PROLOG, verification technology for hardware and software, planning systems, and numerous others are some of its pleasant and fruitful side effects. But, despite these great successes we are still far away from our ultimate goal.

The development of a GLPS is obviously a truly grand research goal. It could well be compared with grand goals in other disciplines like the physics dream to produce energy by the way of fusion reaction as realized in the sun. To further pursue this physics dream, an international project is currently carried out to design and build an experimental fusion reactor, called ITER and based on the *tokamak* concept. A number of billions of Euros are invested into ITER. In contrast to this the logical dream is pursued at a rather small scale and has remained the endeavor of a limited number of independent researchers. Nowhere in the world, can one find an analoguous institution for pursuing logical studies like, say, CERN in Geneva in the case of physics.

With this analogy between the two dreams, we want to make clear what is at stake here. First of all, building a GLPS is too complex to be solved by the way of any number of PhD theses (which, at present, is the main driving force in our field). Like fusion, it had to be pursued on a much grander scale with the brightest minds available. And, like fusion, it will not be completed in just a couple of years despite decades of research already spent on it. On the other hand, GLPS might be regarded as more general than ITER because, had we GLPS available now, then the development of ITER could profit immensely from its capabilities (while conversely I doubt whether fusion will be a necessary prerequisite for the energy needs of GLPS). Also, as evidenced by the examples of success along the road mentioned above, the GLPS approach can be regarded as an *anytime* technology; namely at any time along its development route it offers partial results, which are already useful in numerous applications, including the efforts to build an GLPS. This is in stark contrast with a project like ITER, which will not be profitable in any way before its final goal is fully reached. Hence, by way of these comparative deliberations the question naturally arises, whether we should not spend more on GLPS before spending such huge amounts of money on fusion.

Interest in the logical approach to AI and to an GLPS temporarily waned not to the least because of several myths about its nature. Let us clarify one of these. In

Section 3.1, we mentioned that Brooks claimed that GOFAI has foundered on the issue of representation. He did have a valid point at that time, when indeed there was insufficient emphasis on reactive behavior. But, he also left us under the impression that he then erroneously thought that logic-oriented scientists would want to realize the low-level control of IAAs by way of logical operators rather than by reactive control mechanisms. If so, he would have erred because he apparently would have confused two different levels involved here: At the control level, the reactions of the IAA are of course driven by some low-level mechanism; the development of such a mechanism, however, is the subject of a task given to GLPS to be solved at the development level, a completely different thing.

There might be a reason, why Brooks had fallen into the trap of this confusion. Namely, logic has the unique nice feature that the task specification formalized in logic may already be run as a prototype control program. In other words, the specification at the development level may preliminarily be also used at the control level for testing the specification. So, Brooks and others perhaps erroneously thought that logic-oriented scientists are so naïve to confuse their prototype logic programs with the desired efficient end product. Of course, the end product has eventually to be generated from the specification in some way, for which there are several different but related possibilities.

One of these possibilities consists in claiming the existence of a solution under the formal logical representation of the problem and of the knowledge available for solving it. The constructive proof of this claim by some deductive system, if it did find it, would result in a so-called answer term, which represents the solution mechanism. The other possibility consists in modelling the input-output behavior of the mechanism in logical terms (like the above mentioned prototype control program) and synthesizing from there an efficient low-level program (similar to the hybrid architectures mentioned in the previous subsection). A further popular approach consists in interactively designing the system on the basis of some constructive logic with types [1]. Unfortunately, too little research is carried out these days in any of these directions and the whole area of program synthesis is a sleeping beauty.

In summary, the logical vision comprises two main parts. In its *representational part*, we envisage a huge accumulation of knowledge of all kinds represented in formal logical terms, which continuously expands. Flocke, as we baptized our imagined IAA, would have full access to all this knowledge and would also include its own problem specification.

The other *operational part* is characterized by the knowledge operations defined again in logical terms and developed once and for all, independent from any specific problem. These operations include deduction, synthesis, compilation, optimization, and so forth. They would synthesize from the specification and the knowledge base the complex mechanisms to be installed at the lower layers of Flocke, which actually drive Flocke's behavior. They would also inductively generate new knowledge from Flocke's experiences and sensor data to be added to Flocke's rich knowledge base (described in the representational part). Note in this context that induction is nothing else than a form of deduction that takes into account meta-knowledge describing inductive strategies.

An important aspect is that the operational part would be activated not only during the design phase of Flocke but would also remain active during its entire lifetime in various modes. In one of these modes it would synthesize plans figured out on the basis of the robot's recent intentions to appropriate behavioral changes on the lower layers of the robot's architecture. In another mode, it would adapt the mechanisms on the lower layers to newly acquired knowledge by reactivating the synthesis process, whereby only those parts of the process need to be activated which are affected by the change in the knowledge state. In this and further ways the deliberative layer, as it will be called in the following section, would become an essential part in future IAAs and solve our fourth challenge.

As an aside, if we had already a fully satisfactory solution for the operational part of the logical approach then the pressing user interface problems would be solved at the same time. This is due to the fact that users could adapt the system specification to their personal needs in a natural descriptive way and ask the system to accommodate the low level processes to these changes of the specification.

Current practice to some extent does separate the tasks of specification and developing from there a workable system, similarly as just described. Often the Universal Modelling Language (UML) is used for the specificational part. UML, thus, represents a first step into the right direction. But, UML features a number of deficiencies, lacks the solid theoretical basis enjoyed by logic and especially, anything like the operational part just described.

Given all these considerations in the context of the state of affairs in Automated Deduction and Reasoning, a field which boasts of considerable progress in some of the directions outlined above, the present author continues to make a strong case for advancing the technology of deduction and of synthesis on a logical basis, also and especially, in view of advancing the technology of IAAs. In [7], the author has outlined a research program for deduction serving this purpose. This includes a continuation of the search for a suitable logic in the first place.

6 Considerations for the Development of an IAA

In this section, we summarize our view of what has to be done when faced with the task of designing and developing an advanced IAA given the current state of the art. Let us continue to name it Flocke. Like in the rest of this chapter, we, thereby, focus on the functional design on a rather high level of abstraction rather than going into any details, such as material and mechanical aspects of Flocke.

As we emphasized in Section 2.2, the first step consists of specifying the kind of environment in which Flocke is supposed to operate, whereby the environment includes Flocke's own body. We want Flocke to behave in its environment as intelligent as possible. Of course, intelligence is a rather fuzzy concept, so that we have to be more specific about the kind of intelligent behavior we have in mind and how it should be realized.

We believe that Flocke's architecture should be a hybrid one of the kind already mentioned in Section 5.1, which combines deliberation with reaction. This combination can be achieved by an n-layer architecture, whereby n in a first coarse partition typically is three [15]; i.e., Flocke might feature a three-layer architecture (recall the three functional levels described in Section 3). It consists of a *deliberative* layer, an *executive* layer, and a *reactive* layer.

On the deliberative layer, we represent the wealth of accumulated knowledge in an explicit form, including Flocke's goals, desires, intentions, plans, and utilities, a detailed model of the environment including Flocke's own capabilities, general world knowledge, descriptions and code of existing solutions to pertinent problems, and so forth. At this layer, we also represent all kinds of deliberative capabilities, such as reasoning, planning, deduction, metacognitive abilities, etc. We believe that intelligence, in a stricter sense, will not be possible without such a powerful deliberative layer and without bringing its influence to bear on the agent's behavior. In it, the *real* intelligence resides that we experience in human beings, which is not at all meant to say that the lower layers lack any intelligence, as we will see in the following.

The reactive layer controls the low-level functions of Flocke in the sense of Section 3.1. As we mentioned there, it may itself be structured again in a layered way but in our coarse partition it is regarded as one single layer. It is able to operate without the interference of the higher layers. But, the executive layer may occasionally interact and modify the automatic sensor-action loop in place. In this way, we can avoid the stupidity illustrated with flies in Section 3.1. We want to represent as much as possible the low-level intelligence at the reactive layer. A part of it will be motion intelligence as studied in biology. Another part is behavioral intelligence like that exhibited by insects, such as ants. But, this low-level intelligence must be subjected to changes and adaptions on the basis of new experiences and of learning mechanisms at the higher layers.

The executive layer plays the role of a supervisor to the reactive layer as just described. It, thereby, follows the instructions received from the deliberative layer and translates them into action sequences or changes in the finite state machine installed at the lowest layer. It also feeds the deliberative layer with new knowledge about the environment extracted from sensor data, so that the model of the environment is refined with time through the agent's experiences. The executive layer might also be the place, where the kind of consciousness described as the first challenge in Section 5.1 should be installed, in order to support among other tasks, the optimization of the agent's action sequences. Metaheuristics techniques, such as those described in the context of the third challenge in Section 5.1, e.g., for the fine-tuning of behavior, also fit the best within the executive layer. The same is true for learning techniques of the kind of reinforcement learning, while logical learning algorithms and induction operating on explicit knowledge would rather be located at the deliberative layer. Altogether this would amount to a considerable level of intelligence also at this intermediate layer.

In our vision outlined in Subsection 5.2, the deliberative layer would serve two different functions, one for the design phase of Flocke, the other for its lifetime.

In the design phase, we would specify all the requirements just outlined in logical terms and deduce from there the design and implementation of Flocke, notably its two lower layers. Thereby, the specification would of course include known problem solutions in the form of given algorithms, programs and specifications of existent devices, i.e., the development would be built on the technological basis reached so far in this area rather than starting from the scratch. During Flocke's lifetime the knowledge accumulated in the deliberative layer will increase. In the ongoing change of this layer lies the potential to improve Flocke's design at the lower layers by restarting the deductive mechanisms to identify and install promising changes in the implementation.

In the meantime, until this vision can at least partially be realized, a Flocke of the kind just described needs to be programmed in the traditional way, which is to say that much of the envisioned function of the deliberative layer will be carried out by engineers and programmers rather than by the robot itself.

7 Conclusions

The area of intelligent autonomous agents (IAAs), or systems, spans a wide variety of aspects. This chapter could, therefore, only touch on some general aspects. In particular, we wanted to outline the area at a high level of abstraction and to point to the directions of research with the greatest promise.

In our opinion, IAA technology at its lower level functionality has reached a maturity, which allows it now to focus on high-level cognitive functions. Since some of these functions have already been studied extensively in the discipline of Artificial Intelligence (AI), we made a case in Section 4 towards incorporating existing AI methods into the IAAs of the next generation. In the longer term perspective treated in Section 5, we put forward five challenges which we consider to be crucial for coming closer to autonomous agents with true intelligence. In this context, we have pleaded for the logical approach to AI, pointing out its unrivalled advantages and promises.

As just said, this chapter could in the given space only address a few important general aspects of autonomous systems. Much has just been alluded to, such as many technical issues of detail, but also ethical and regulatory issues (Sect.2.3), which were just mentioned in passing; much could not be noted at all, such as inter-agent communication, negotiation, coordination, conflict, and social behavior which are fundamental for any society of autonomous agents.

The contents of the entire book will compensate for this imperfection at least to some extent. Altogether it demonstrates the great promise and potential of the important area of IAAs. As one of the potential application sectors, we have outlined in Section 4.1, an IAA-based flexible future public transportation system which would save tremendous amounts of money and above all benefit the environment.

In the grand picture, the development of IAAs could become a crucial part in the forthcoming period of enhanced evolution. Namely, one could speculate that IAAs, perhaps playing the role as our assistants and rational partners, could help to overcome the deficiencies of human beings within our modern environments. These deficiencies stem from human features evolved in the stone-age environment, some of them completely inappropriate or even harmful for our global, mechanized world. Such rational partners could guide us in realizing an altogether more rational behavior to the benefit of our and other species and of the entire globe.

Acknowledgments. The author is very grateful to Prof. Pratihar for the invitation to write this chapter as well as for his helpful advice. Thanks are also due to Sebastian Thrun, whose helpful suggestions led to an improvement of the text, as well as to Josef Schneeberger for further valuable comments.

References

1. Allen, S.F., Bickford, M., Constable, R.L., Eaton, R., Kreitz, C., Lorigo, L., Moran, E.: Innovations in computational type theory using nuprl. Journal of Applied Logic 4(4), 428–469 (2006)
2. Anderson, M.L., Fults, S., Josyula, D.P., Oats, T., Perlis, D., Schmill, M., Wilson, S., Wright, D.: A self-help guide for autonomous systems. AI Magazine 29(2), 67–76 (2008)
3. Behnke, S.: Humanoid robots – from fiction to reality? KI - Künstliche Intelligenz 4/08, 5–9 (2008)
4. Bibel, W.: Intellectics. In: Shapiro, S.C. (ed.) Encyclopedia of Artificial Intelligence, pp. 705–706. John Wiley, New York (1992)
5. Bibel, W.: Lehren vom Leben – Essays über Mensch und Gesellschaft. Sozialwissenschaft. Deutscher Universitäts-Verlag, Wiesbaden (2003)
6. Wolfgang Bibel, L.: AI and the conquest of complexity in law. Artificial Intelligence and Law Journal 12, 159–180 (2005)
7. Bibel, W.: Research perspectives for logic and deduction. In: Stock, O., Schaerf, M. (eds.) Reasoning, Action and Interaction in AI Theories and Systems. LNCS (LNAI), vol. 4155, pp. 25–43. Springer, Heidelberg (2006)
8. Braitenberg, V.: Vehicles: Experiments in Synthetic Psychology. MIT Press, Cambridge (1984)
9. Brooks, R.A.: Intelligence without representation. Artificial Intelligence Journal 47, 139–159 (1991)
10. Buss, D.M.: Evolutionary Psychology – The New Science of the Mind, 3rd edn. Pearson, Boston (2009)
11. Collins, S., Ruina, A., Tedrake, R., Wisse, M.: Efficient bipedal robots based on passive-dynamic walkers. Science 307, 1082–1085 (2005)
12. Dickmanns, E., Behringer, R., Dickmanns, D., Hildebrandt, T., Maurer, M., Schiehlen, J., Thomanek, F.: The seeing passenger car VaMoRs-P. In: Proceedings of the International Symposium on Intelligent Vehicles, Paris, France, pp. 68–73 (1994)
13. Dearden, A., Demiris, Y.: Learning forward models for robots. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI 2005, Nineteenth International Joint Conference on Artificial Intelligence, pp. 1440–1445 (2005)
14. Dörner, D.: Bauplan für eine Seele. Sachbuch 61193. Rowohlt Taschenbuch Verlag, Reinbek (2001)
15. Gatt, E.: Three-layered architectures. In: Kortenkamp, D., Bonasso, R.P., Murphy, R. (eds.) AI-based Mobile Robots: Case Studies of Successful Robot Systems, pp. 195–210. MIT Press, Cambridge (1998)

16. Hoos, H.H., Stützle, T.: Stochastic Local Search – Foundations and Applications. Morgan Kaufmann, San Francisco (2004)
17. Koza, J.R., Bennett III, F.H., Andre, D., Keane, M.A.: Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufmann Publishers, Inc., San Francisco (1999)
18. Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G.: Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Kluwer Academic Publishers, Norwell (2003)
19. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge (1992)
20. Koza, J.R.: Genetic Programming II: Automatic Discovery of Reusable Programs. The MIT Press, Cambridge (1994)
21. Nilsson, N.J.: The Quest for Artificial Intelligence: A History of Ideas and Achievements. Cambridge University Press, Cambridge (2010)
22. Pratihar, D.K., Bibel, W.: Time-optimal, collision-free path generation for multiple robots working in the same workspace using a genetic-fuzzy system. Machine Intelligence and Robotic Control 5(2), 45–58 (2003)
23. Pratihar, D.K., Hui, N.B.: Evolution of fuzzy controllers and applications. In: Jain, L.C., Palade, V., Srinivasan, D. (eds.) Advances in Evolutionary Computing for System Design. Studies in Computational Intelligence, vol. 66, pp. 47–69. Springer, Berlin (2007)
24. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall Series in Artificial Intelligence. Pearson Education, Upper Saddle River (2003)
25. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press, Cambridge (2005)
26. Thrun, S., Montemerlo, M., et al.: Stanley: The robot that won the DARPA Grand Challenge. Journal of Field Robotics 23(9), 661–692 (2006)
27. Wolpert, D.M., Miall, R.C., Kawato, M.: Internal models in the cerebellum. Trends in Cognitive Sciences 2(9), 338–347 (1998)

3

Design and Development of Intelligent Autonomous Robots

Nirmal Baran Hui and Dilip Kumar Pratihar

Abstract. The present chapter deals with the issues related to design and development of autonomous mobile robots. One of the major issues in developing an intelligent and autonomous robot is to design an appropriate scheme for planning its motion without any human intervention. Both conventional potential field method as well as soft computing-based approaches have been developed for the said purpose. Initially, the performances of all the approaches have been studied through computer simulations. Thereafter, real experiments are conducted to test the effectiveness of the said approaches. A camera-based vision system has been used to collect information of the environment, while carrying out the real experiments.

Keywords: Autonomous mobile robots, Neuro-fuzzy system, Genetic-fuzzy system, Genetic-neural system, Potential field method.

Nirmal Baran Hui
Assistant Professor
Department of Mechanical Engineering
National Institute of Technology, Durgapur
West Bengal – 713209
India
e-mail: nirmal.hui@nitdgp.ac.in

Dilip Kumar Pratihar
Professor
Soft Computing Lab.
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur
Kharagpur-721 302
India
e-mail: dkpra@mech.iitkgp.ernet.in

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 29–56.
springerlink.com

1 Introduction

Current research in robotics mainly focuses on design and development of intelligent and autonomous mobile robots. They are used in industry, space, sea-beds, military and security environments, and appear as consumer products. Mobile robots include wheeled robots, legged robots (e.g., hexapod, quadrapod, biped and others), tracked vehicles and aerial robots, which are usually referred to as Unmanned Aerial Vehicles (UAVs). Robots may be controlled directly by a human being, such as remotely-controlled robots and robotic arms or may act according to their own decision making ability provided by artificial intelligence. However, the majority of robots fall in-between these extremes and are controlled by a pre-programmed computer.

Mobile robots are mostly known as automatic machines that are capable of moving in a dynamic environment. The control of such a robot involves three distinct phases - perception, processing and action. Sensors / cameras give information of the environment (i.e., the positions and orientations of both the robot as well as other objects present in the environment). Using strategies related to the field of control theory, this information is processed to determine appropriate signals for the actuators (motors), which drive the mechanical structure to its desired location. Moreover, control of a mobile robot also involves aspects like path planning, navigation, pattern recognition, obstacle avoidance, and others. To give a proper dimension to this field, the robotics researchers are trying to enable the robot to cope with its environment, which could be on land, underwater, in the air, underground or in space. To achieve this, a robot will have to be intelligent and autonomous one.

1.1 Autonomous Mobile Robots

Autonomous robots are those which can perform desired tasks in unstructured environments without continuous human guidance. For example, a high degree of autonomy is particularly desirable in the fields, such as space exploration, where communication delays and interruptions are unavoidable. Some modern factories (particularly in Japan) also include a number of autonomous robots.

In 1993, Alan Mackworth proposed in a paper entitled ”On Seeing Robots” that soccer could be a good test-bed of robotics and AI research [26]. Thus, RoboCup (i.e., World Cup Robot Soccer) has been considered as a vehicle to promote robotics and AI research. The ultimate goal of the RoboCup has been set as follows [4]: ”By the mid-21-st century, a team of autonomous humanoid robots shall beat the human world cup champion team under the official regulations of FIFA”.

To meet these requirements, the motion planner and controller of the robot should have the following properties [3]:

- *Programmability:* a useful robot should be designed and programmed, in detail, to perform multiple tasks and its functions should be combined according to the task to be executed.

- *Autonomy and adaptability:* a robot should be able to carry out its actions and modify the task and its own behavior, as the situation demands.
- *Reactivity:* a robot must take into account events with time bound compatible with the correct and efficient achievement of its goal (including its safety).
- *Consistent behavior:* a robot's reaction to events must be guided by its task objectives.
- *Robustness:* control architecture should be able to exploit the redundancy of the processing functions. It will require the control to be decentralized to some extent.
- *Extensibility:* integrations of new functions and definitions of new tasks should be easy. Learning capabilities are important to be considered here.

A robot may also be able to learn autonomously. Autonomous learning includes the ability to: (i) learn or gain new capabilities without any outside assistance, (ii) adjust strategies based on the surroundings and (iii) adapt to the surroundings without any outside assistance. However, autonomous robots still require regular maintenance, as do other machines.

During the last few years, a variety of mobile intelligent robots had been designed to be operated in highly cluttered environments. These robots have been deployed in hospitals, office buildings, department stores and museums and can perform various services, such as delivery, education, providing tele-presence, cleaning or entertainment. Furthermore, a few prototypes of autonomous wheelchairs and intelligent service robots are also available in the market, which are basically designed to assist people in their homes. To fulfill these increasing demands, current research in robotics aims to build an intelligent and autonomous robot. An autonomous robot requires minimal or no human interactions to perform the tasks and it should have the capability to plan its motion in a highly dynamic environment. Therefore, in an unknown or a partially-known environment, an autonomous robot should be able to perform the following tasks:

- collecting information of the environment with the help of sensors and/or cameras,
- building world model,
- taking the decision as the situation demands without any intervention from human,
- planning its motion in a timely fashion,
- learning from past experience to improve its performance.

Moreover, it may need to behave cooperatively by keeping proper communication with other robots or agents, possibly including humans. Hence, it should have *a real-time sensing assembly, an intelligent decision maker* and **precise actuators**, which lead to the development of hardwares like sensors, motor actuators and softwares, such as motion planning and control algorithms. Over the last few decades, much attention had been paid to improve the intelligence of the robot and many traditional and a few non-traditional approaches came out in the related literature. However, the progress of this field is not sufficient enough, in spite of the excitement and relatively rapid advancement of the early research. Hence, the need for an adaptive decision making scheme is still felt in this field of research.

2 Literature Review

The prime aim of robot motion planning is to determine its collision-free paths in varying situations, while moving from an initial position to a final position. The environment may be either static or dynamic. Planning the paths in a structured environment may be easy due to the fact that the obstacles are static and the geometries of both the robot as well as the obstacles are known beforehand. Thus, it is possible to form a global map of the environment and the task is executed after the complete solution is obtained. Both analytical methods (namely potential field approach proposed by Khatib [19] and other techniques) as well as graph-based methods, such as visibility graph [30], Voronoi diagram [24], tangent graph [25] and others, have been applied for this purpose. However, avoiding collisions in a dynamic environment, in which the obstacles are moving with the varying speeds in different directions seems to be a tough task for the robot motion planner.

2.1 Robot Motion Planning Approaches

Motion planning of a robot navigating among moving obstacles depends on the information of the environment collected through sensors/cameras, on-line. Thus, building a global map of the environment is not possible. Various investigators have developed some suitable methods of motion planning for this purpose. These are based on either algorithmic approaches or soft computing [34]. Some of these methods are mentioned below.

2.1.1 Algorithmic Approaches

A large number of algorithmic approaches, such as path velocity decomposition [18], accessibility graph [14], space-time concept [21], incremental planning [38], relative velocity approach [12], potential field method [19] and others are available in the literature. Latombe [22] provides with an extensive survey of these approaches. Borenstein and Koren [6, 7] developed two modified versions of potential field approach, namely virtual force field and vector field histogram for real-time obstacles' avoidance of the mobile robots. Out of all these algorithmic methods, potential field method is found to be the most popular one. Moreover, Svestka and Overmars [39] proposed an approach based on probabilistic road-maps for motion planning of the car-like robots. However, all the algorithmic methods have some drawbacks. Some of these are as follows: (a) not all the approaches are computationally tractable for solving a variety of complex motion planning problems, on-line, (b) each of these methods is suitable for a particular type of problems, however, there is no versatile algorithm which can provide with solution to a variety of motion planning problems, (c) as most of the approaches do not have any in-built optimization module, the generated motion of the robot may not be optimal, (d) for some of these methods, the solutions may get stuck at the local minima. Thus, there is still a need to develop an efficient, adaptive, versatile and computationally tractable algorithm for solving the motion planning problem of mobile robots.

2.1.2 Soft Computing-Based Approaches

Fuzzy logic techniques, neural networks, genetic algorithms and their different combinations (popularly known as soft computing approaches [36]) are found to provide with feasible solutions to some complex real-world problems within a reasonable accuracy. Moreover, the computational complexity of such methods is expected to be low due to the fact that most of them are heuristic in nature and an exact mathematical formulation of the problem is not necessary. Understanding the advantages of these approaches, quite a few researchers have tried to use them for solving a variety of robot motion planning problems.

Fraichard and Garnier [13], Akbarzadeh et al. [2] used a Fuzzy Logic Controller (FLC) for planning the motion of a mobile robot in dynamic environments. However, the performance of an FLC depends on the selection of membership function distributions (known as data base) and its rule base. In most of the fuzzy control systems, fuzzy if-then rules are designed by human experts. A human expert may find it difficult to express the control actions, which are often decided on a subconscious level. Thus, there is a possibility that the manually designed rule base may not provide with the optimal solution and may have a less relevance to the problem under study.

Several methods had been proposed by various investigators to develop an optimal knowledge base (consisting of both rule base as well as data base) of an FLC. Quite a variety of techniques, such as least-squared method [33], gradient descent technique [32], back-propagation algorithm of neural network[43], reinforcement learning [5], Tabu search algorithm [10], ant colony optimization [8], genetic algorithm [37] and others, had been used for the said purpose. However, all these methods have some limitations, such as (i) not all the methods are able to generate the globally optimal rules, as they may suffer from the local minima problem, (ii) when the number of rule increases in the rule base, not all the methods may provide with the feasible solution, (iii) computational complexity of some of those techniques may be quite high and thus, they may not be suitable for on-line implementations, (iv) most of these methods may generate some redundant rules, which will have a less importance to the solution of the problem. Thus, it is still an issue to search for a method, which will design a globally optimal Knowledge Base (KB) of an FLC within a reasonable time.

Neural networks (NNs) had also been utilized by some researchers for solving the motion planning problems of mobile robots. In this connection, work of Yang and Meng [46], Gu and Hu [15], Mondada and Floreano [29], Nolfi and Parsi [31], Yamada [45] are important to mention. However, the performance of an NN depends on its architecture and connecting synaptic weights, optimal selection of which is a tedious job. A variety of tools, such as supervised and reinforcement learning algorithms [5], Simulated Annealing (SA) [28], Genetic Programming (GP) [20], Genetic Algorithms (GA) [16] have been used by some investigators to improve the performance of NN. Out of these, the GA is found to be the widely accepted one. Moreover, the GA along with NN has added a new dimension to the field of robotic research, which is known as *evolutionary robotics* [35]. Here, a suitable NN architecture is evolved using a GA through its proper interactions with the environment. However, a GA is basically a fitness function-driven search tool and it is blind for any other aspect that is not explicitly considered in its fitness function. Thus, it may come up with an architecture of NN, which may not have a good generalization capability. So, there is still a scope for designing an efficient technique of NN optimization.

2.2 Environment Modeling

In an unknown and dynamic environment, motion planning depends on the information of the environment collected using the sensors/cameras. The choice of the sensors plays an important role in this regard. Sonar is the most widely used sensor for obstacle detection. Borenstein and Koren [6] used a sonar ring around their robot for obstacle detection. However, the main drawback of sonar lies in the fact that one sensor can provide with only one distance information. Other than the sonar sensor, cameras had also been used by some investigators. The cameras are passive sensors, which require an ambient light to illuminate its field of view. Moreover, the performance of the camera depends on some parameters, which are to be determined during calibration. Three types of camera calibration techniques are available in the literature, such as direct nonlinear optimization [11], closed form solution technique [1] and two-plane method [41, 44]. However, all such camera calibration methods have some limitations, which are mentioned below.

- Not in all the methods, lens distortion parameters were taken into account. Thus, the performance of the camera was not found to be good always.
- Most of these methods need a high amount of computation to find a solution. Thus, these techniques may not be suitable for on-line camera calibration.

Collected information through the cameras are to be interpreted properly to know the environmental changes. For which, we need a fast and noise sensitive image processing technique during motion planning. Tsukiyama and Huang [42] reported a scene interpretation approach for motion planning of autonomous vehicles. Lee and Sheen [23] proposed a model-based approach for this purpose. Cokal and Erden [9] developed a step-wise image processing method. However, there is a need for developing a suitable and fast image processing technique, which will be able to analyze the raw data and can provide with the desired information, such as obstacles' position and orientation.

2.3 Scope of the Chapter

The present chapter deals with the following issues:

- **Design of various robot motion planning approaches and performance testing through computer simulations:** An attempt has been made to develop the following approaches for solving the motion planning problems of a two-wheeled differential drive robot among some moving obstacles:

  1. Neuro-fuzzy system,
  2. Genetic-neuro-fuzzy system,
  3. Automatic design of neuro-fuzzy system using a GA,
  4. GA-tuned adaptive network-based fuzzy inference system,
  5. Genetic-fuzzy system,
  6. Back-propagation neural network,
  7. Genetic-neural system,
  8. Potential field approach.

  Comparison of these approaches has been made in terms of their computational complexities and performances through computer simulations. The best technique has been identified to solve the motion planning problems of a two wheeled mobile robot in the presence of some moving obstacles. It is important to mention that the motion of the robot will be planned after maintaining its both kinematic as well as dynamic constraints.
- **Motion planning of a real robot:** The performances of the best FL-based and NN-based approaches and potential field method has been tested through real experiments on a differential drive two-wheeled robot. A camera will be used to collect information of the dynamic environment. An on-line image processing scheme has been developed for the said purpose.

3 Statement of the Problem

Navigation problem of a two-wheeled differential drive robot has been studied in the present chapter. The motion of the robot is restricted due to its kinematic, dynamic constraints and turning radius of the robot is lower bounded due to the limits on the steering angle. Moreover, the robot's motion is obstructed by the partially-unknown movement of the moving obstacles. Hence, the robot will have to find its collision-free path, while navigating among some moving obstacles after satisfying its kinematic and dynamic constraints. Depending on the position, size and velocity of the obstacles, the robot may find a number of collision-free paths. However, the main aim of this study is to identify that particular path, which is not only collision-free but also time-optimal one. Therefore, the present problem can be treated as a constrained optimization problem.

To reduce the complexity of the present problem, all the moving obstacles have been represented by their bounding circles and at a time, only one obstacle is treated to be critical. Moreover, the wheels of the robot are assumed to move due to pure rolling action only and Coriolis component of the force is not taken into consideration in the present study. It is important to mention that in actual motion planning, the robot plans its path based on collected information from the environment using some sensors and/or cameras, on-line. However, in computer simulations, the motion of the robot is planned based on predicted position ($P_{predicted}$) of the obstacles after the time step ΔT, which may be determined by linearly extrapolating from its present ($P_{present}$) and previous ($P_{previous}$) positions as given below.

$$P_{predicted} = P_{present} + (P_{present} - P_{previous}) \quad (1)$$

3.1 Proposed Motion Planning Scheme and Mathematical Formulation of the Problem

The proposed motion planning scheme of the robot is explained with the help of Fig. 1. The robot starts its motion from a predefined starting position and reaches the goal by avoiding collision with several moving obstacles. The total path of the robot is assumed to be a collection of some small segments, each of which is traversed during a fixed time ΔT. The path followed by the robot during this small time ΔT is either a straight or a curved one. Before the starting of a step, if the robot finds any critical obstacle ahead of it, the motion planner is activated. Otherwise, the robot moves towards the goal in a straight path with a maximum possible velocity. If the robot starts its motion from rest, it is assumed that an additional $\frac{\Delta T}{4}$ time is required to align its main axis towards its future direction of movement. The outputs of the motion planner are considered to be the acceleration of the robot and deviation necessary to avoid collision with the most critical obstacle. Moreover, if required, the deviation provided by the motion planner is to be corrected using a collision-avoidance scheme. Again, if the generated motion of the robot violates its kinematic and/or dynamic constraints, the robot is stopped for ΔT time at its present position. Thus, there will be an additional ΔT time per such occasion. This process will continue, until the robot reaches its destination. It is important to mention that the last time step (T_{rem}) may not be a complete one and it depends on the distance left uncovered (d_{goal}) by the robot. If the distance between the robot and its goal (d_{goal}) comes out to be less than a predefined minimum distance (d_{min}), it starts decelerating and stops at the goal. The total traveling time T is then calculated by adding all intermediate steps needed by the robot to reach its destination. The aim of this study is to design a suitable motion planner, so that the robot will be able to reach its destination with the lowest possible traveling time by avoiding collision

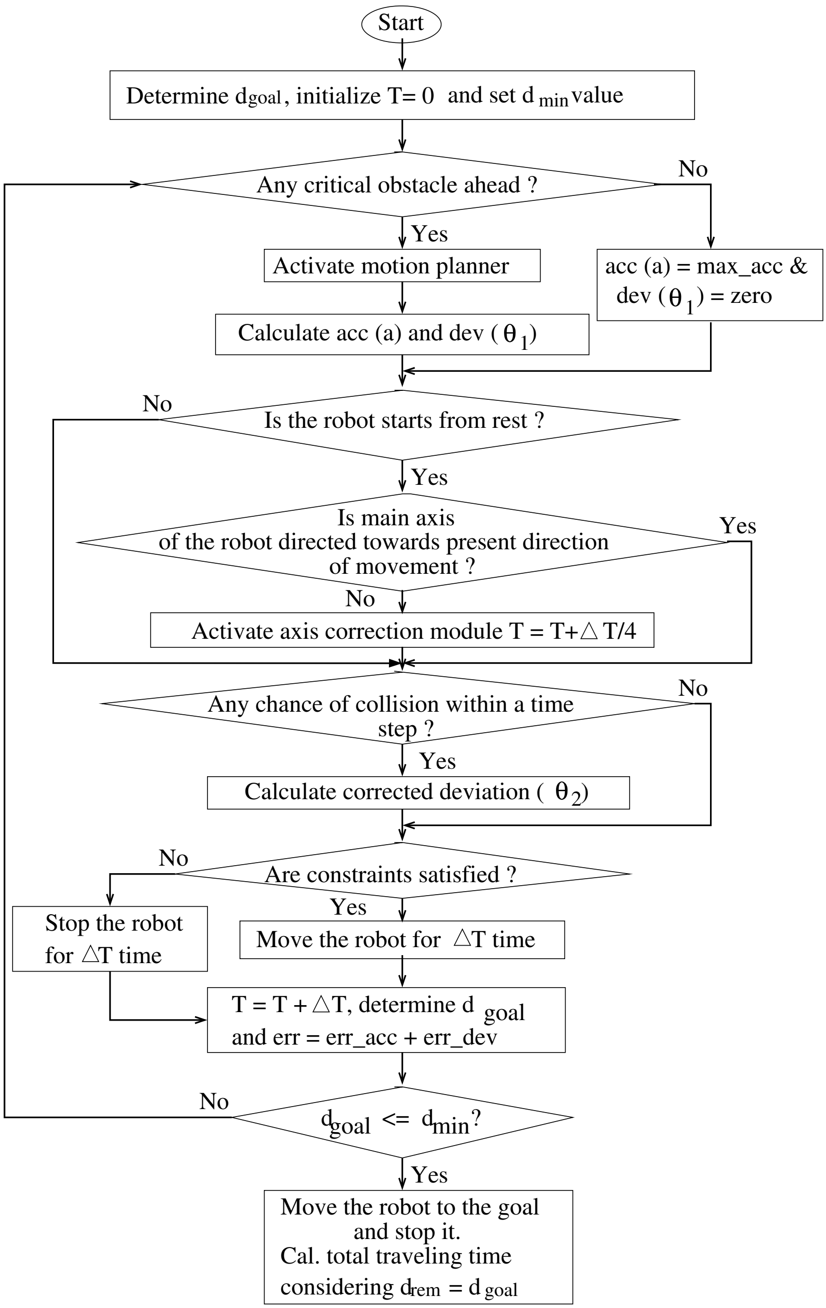


Fig. 1 A schematic view showing the flowchart of the developed motion planning scheme

with the obstacles. Moreover, it will have to satisfy the following kinematic and dynamic constraints, as given below.

Kinematic constraints
The following kinematic constraints are to be satisfied by the robot:

(i) $-\dot{X}cos\theta + \dot{Y}sin\theta = 0,$
(ii) $(\dot{X})^2 + (\dot{Y})^2 - (\rho_{min}\dot{\phi})^2 \geq 0,$

where $\dot{X}$ and $\dot{Y}$ are the components of tangential velocity along +ve X-axis and +ve Y-axis, respectively and θ is the angle between the X-axis and main axis of the robot. The minimum radius of curvature is represented by ρ_{min} and $\dot{\phi}$ denotes the rate of change of steering angle during turning.

Dynamic constraints
During navigation, the robot will have to follow the dynamic constraints given below.

(i) $-\sqrt{(\mu_f g)^2 - (v\dot{\phi})^2} \leq a \leq \sqrt{(\mu_f g)^2 - (v\dot{\phi})^2},$
(ii) $a \geq \frac{60P}{2\pi r \times GR \times M \times N_m},$
(iii) $v \geq \rho_{min}\dot{\phi},$

where v and a denote the tangential velocity and acceleration of the CG of the robot, respectively and the power required by the motor to create maximum angular speed N_m is represented by the term P. Moreover, GR indicates the gear ratio of the wheels, r is the radius of the wheels of the robot and the mass of the robot is denoted by M. Again, μ_f indicates the coefficient of friction between the wheels and the surface of the terrain and acceleration due to gravity is represented by g.

Therefore, the present problem may be treated as a constrained traveling time (T) minimization problem as stated below.

$$Minimize\ T = \sum_{i=1}^{U} \Delta T + \sum_{j=1}^{U'} \frac{\Delta T}{4} + T_{rem}, \qquad (2)$$

subject to

- The path is collision-free;
- The constraints of the robot are not violated.

Here, U represents the number of complete time steps and U' indicates the number of times the robot is required to take turn to align its axis with the direction of its goal.

4 Developed Motion Planning Approaches

Eight different approaches have been developed for solving the motion planning problems of a two-wheeled differential drive robot among moving obstacles [17]. All these approaches are discussed below, in brief.

Approach 1: Neuro-Fuzzy System
In this approach, a Fuzzy Logic Controller (FLC) [27] has been expressed using the structure of a Neural Network (NN) and a back-propagation algorithm has been utilized to optimize its Knowledge Base (KB). Fig. 2 shows the schematic view of the

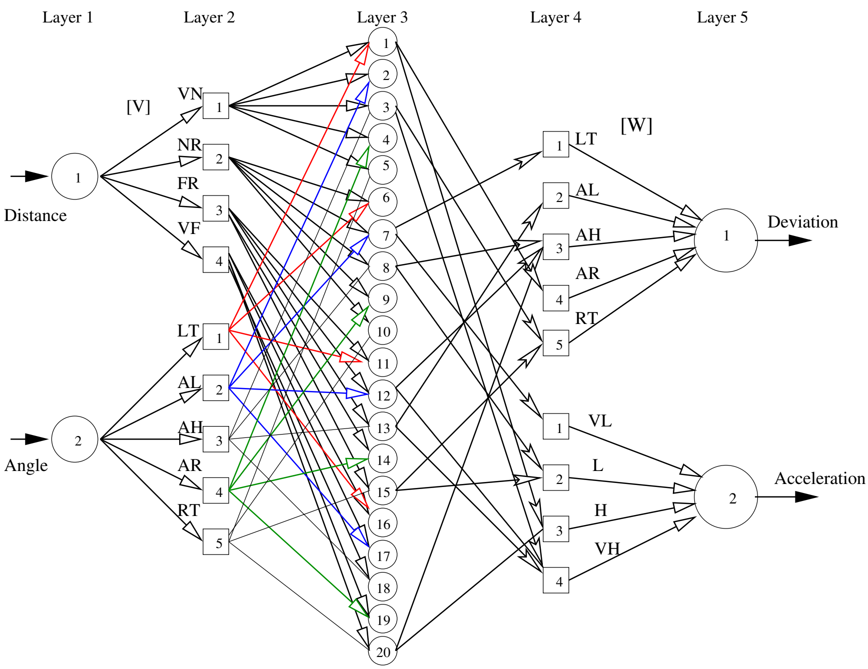


Fig. 2 A schematic view of the neural network-structured FLC

developed neuro-fuzzy system, which consists of five layers - Layer 1 is the input layer, fuzzification is done in Layer 2, Layer 3 indicates AND operation and OR operation is performed in Layer 4 and the next layer is the output layer. The initial weights of the NN carrying information of the membership function distributions of the FLC are generated, at random. A batch mode of training is adopted. The training cases are passed through the NN (i.e., forward propagation) and the average error in prediction is determined. As this error depends on the weights also, it can be minimized by updating the weight values. A back-propagation algorithm is used to minimize the error. It is to be noted that the membership function distributions of the FLC are taken to be triangular in nature and a manually constructed rule base is utilized during the optimization.

Approach 2: Genetic-Neuro-Fuzzy System
In Approach 1, the error in prediction is minimized using a steepest descent method, which may suffer from the local minima problem. To overcome this problem, the back-propagation algorithm of Approach 1 is replaced by a GA in Approach 2.

Moreover, the rule base of the FLC is not optimized in Approach 1. In this approach, both the rule base as well as membership function distributions are optimized simultaneously using a GA. It is to be noted that there might be some redundant rules present in the GA-designed rule base. It may happen due to the iterative nature of the GA. To identify the redundant rules (if any), a method is proposed, in which importance of a rule is calculated considering its frequency of occurrence and worth with respect to the objectivity of the problem. Based on the value of this importance factor, decision is taken whether a particular rule will be declared a redundant one and can be eliminated from the GA-designed rule base. Thus, Approach 2 is expected to perform better than Approach 1.

Approach 3: Automatic Design of Neuro-Fuzzy System
To increase the search space of GA, a method for automatic design of neuro-fuzzy system is proposed. In this approach, the outputs of different rules are evolved by the GA itself and no effort is made to design the rule base manually. The GA through its exhaustive search, determines a good rule base of the FLC. The redundant rules are identified and eliminated from the GA-designed rule base by following a procedure similar to that of Approach 2.

Approach 4: GA-tuned ANFIS
In this approach, a Takagi-Sugeno type neuro-fuzzy controller [40] has been developed. It is popularly known as Adaptive Network-based Fuzzy Inference System (ANFIS). Fig. 3 shows the schematic view of an ANFIS structure, which consists of six layers. First two layers of ANFIS perform the tasks similar to that of Layers 1 and 2 of Approach 1. Layer 3 is known as rule base layer, in which the rules are defined. Each neuron corresponding to this layer represents a fuzzy rule and is termed as a rule node. Layer 4 consists of the same number of nodes as considered in the previous layer and the normalized firing strength of each node is calculated in this layer. Layer 5 is termed as a consequence layer. Here, the output of a neuron is determined using a functional relationship with the inputs of the first layer. In case of first order Takagi-Sugeno's method, the outputs are considered to have linear relationship with the inputs of the first layer. Finally, the outputs of the controller are obtained in Layer 6. It is to be noted that the performance of the present approach is improved by optimizing both its antecedent and consequent parameters using a GA.

Approach 5: Genetic-Fuzzy System
In genetic-fuzzy system, a GA is utilized to obtain an optimal KB of the FLC. Instead of manually designing the rule base, a method for automatic design is considered using a GA. Redundant rules are identified and eliminated following the similar method as considered in Approaches 2 and 3. Fig. 4 shows the schematic view of the genetic-fuzzy system. The GA begins its search by randomly creating a number of solutions (equals to the population size) represented by binary strings, each of which indicates a typical FL/NN-based motion planner. Each solution in the population is then evaluated to assign a fitness value and modified using three operators: reproduction, uniform crossover and bit-wise mutation. One iteration involving

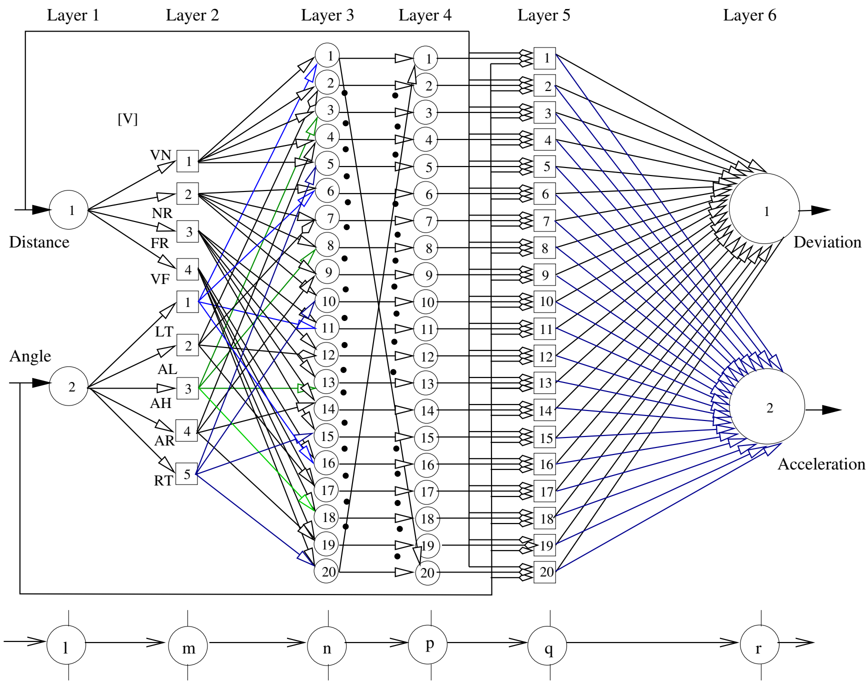


Fig. 3 A schematic view showing the structure of an ANFIS

these three operations followed by the fitness evaluation, is called a generation. Generations proceed until a termination criterion is satisfied. In this approach, the GA has been allowed to run for a pre-specified number of generations.

Approach 6: Back-propagation Neural Network

The said problem has also been solved using a three-layered neural network, whose architecture is shown in Fig. 5. It contains three layers, such as input, hidden and output layers. Signal-flow through the network progresses in a forward direction, from the left to right. The activation functions in each layer are considered to be tangent hyperbolic and a fixed value of bias has been assumed for each neuron. In this approach, the synaptic weights are updated using a back-propagation algorithm. Several combinations of learning parameters and number of hidden neurons are tried to get the best convergence of the network. It is to be noted that during learning, the updated synaptic weights may come out of their ranges. In such a situation, those particular weights are penalized using some continuous penalty function.

Approach 7: Genetic-Neural (GA-NN) System

Realizing the fact that it is difficult to develop an NN through an explicit design, researchers working in this field started thinking whether it can be evolved using an evolutionary technique. Simultaneous optimization of the weights and architecture

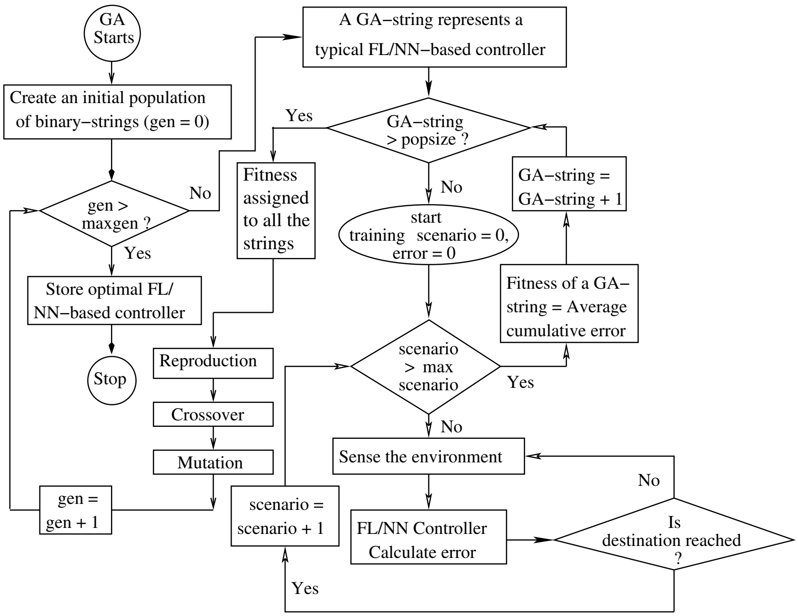


Fig. 4 A schematic view showing the working principle of the genetic-fuzzy/neural system

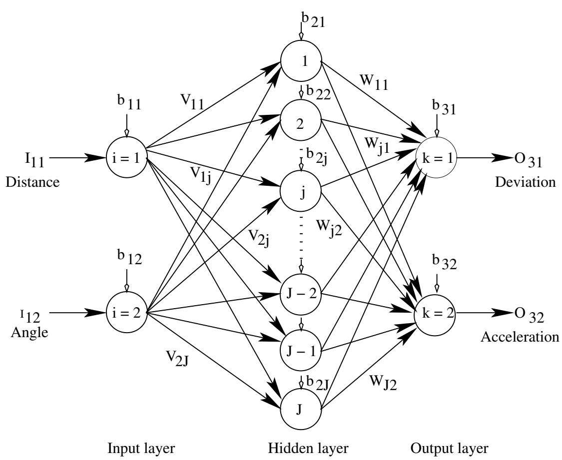


Fig. 5 A schematic view of the neural network structure

of an NN is addressed in this approach using a GA. Fig. 4 shows the working principle of the combined GA-NN approach.

Approach 8: Potential Field Method

Potential field method introduced by Khatib [19], has been widely used for real-time collision-free path planning of manipulators, mobile robots. In this approach, the robot is modeled as a particle moving under the influence of artificial potential fields created by a set of obstacles and the goal. The goal is assumed to have attractive potential and the obstacles generate repulsive potentials. The movement of the robot is then determined using the resultant force obtained due to these two potential fields. It is important to note that acceleration of the robot has been taken to be proportional to the magnitude of the resultant force and deviation is considered as the angle made between the direction of the resultant force and a new reference line joining the CG of the robot at the present time step and the goal position.

5 Results and Discussion

The performances of the developed motion planning approaches are studied through both computer simulations as well as real experiments for solving the motion planning problems of a two-wheeled differential drive robot among moving obstacles. Initially, all the eight approaches are compared among them through computer simulations, in terms of computational complexity, traveling time and adaptability. From this study, the best FL-based and NN-based approaches have been identified. Thereafter, the performances of these two soft computing-based approaches and a potential field-based approach are studied on a real two-wheeled differential drive robot.

5.1 Performance Testing through Computer Simulations

The performances of all the developed eight approaches have been compared through computer simulations for solving the motion planning problems of a two-wheeled differential drive robot in the presence of some moving obstacles. The robot is allowed to navigate in a grid of $19.95 \times 19.95m^2$ among sixteen moving obstacles. The time interval (ΔT) is taken to be equal to sixteen seconds and the robot is assumed to have a maximum and minimum accelerations of $0.05m/s^2$ and $0.005m/s^2$, respectively. Two hundred training scenarios are generated at random for the training purpose. A particular training scenario is different from the other, in terms of initial positions of the obstacles and their sizes, speeds and directions of movement. After the training is over, the performances of seven soft computing-based approaches and potential field method have been studied for twenty test scenarios (selected at random), which are not included in the training set. The values of traveling time taken by the robot using all these approaches, while moving among sixteen obstacles are shown in Fig. 6. A particular test scenario (say 4^{th}) is shown in Fig. 7, where the movements of both the obstacles as well as the robot are shown. From this study, the following observations have been made:

- In most of the test cases, the time taken by the robot using the genetic-fuzzy approach (i.e., Approach 5) is the least.

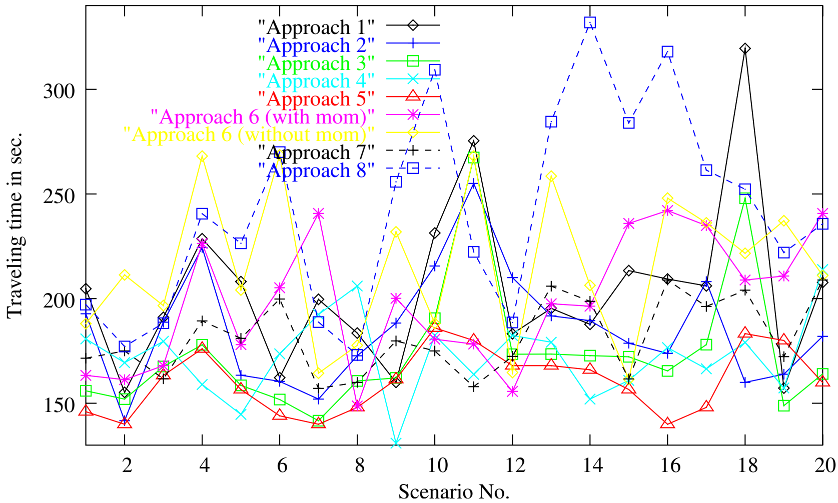


Fig. 6 Comparison of eight approaches in terms of traveling time

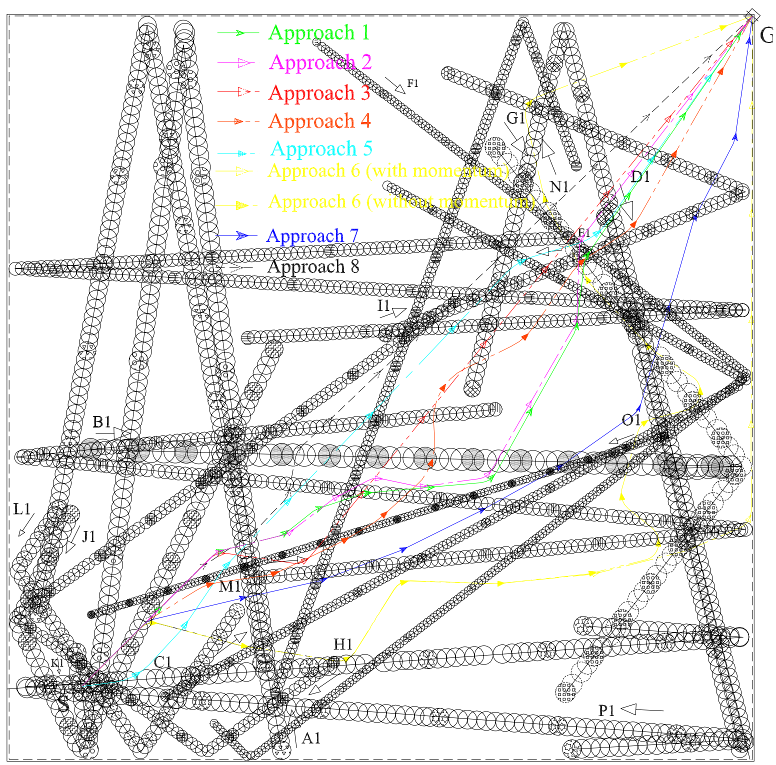


Fig. 7 Collision-free paths obtained by the robot using eight different approaches (4th test scenario)

- In some critical situations, potential field method may not provide with any feasible solution, whereas soft computing-based approaches are able to tackle those situations well.
- Potential field method is found to be the fastest of all but soft computing-based approaches are also seen to be computationally tractable. Thus, they are suitable for on-line implementations.

5.2 Camera Calibration and Image Processing

Experiments are carried out on a real two-wheeled differential drive robot as shown in Fig. 8 (Model: Soccer Robot, Make: Microrobot Co. Ltd., Seoul, Korea) and an overhead CCD camera is used to collect information of the environment. The positions and orientations of the objects moving on a flat terrain are obtained by taking images at successive time interval using the camera. However, the performance of the camera depends on its parameters, which are to be calibrated first to ensure good results. A suitable image processing algorithm is also essential to extract necessary information from the image. In this study, an attempt is made to develop a camera calibration method, in which both distortion of lens as well as computer frame uncertainty factor are considered in the camera model. The problem of camera calibration has been posed as an optimization problem and solved using a binary-coded GA. Moreover, a suitable and fast image processing technique has also been adopted in this study.

Experiments are conducted with the help of an experimental set-up shown in Fig. 9. The following steps are adopted to carry out the experiment:

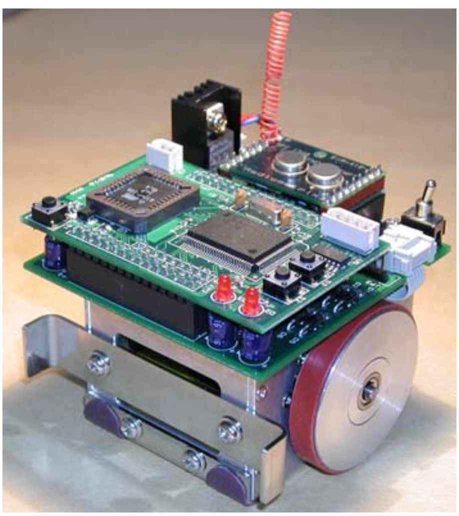


Fig. 8 Photograph of the robot used in the experiment (Make: Micro Robot NA, South Korea)

Fig. 9 A schematic view showing the experimental setup

- **Camera Calibration:** The performance of a CCD camera depends on some of its internal and external parameters. A binary-coded GA is used in the present study to calibrate it, before it can be used on-line. It is to be mentioned that the calibration is done off-line.
- **On-line Image Processing:** The images captured using the camera are analyzed by developing a suitable image processing method, in which the following steps are considered: (i) removal of noises using a 3x3 median filter, (ii) binarize the images by means of a threshold value, (iii) estimation of perimeters, area with the help of a perimeter descriptor, (iv) labeling of the objects, so as to identify their entity properly, (v) removal of some of the extraneous components based on their sizes.
- **Control of the robot:** Based on the outputs of the motion planner, the angular speeds of the wheels are derived and the motors are controlled using a PD control law, which are then communicated to the robot by means of a Radio-Frequency (RF) module. Finally, movement of the robot takes place with the help of two separately controlled differential drive DC motors.

5.3 Performance Testing through Real Experiments

The present chapter deals with motion planning problems of a real two-wheeled differential drive robot in the presence of some moving obstacles. Three motion planning approaches (which are found to provide with the better results in simulations), namely genetic-fuzzy system, genetic-neural system and potential field method (let us call them Approaches A, B and C, respectively) have been developed for this purpose. It is to be noted that Approaches 5, 7 and 8 of Section 4 have been renamed as Approaches A, B and C, respectively, for carrying out the experiments. Soft Computing (SC)-based approaches are tuned using a GA, off-line and the performances of the optimized motion planners are tested through real experiments. The time interval (ΔT) for conducting the experiments is taken to be equal to one second. Experiments are carried out for a case, where the robot is allowed to navigate among two moving obstacles.

After the training is over, the effectiveness of the SC-based approaches are compared among them and with that of the potential field method for four test scenarios. Table 1 shows the traveling time taken by the real robot using three approaches. The

Table 1 Comparison of three approaches in terms of traveling time in seconds

Scen.	Traveling time in seconds		
No.	Approach A	Approach B	Approach C
1	37	36	40
2	39	42	47
3	42	37	57
4	40	36	48

performance of Approach B is found to be better than that of Approach A in three out of four test scenarios. Moreover, in all the scenarios, Approaches A and B have performed better than Approach C. It may be due to the fact that there is no in-built optimization module in Approach C. For the 1st test scenario of Table 1, positions of the robot and obstacles at eight different instants of time are shown in Figs. 10, 11 and 12 for Approaches A, B and C, respectively. Moreover, the movement of the robot following all three approaches along with the movement of the obstacles are shown in Fig. 13 for the Scenarios 1, 2, 3 and 4 of Table 1. It is interesting to note that the path generated by Approach C is found to be straight, in most of the scenarios. It could be due to fact that the robot in Approach C is stopped for most of the times to avoid collisions with the most critical obstacle and it allows the obstacle to pass by. Thus, it is unable to generate the feasible motion of the robot for most of the times. It is also to be noted that the robot has taken left turn in most of the situations to avoid collisions with the critical obstacle in Approach A, as the GA-designed optimal rule base contains the deviation output to be ahead left (AL) in most of the rules.

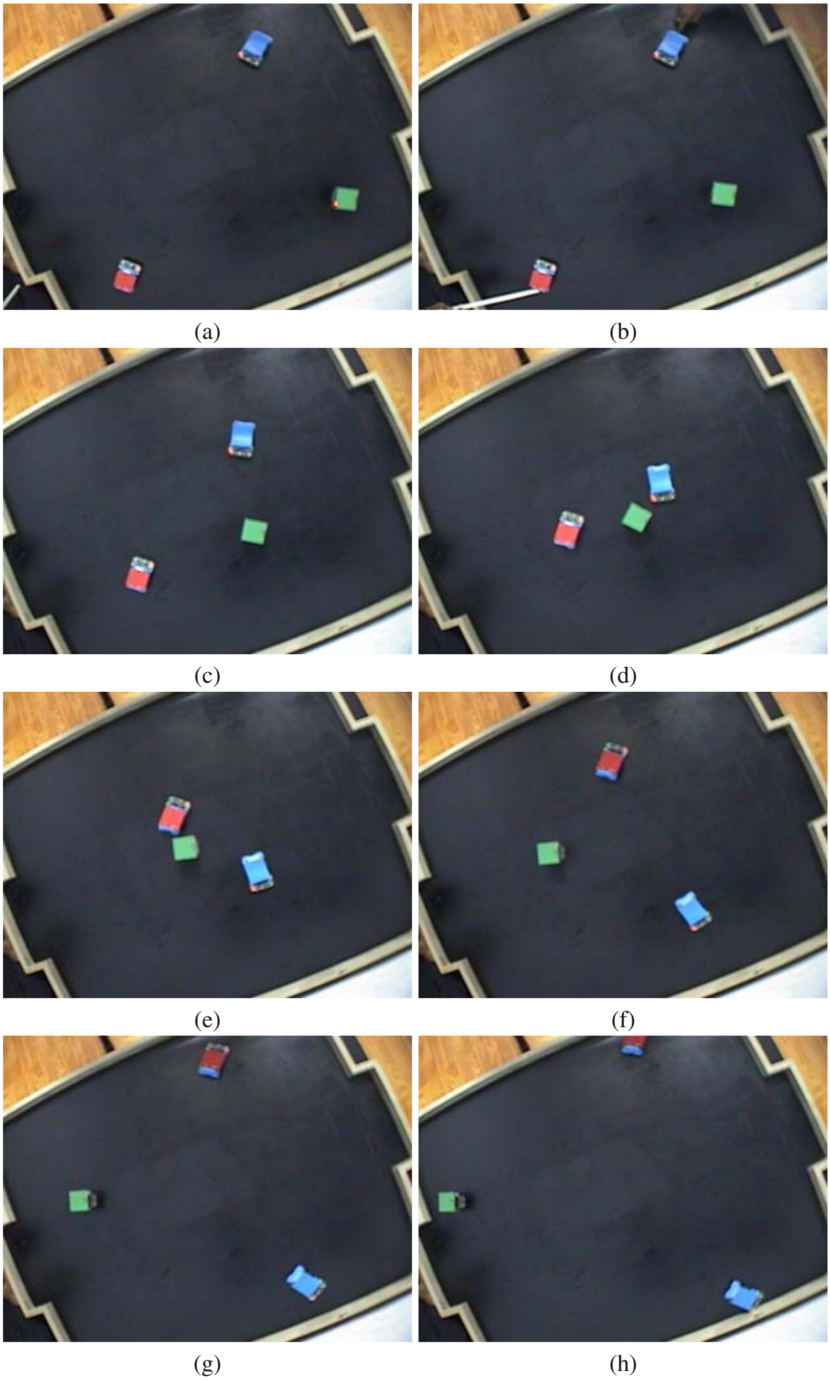


Fig. 10 Positions of the robot moving among two moving obstacles at eight instants of time – Approach A, Scenario 1 of Table 1

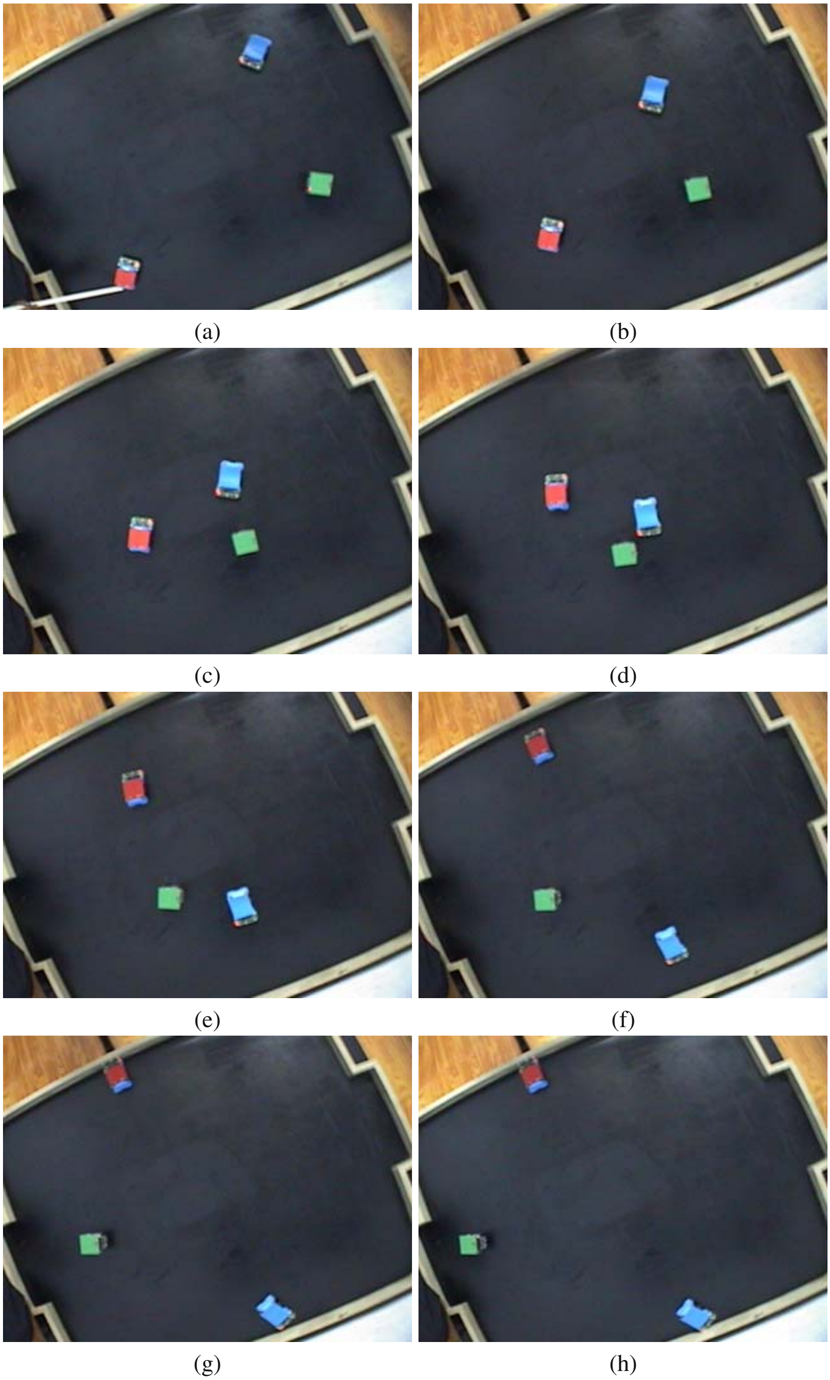


Fig. 11 Positions of the robot moving among two moving obstacles at eight instants of time – Approach B, Scenario 1 of Table 1

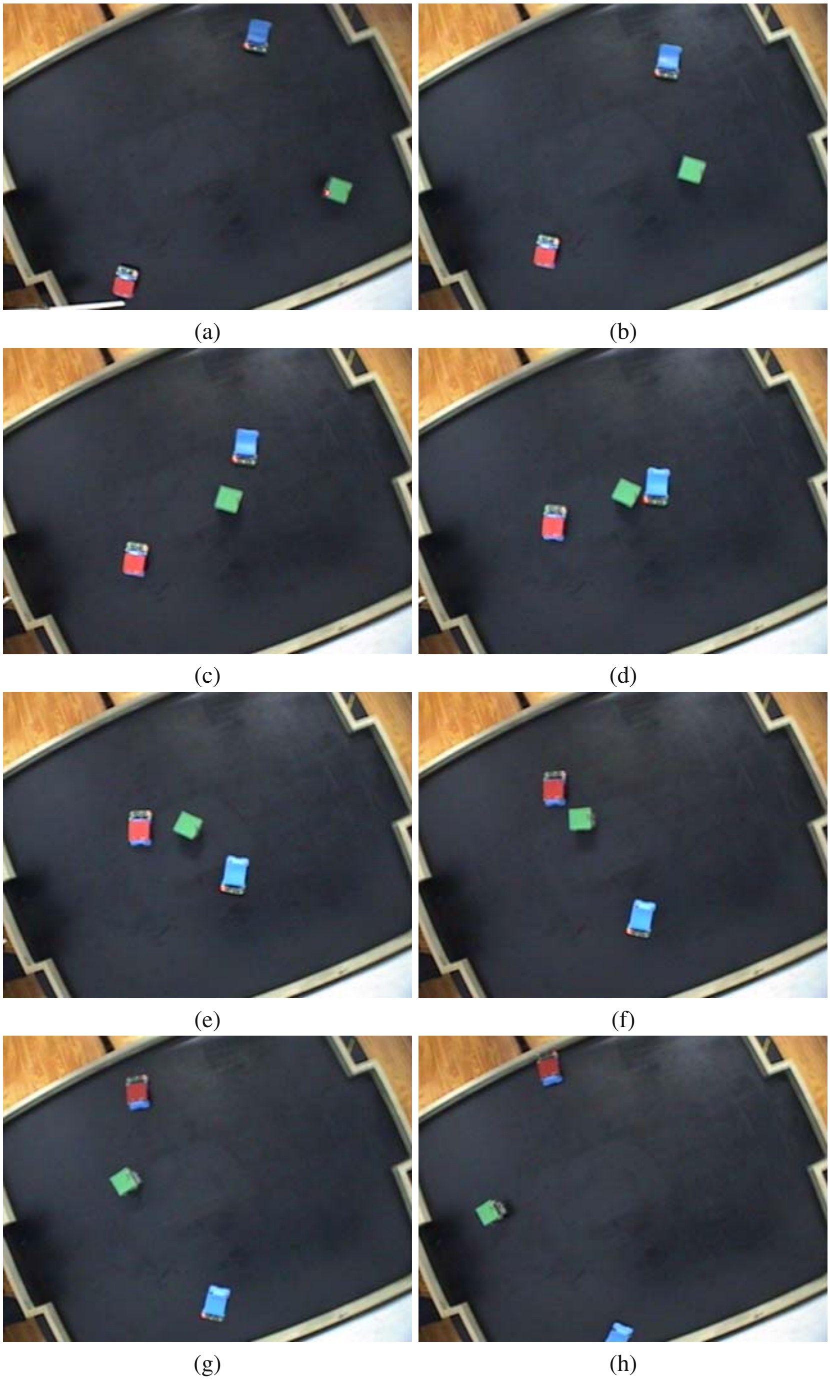
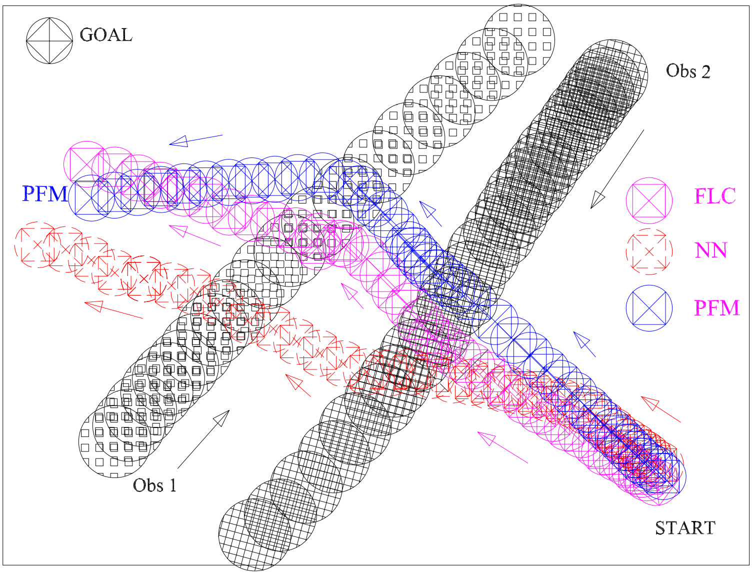
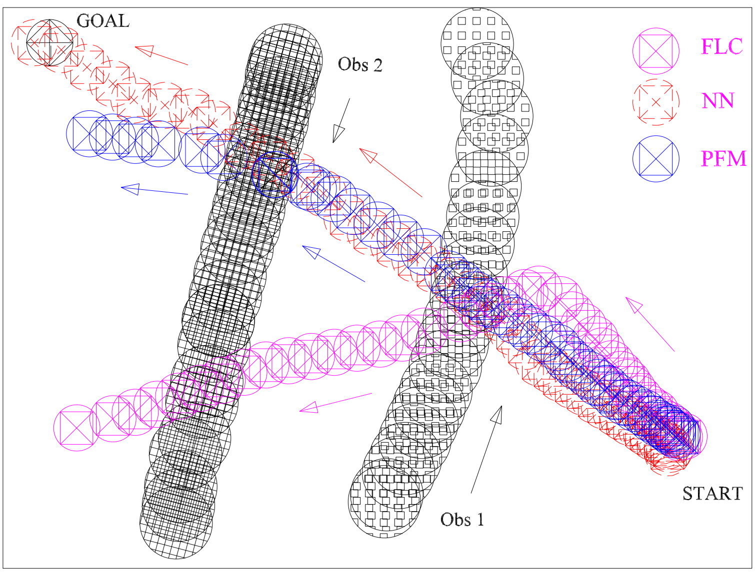


Fig. 12 Positions of the robot moving among two moving obstacles at eight instants of time – Approach C, Scenario 1 of Table 1

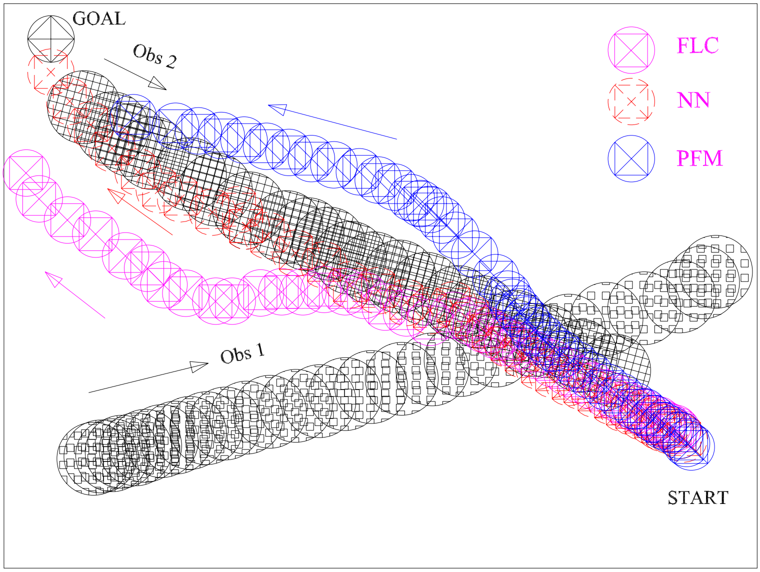


(a)

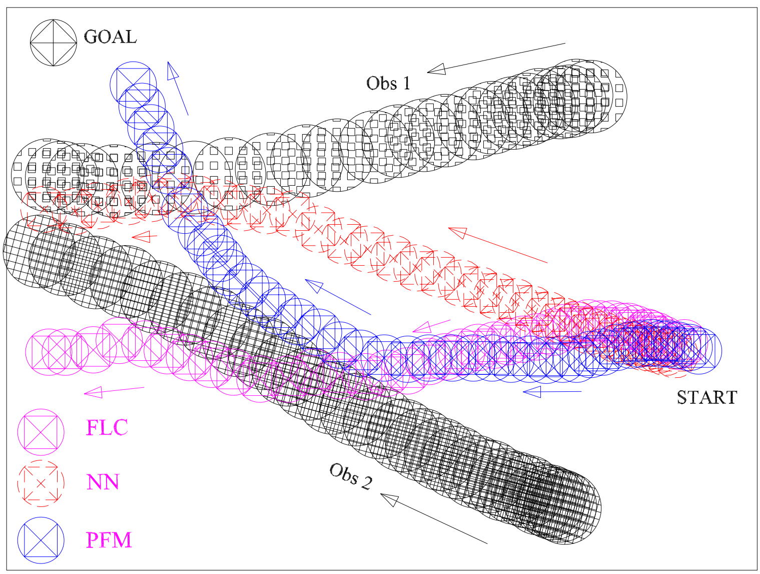


(b)

Fig. 13 Movement of the robot among two moving obstacles: (a) Scenario 1, (b) Scenario 2, (c) Scenario 3 and (d) Scenario 4 of Table 1



(c)



(d)

Fig. 13 (*continued*)

6 Concluding Remarks and Scope for Future Work

In the present chapter, motion planning problems of a two-wheeled differential drive robot navigating among some moving obstacles have been solved. A few soft computing-based approaches and a conventional potential field method have been developed for the said purpose. Initially, the performances of all the approaches have been studied through computer simulations. Thereafter, real experiments are conducted to test the effectiveness of various motion planning approaches. A camera-based vision system has been used in the real experiments to collect information of the environment. From the above study, some concluding remarks have been made as follows:

- There is an acute need of an intelligent and autonomous decision maker for solving the motion planning problems of mobile robots navigating in a dynamic environment.
- In Potential Field Method (PFM), the attractive force decreases linearly with the distance between the robot and its goal. Thus, the motion planner developed based on this method is unable to yield a high value of acceleration, when the robot comes closer to the goal, irrespective of the obstacles' position in the environment. Consequently, the shortest distance path generated by the PFM may not always be the time-optimal one. Moreover, there is a chance of the solutions of potential field method for being trapped into local minima, particularly when the robot faces any concave obstacle.
- In some occasions, the robot has failed to find any feasible solution using the PFM. This may happen, when the repulsive potential of the obstacle balances the attractive potential of the goal. It is known as the dead-lock situation.
- Although the genetic-neural system sometimes has generated the longest distance path, its performance in terms of traveling time is not found to be the worst, due to the relatively higher speed of the robot during its movement.
- Computational complexity of all the developed motion planning approaches is found to lie within an acceptable limit. Thus, they are suitable for on-line implementations.
- The performances of both the soft computing-based approaches are found to be comparable in the real experiments. However, traveling time taken by the robot using the potential field method has come out to be the maximum in most of the cases. It might have happened due to the reason that there is no in-built optimization module in the PFM.
- Soft computing-based approaches are found to be more robust and adaptive compared to the conventional PFM. It may be due to the fact that the PFM does not have any in-built learning module. All the parameters used in this approach are static in nature and do not adapt automatically, as the situation changes. Moreover, all the approaches are found to be reasonably good in terms of repeatability and goal reaching capability.

Therefore, the SC-based motion planning approaches have come out to be promising ones for the development of intelligent and autonomous robots. However, design and development of a suitable SC-based motion planner is not an easy task. Once optimized, they will perform in an optimal sense and provide with some feasible solutions in an adaptive way, on-line.

6.1 Scope for Future Work

The present study may be extended in a number of ways. Some of these are mentioned below.

- In the present study, an attempt is made to solve the motion planning problems of a two-wheeled differential drive robot navigating among moving obstacles. For this purpose, a few soft computing-based approaches have been developed. Training to the SC-based approaches are provided off-line, whereas the training can also be given on-line, in which the motion planner will learn during its course of action. Thus, the performances of the motion planner may not be good initially, but they are expected to perform better with the time. The method of on-line training may be implemented in future.
- Only one planning robot has been taken into consideration in the present study. However, it will be more interesting to study the coordination issues of multiple mobile robots working in the common environment.

Acknowledgements. A part of this work is supported by the Department of Science and Technology, Govt. of India, New Delhi (Sanction No. SR/S3/RM/28/2003 dt. 12.12.2003). The first author also gratefully acknowledges the financial support of the SRIC, IIT Kharagpur, India, for providing him with the High Value Ph.D. Fellowship.

References

1. Abdel-Aziz, I.I., Karara, H.M.: Direct linear transformation into object space coordinates in close-range photogrammetry. In: Proceedings Symposium Close-Range Photogrammetry, University of Illinois at Urbana Champaign, Urbana, pp. 1–18 (1971)
2. Akbarzadeh, M., Kumbla, K., Tunstel, E., Jamshidi, M.: Soft computing for autonomous robotic systems. Computers and Electrical Engineering 26, 5–32 (2000)
3. Alami, R., Chatila, R., Fleury, S., Ghallab, M., Ingrand, F.: An architecture for autonomy. Special issue on: Integrated architectures for robot control and programming 17(4) (1998)
4. Asada, M., Kitano, H., Noda, I., Veloso, M.: Robocup: today and tomorrow - what we have learned. Artificial Intelligence 110, 193–214 (1999)
5. Berenji, H.R.: Learning and tuning fuzzy logic controllers through reinforcements. IEEE Transactions on Neural Networks 3, 724–740 (1992)
6. Borenstein, J., Koren, Y.: Real time obstacle avoidance for fast mobile robots. IEEE Transactions on Systems Man and Cybernetics 19(5), 1179–1187 (1989)
7. Borenstein, J., Koren, Y.: The vector field histogram – fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation 7(3), 278–288 (1991)
8. Casillas, J., Cordon, O., Herrera, F.: Learning fuzzy rules using ant colony optimization algorithms. In: Proceedings of 2nd International Workshop on Ant Algorithms, Brussels, Belgium, pp. 13–21 (2000)

9. Cokal, E., Erden, A.: Development of an image processing system for a special purpose mobile robot navigation. In: Proceedings of Fourth Annual Conference on Mechatronics and Machine Vision Practice, Toowoomba, Australia, pp. 246–252 (1997)
10. Denna, M., Mauri, G., Zanaboni, A.M.: Learning fuzzy rules with tabu search - an application to control. IEEE Transactions on Fuzzy Systems 7(2), 295–318 (1999)
11. Faig, W.: Calibration of close-range photogrammetry systems: mathematical formulation. Photogrammetric Engineering and Remote Sensing 41, 1479–1486 (1975)
12. Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using the relative velocity paradigm. In: Proceedings of IEEE Conference on Robotics and Automation, pp. 560–565 (1993)
13. Fraichard, T., Garnier, P.: Fuzzy control to drive car-like vehicles. Robotics and Autonomous Systems 34, 1–22 (2001)
14. Fujimura, K., Samet, H.: Accessibility: a new approach to path planning among moving obstacles. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Ann Arbor, MI, pp. 803–807 (1988)
15. Gu, D., Hu, H.: Neural predictive control for a car-like mobile robot. Robotics and Autonomous Systems 39, 73–86 (2002)
16. Holland, J.H.: Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor (1975)
17. Hui, N.B.: Design and development of adaptive robot motion planner for wheeled robots. Ph.D. thesis, Indian Institute of Technology, Kharagpur (2008)
18. Kant, K., Zucker, S.W.: Towards efficient planning: the path velocity decomposition. International Journal of Robotics Research 5(1), 72–89 (1986)
19. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research 5(1), 90–98 (1986)
20. Koza, J.R.: Genetic Programming: on the programming of computers by means of natural selection. The MIT Press, Cambridge (1992)
21. Lamadrid, J.G., Gini, M.L.: Path tracking through uncharted moving obstacles. IEEE Transactions on Systems, Man and Cybernetics 20(6), 1408–1422 (1990)
22. Latombe, J.C.: Robot motion planning. Kluwer Academic Publishers, Dordrecht (1991)
23. Lee, P.S., Shen, Y.E.: Model-based location of automated guided vehicles in the navigation sessions by 3d computer vision. Journal of Robotic Systems 11(3), 181–195 (1994)
24. Leven, D., Sharir, M.: Planning a purely translational motion for a convex object in two-dimensional space using generalized voronoi diagrams. Discrete Computational Geometry 2, 9–31 (1987)
25. Liu, Y.H., Arimoto, S.: Path planning using a tangent graph for mobile robots among polynomial and curved obstacles. International Journal of Robotics Research 11(4), 376–382 (1992)
26. Mackworth, A.: On seeing robots. In: Computer Vision: Systems, Theory and Applications, pp. 1–13. World Scientific Press, Singapore (1993)
27. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7, 1–13 (1975)
28. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state claculations by fast computing machines. Journal of Chemical Physics 21, 1087–1092 (1953)
29. Mondada, F., Floreano, D.: Evolution of neural control structures: some experiments on mobile robots. Robotics and Autonomous Systems 16, 183–195 (1995)
30. Nilsson, N.J.: A mobile automation: an application of artificial intelligence techniques. In: Proceedings of the 1st International Joint Conference on Artificial Intelligence (IJCAI), Washington, DC, pp. 509–520 (1969)

31. Nolfi, S., Parsi, D.: Learning to adapt to changing environments in evolving neural networks. Adaptive Behavior 5(1), 75–98 (1997)
32. Nomura, H., Hayashi, I., Wakami, W.: A learning method of fuzzy inference rules by descent method. In: Proceedings of IEEE International Conference Fuzzy Systems, San Diego, CA (1992)
33. Pham, T.D., Valliappan, S.: A least squares model for fuzzy rules of inference. Fuzzy Sets and Systems 64, 207–212 (1994)
34. Pratihar, D.K.: Algorithmic and soft computing approaches to robot motion planning. Machine Intelligence and Robotic Control 5(1), 1–16 (2003)
35. Pratihar, D.K.: Evolutionary robotics - a review. Sadhana 28(6), 999–1003 (2003)
36. Pratihar, D.K.: Soft computing. Narosa Publishing House, New Delhi (2008)
37. Pratihar, D.K., Bibel, W.: Near-optimal, collision-free path generation for multiple robots working in the same workspace using a genetic-fuzzy systems. Machine Intelligence and Robotic Control 5(2), 45–58 (2003)
38. Slack, M.: Navigation template: mediating qualitative guidance control in mobile robots. IEEE Transactions on Systems, Man, Cybernetics 23(2), 452–466 (1993)
39. Svestka, P., Overmars, M.: Motion planning for car-like robots using a probabilistic learning approach. International Journal of Robotics Research 16(2), 119–143 (1997)
40. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 15, 116–132 (1985)
41. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 2d machine vision meteorology using off-the-shelf tv cameras and lenses. IEEE Journal of Robotics and Automation 4, 323–344 (1987)
42. Tsukiyama, T., Huang, T.: Motion stereo for navigation of autonomous vehicle in man-made environments. Pattern Recognition 20(1), 105–113 (1987)
43. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. IEEE Transactions on Systems, Man and Cybernetics 22(6), 1414–1422 (1992)
44. Weng, J., Cohen, P., Herniou, M.: Camera calibration with distortion models and accuracy evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(10), 965–980 (1992)
45. Yamada, S.: Evolutionary behavior learning for action based environment modeling by a mobile robot. Applied Soft Computing 5, 245–257 (2005)
46. Yang, S.X., Meng, M.: An efficient neural network approach to dynamic robot motion planning. Neural Networks 13(2), 143–148 (2000)

4

Gait Planning of Biped Robots Using Soft Computing: An Attempt to Incorporate Intelligence

Pandu Ranga Vundavilli and Dilip Kumar Pratihar

Abstract. This chapter deals with the issues related to incorporation of intelligence using soft computing to biped robots moving on some uneven terrains, such as staircases, sloping surfaces, ditches, and others. The said problems have been solved utilizing both analytical as well as soft computing-based approaches. In the analytical approach, the concepts of inverse kinematics and static balance have been used to generate the gaits of lower limbs and trunk, respectively, and its dynamic balance has been verified utilizing the position of zero moment point, whereas in soft computing-based approaches, either neural network- or fuzzy logic-based gait planner has been used to generate the motion of trunk and swing foot of the biped robot. The knowledge bases of neural network and fuzzy logic-based approaches have been optimized separately using a genetic algorithm off-line. The performances of the developed approaches have been tested through computer simulations.

1 Introduction

Last three decades had witnessed a huge development in the area of biped walking. To start wirth, early study on biped robot used to concentrate on its static walking with a very low walking speed [1], and it is said to be balanced, if the projected point of the Center of Gravity (COG) falls within the foot support area. Later on, researchers focussed on dynamic walking of the biped robots, in which the robot could move at a specified speed to perform the task assigned to it. A two-legged

Pandu Ranga Vundavilli
Department of Mechanical Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla-521180, India
e-mail: panduvundavilli@gmail.com

Dilip Kumar Pratihar
Department of Mechanical Engineering, Indian Institute of Technology, Kharagpur-721302, India
e-mail: dkpra@mech.iitkgp.ernet.in

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 57–85.
springerlink.com

robot is said to be dynamically balanced as long as Zero Moment Point (ZMP) [2], a point about which the moments generated by all active forces will be made equal to zero, falls inside the foot support polygon. Planning its gaits to ensure a dynamically balanced walk in complex environments has been identified as a potential field of research. The environment may be either indoor or outdoor, which typically includes doors, objects, staircases, sloping surface and ditches. It is important to mention that the biped robots can negotiate these scenarios, as they are able to perform their walking with discrete foot-holds.

The most important aspects of biped locomotion are the determination of path to be followed by the biped robot in an environment, and generation of suitable gaits in order to avoid collision with the obstacles and uneven terrain existing in its path. For example, a biped robot should be able to lift its foot high enough to negotiate staicases, obstacles, or have support foot with suitable angles to match the inclination of the terrain. It is also important to note that the repeatability conditions are to be fulfilled to accomplish the cyclic gait. Whenever a biped robot is working in an environment, it needs to identify the terrain and plan its gaits to perform dynamically balanced walk on that terrain. This can be achived only if the robot has a certain level of intelligence. Thus, design and development of an intelligent and autonomous biped robot has become a thrust area in robotics research.

2 Literature Review

During locomotion, a biped robot will have to plan its suitable gaits in varying situations after maintaining its dynamic balance, which is measured using the concept of Zero Moment Point (ZMP) [2]. Mathematical modeling and simulation of these mechanisms using analytical approaches [2]-[4] played a key role in the research of two-legged robots. A semi-inverse method was proposed by Juricic et al. [5] for determination of the trunk motion of a biped robot. Lee and Chen [6] obtained minimum-energy trajectories of a biped robot using non-linear programming. The state and control variables were approximated with the help of B-spline functions and a gradient-based algorithm was used to obtain minimum fuel trajectories. To extend the minimum-energy walking method to level ground and uphill slopes, Channon et al. [7] and Roussel et al. [8] had proposed methods of gait generation by minimizing the cost function of energy consumption. Since a two-legged robot has a tendency to tumble easily, it is necessary to consider its balance while determining a walking pattern. Some researchers utilized the concept of static balance [9], and others used the dynamic balance [10]- [12] to determine suitable walking patterns. Moreover, a few conventional optimization techniques, such as Successive Quadratic Programming (SQP) [13] and Second-Order Cone Programming (SOCP) [14] were also developed to tackle the gait generation and balancing problems of a biped robot on different terrains. The said analytical approaches may not be suitable for on-line implementations, due to their inherent computational complexity. Moreover, the solutions of above optimization techniques may get stuck at the local minima.

A lot of parameters are to be considered to minimize the energy consumed by the biped robot. In addition to these problems, there is also a need for the development of adaptive locomotion algorithms, which can negotiate an unknown terrain in an optimal sence. To solve such problems, soft computing-based (that is, Neural Networks (NNs), Fuzzy Logic (FL), Genetic Algorithms (GAs) and their various combinations) techniques, which are adaptive in nature have been studied. These approaches are found to provide with feasible solutions to some complex real-world problems within a reasonable accuracy limit [15]. NNs had been used to modify the gaits of a biped robot moving on various terrains. Several learning methods [16]-[18] and a variety of architectures, such as Cerebellar Model Arithmetic Computer (CMAC) NNs [19], Recurrent Neural Network (RNN) [20] and others, were tried to improve their learning and prediction capabilities. Moreover, Capi et al. [21] developed an approach using GA and Radial Basis Function Neural Network (RBFNN) for real-time gait generations based on two different cost functions, such as minimum consumed energy and minimum torque change. This approach was found to be computationally expensive and hence, slow.

Fuzzy Logic (FL) technique had also been utilized to solve the gait generation problems of a biped robot. A variable gain FL [22] was developed for the intelligent control of a biped walking robot in the double-support phase. In [23], Park proposed a method to reduce the trunk motion of the biped robot with the help of an FL-system based on the ZMP trajectory. This algorithm was tested on a 7-DOF biped robot through computer simulations and it was observed that the stability of the locomotion had been increased with the reduced motion on the trunk. Later on, Cuevas et al. [24] suggested a fuzzy-PD incremental algorithm to control the balance of the biped robot. The algorithm was seen to be computationally economic and was implemented in a micro-controller to control the balance of the robot. Moreover, some investgators had developed hybrid intelligent systems by combining the NN, FL and GA. A hierarchical control system consisting of walking planning level, gait generation level and joint control level was developed with the help of an Adaptive Network-based Fuzzy Inference System (ANFIS) [25], which enhances Takagi and Sugeno type of fuzzy controller with self-learning capability. In [26], Zhou and Meng presented a Fuzzy Reinforcement Learning (FRL) method for dynamic balance control of a biped robot based on neuro-fuzzy network architecture. Later on, Gu and Hu used a GA to evolve the membership functions of an FL [27], in which the FL was encoded as an individual. The effectiveness of the developed controller had been tested on SONY legged robots. Moreover, a method for on-line stable gait generation of a biped robot was proposed in [28] utilizing a genetic-fuzzy system, in which a GA was used to optimize the knowledge base of the Fuzzy Logic Controller (FLC), offline.

3 Research Issues

There is an enough scope for carrying out research in various fields of biped robotics, such as design and development of the robots, identification of the terrain, path and gait planning, and providing autonomy to them. Design and development stage involves construction of the structure of biped robot in such a way that it can work for a long duration with the limited on-board power supply. Identification of the terrain implies recognizing the nature of the environment in which the robot is supposed to move and it can be done with the help of a camera (either single or double) system mounted on-board or off the shelf. The next important area of research deals with planning the path and gait of the biped robot in motion. It involves detecting the collision-free path to be followed after identifying the environment. Autonomy helps the robot to take the decisions independently as the situation demands in a dynamic environment and the principle of Artificial Intelligence (AI) is to be applied for the same. The present chapter deals with the planning of dynamically balanced gaits for a biped robot moving on uneven terrains, such as staircases, sloping surface and ditch using the principles of soft computing.

4 A Case Study

A 7-DOF biped robot will have to plan its dynamically balanced gaits in an optimal sense, while moving through various terrains, such as staircases, sloping surfaces and ditch. Two different approaches, namely analytical and soft computing-based ones have been developed for solving the said problem. The performances of the developed approaches are tested through computer simulations. The following assumptions are made to simplify the problem:

1. The biped robot is assumed to move in the sagittal plane and accordingly, its balance is considered in the direction of motion only.
2. No impact is considered after landing and dynamic balance analysis in the double support phase has been neglected for simplicity.
3. The inputs to the gait planners (that is, positions of foot placement) are generated with the help of human expertise and no sensor is used to extract information of the environment.

4.1 Analytical Approach

The mathematical formulation related to dynamically balanced ascending and descending gaits of a biped robot moving on staircases, sloping surface and crossing a ditch has been discussed in detail, in the subsequent sub-sections.

4.1.1 Ascending the Staircase, Sloping Surface and Crossing the Ditch

A 7-DOF biped robot (three at hip, two at knee and two at ankles) with lumped masses attached to its limbs has been considered in the present study. Figures 1, 2 and 3 show the biped robot ascending the staircase, sloping surface and crossing the ditch, respectively. The trajectories of the swing foot and hip joint are assumed to

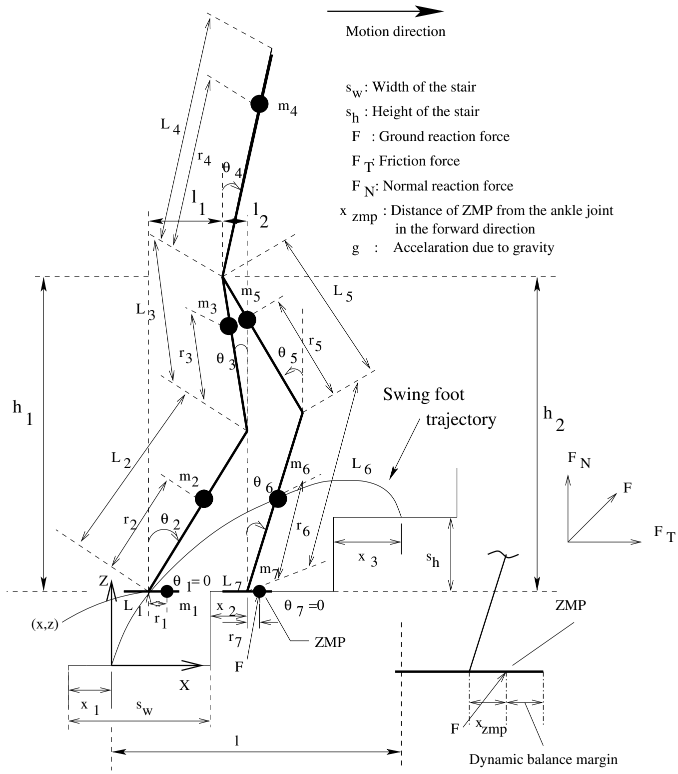


Fig. 1 A schematic view of a two-legged robot (7-DOF) climbing the stairs

be a cubic polynomial and a straight line, respectively. However, it is to be noted that the boundary conditions used to solve the coefficients of cubic polynomial are dependent on the terrain conditions, as shown in Appendix A. The lower limbs' gait has been generated utilizing the concept of inverse kinematics, as given below.

$$\theta_2 = sin^{-1}\left(\frac{h_1 L_3 sin\psi_1 + l_1 (L_2 + L_3 cos\psi_1)}{(L_2 + L_3 cos\psi_1)^2 + (L_3 sin\psi_1)^2}\right), \quad (1)$$

where h_1 and l_1 are the height and distance of hip joint from the ankle joint of the swing leg; L_2 and L_3 are the lengths of two links displayed in Figs. 1 through 3; and the angle ψ_1 has been calculated using the expression $\psi_1 = arccos\left((h_1^2 + l_1^2 - L_2^2 - L_3^2)/2\ L_2\ L_3\right)$. After determining the value of θ_2 using equation (1), another joint angle θ_3 can be calculated as follows: $\theta_3 = \theta_2 - \psi_1$. It is to be noted that h_2 and l_2 are the height and distance of hip joint from the ankle joint of the ground leg, which have been determined utilizing the information related to the geometries of the terrains. Now, the angle ψ_2 can be calculated using the link lengths L_5 and L_6 (refer to Figs. 1

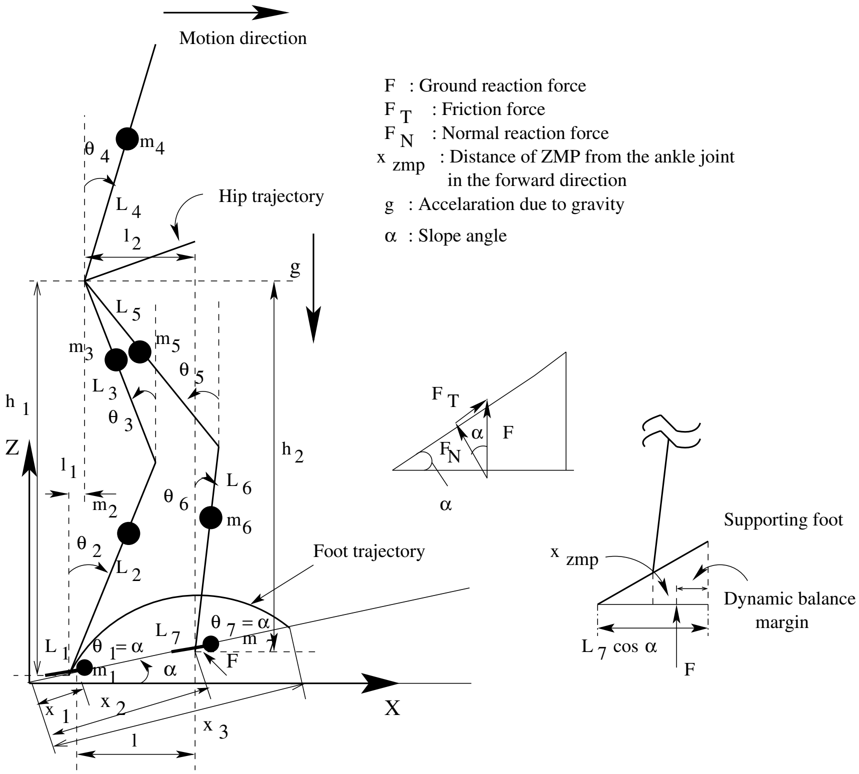


Fig. 2 A schematic view of a two-legged robot (7-DOF) ascending the sloping surface

through 3) and the values of h_2 and l_2, as follows: $\psi_2 = arccos\ ((h_2^2 + l_2^2 - L_6^2 - L_5^2)/2\,L_6\,L_5)$. The angle θ_6 can then be determined using the expression given below.

$$\theta_6 = sin^{-1}\left(\frac{h_2 L_5 sin\psi_2 + l_2\,(L_6 + L_5 cos\psi_2)}{(L_6 + L_5 cos\psi_2)^2 + (L_5 sin\psi_2)^2}\right), \tag{2}$$

It is important to note that θ_5 can be calculated utilizing the following relationship: $\theta_5 = \theta_6 - \psi_2$. The trunk motion is determined first based on the concept of static balance and it is then verified for its dynamic balance using the concept of ZMP. The following repeatability conditions are to be followed by the limbs to ensure a cyclic gait: $\theta_{i,\ initial} = \theta_{j,\ final}$; $\dot{\theta}_{i,\ initial} = \dot{\theta}_{j,\ final}$; $\theta_{k,\ initial} = \theta_{k,\ final}$; $\dot{\theta}_{k,\ initial} = \dot{\theta}_{k,\ final}$, where (i and j) take the values of (2 and 6) and (3 and 5), respectively, and k = 1, 4. Dynamic Balance Margin (DBM) of the biped robot while moving on various terrains is calculated as follows:

$$x_{DBM} = \left(\frac{L_7}{2} cos\ \alpha - |x_{ZMP}|\right), \tag{3}$$

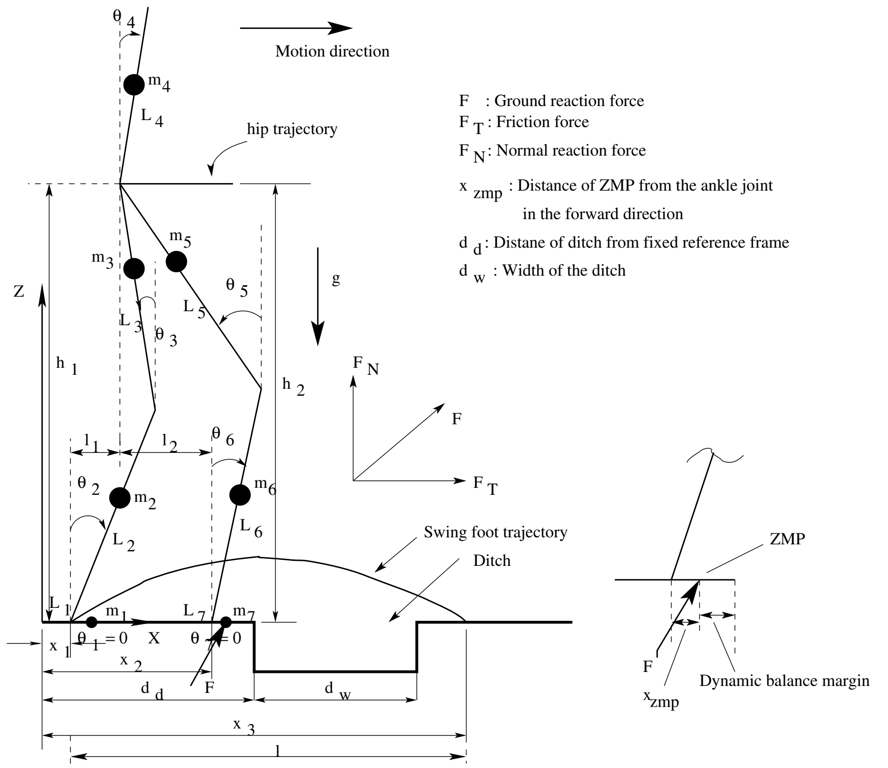


Fig. 3 A schematic view of a two-legged robot (7-DOF) crossing the ditch

where L_7 is the length of the supporting foot, α is the angle of slope, which is kept equal to zero for the problems related to the staircase and ditch and x_{ZMP} is the position of ZMP in the direction of motion and can be calculated as given below.

$$x_{ZMP} = \frac{\sum_{i=1}^{7} \left(I_i \dot{\omega}_i + m_i x_i \left(\ddot{z}_i - g \right) - m_i \ddot{x}_i z_i \right)}{\sum_{i=1}^{7} m_i \left(\ddot{z}_i - g \right)}, \tag{4}$$

where I_i denotes the moment of inertia of i-th link (kg-m^2), $\dot{\omega}_i$ is the angular acceleration of link i (rad/s^2), m_i denotes the mass of i-th link (kg), (x_i, y_i, z_i) is the coordinate of i-th lumped mass, g is the acceleration due to gravity (m/s^2), $\ddot{z}_i$ is the acceleration of link i in z-direction (m/s^2), $\ddot{x}_i$ is the acceleration of link i in x-direction (m/s^2), $\ddot{y}_i$ is the acceleration of link i in y-direction (m/s^2). The dynamics of the biped robot has been solved utilizing Lagrange-Euler formulation. The inertia, Coriolis/centrifugal and gravity terms are shown in Appendix B. The joint angles are assumed to follow a fifth-order polynomial. The amount of power required P is then determined utilizing the following expression: $P = \frac{1}{T} \sum_{i=1}^{n} | \int_0^T \tau_i \dot{q}_i dt |$, where T is the time of travel.

4.1.2 Descending the Staircase and Sloping Surface

The schematic views of the biped robot descending the staircase and sloping surface are shown in Fig. 4. The gait generation process is similar to that of ascending the staircase/sloping surface. However, in descending gait generation, the following variations are to be considered from the ascending gait generation: The first variation lies with the boundary conditions used to determine the coefficients of cubic polynomial trajectory of the swing (refer to Appendix A). The other variation is related to the direction of motion of the robot. In case of descending the staircase and sloping surface, the robot moves in the direction of gravity, due to which, the value of g in Eq. (4) is to be replaced by $-g$.

One drawback of this approach lies in the fact that the obtained solutions may not be optimal in any sense, as no optimizer has been used. Moreover, the concept of static balance has been employed to determine the trunk motion and finally verified for its dynamic balance. It is also important to note that the extracted information of the terrain with the help of camera/sensor may contain some noisy data. To handle the noisy and imprecise data, avoid complex mathematical computations in determining the trunk motion, and generate the dynamically balanced gaits in an optimal sense on various terrains, the robot needs to have an adaptive and robust gait planner. Researchers have felt recently that soft computing-based approaches

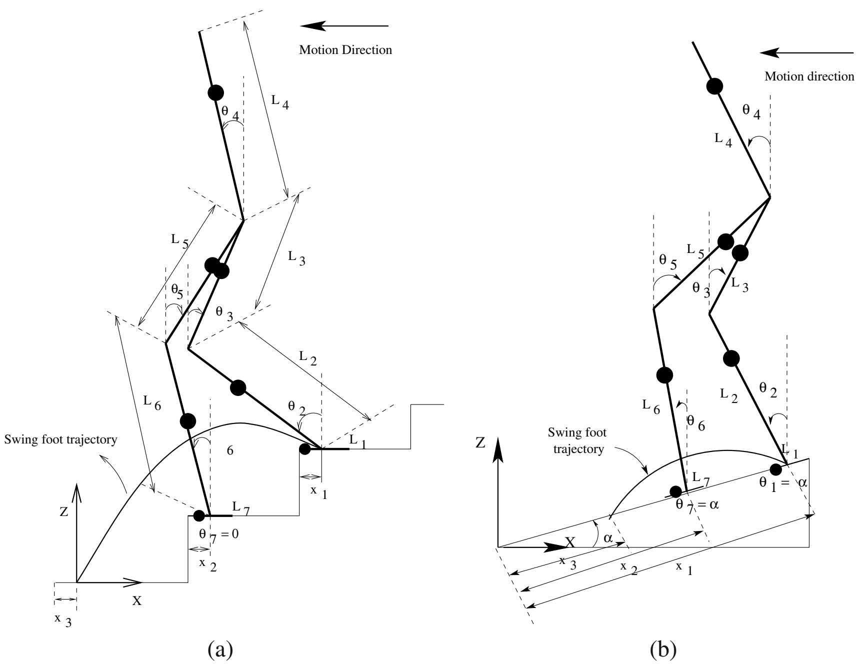


Fig. 4 Schematic views of a two-legged robot (7-DOF) descending a (a) staircase, (b) sloping surface

could serve the said purpose. The next sub-section deals with the development of soft computing-based approaches for planning the gaits of the biped robot.

4.2 Soft Computing-Based Approaches

Neural Network (NN)- and Fuzzy Logic (FL)-based gait planners have been developed for solving the gait generation problems of the biped robot on various terrains, such as staircase, sloping surface and ditch, after considering it as both unconstrained and constrained optimization problems separately. The flowchart of the developed algorithm is shown in Fig. 5, and it has been explained below.

4.2.1 Un-constrained Optimization Using Soft Computing

A biped robot will have to ascend and descend the staircase, sloping surface, and cross a ditch, as the situation demands, after maintaining its maximum dynamic balance during locomotion without considering any functional constraint. Thus, it may be posed as an unconstrained optimization problem as given below.

$$Maximize \ (\frac{L_7}{2} cos\ \alpha - |x_{ZMP}|), \tag{5}$$

where the terms carry usual meaning.

Approach I: Neural Network (NN)-Based Gait Planner

To solve the said problem, two modules of NN have been used, as shown in Fig. 6. A three-layered feed-forward NN is utilized here for both the modules of NN. For the first module of NN, the first layer is the input layer, containing two inputs (x_1 and x_2). The second layer is the hidden layer, whose number of neurons is to be decided through a systematic study. The last layer is the output layer containing two outputs: h_1 and l_1. The angles: θ_2, θ_3, θ_5 and θ_6 are determined with the help of Eqns. (1) and (2). Later on, the changes in angles, that is, $\delta\theta_2$ and $\delta\theta_3$ are given as inputs to the first layer of the second module of NN. Two ouputs, namely $\delta\theta_1$ and $\delta\theta_4$ are obtained from this module of NN. The proposed architecture of the NN used to tackle the problems related to ascending and descending the staircase, sloping surface and crossing the ditch is kept the same, as shown in Fig. 6. However, the number of neurons in the hidden layer of the network may be different. A GA is utilized to optimize the connecting weights of the fully-connected feed-forward NNs. Thus, the optimized NNs will be evolved by a GA-based training carried out off-line.

Fitness Calculation

Two hundred and sixteen training cases are generated by taking six equi-spaced values for each of x_1, x_2 and x_3. A batch mode of training has been employed to

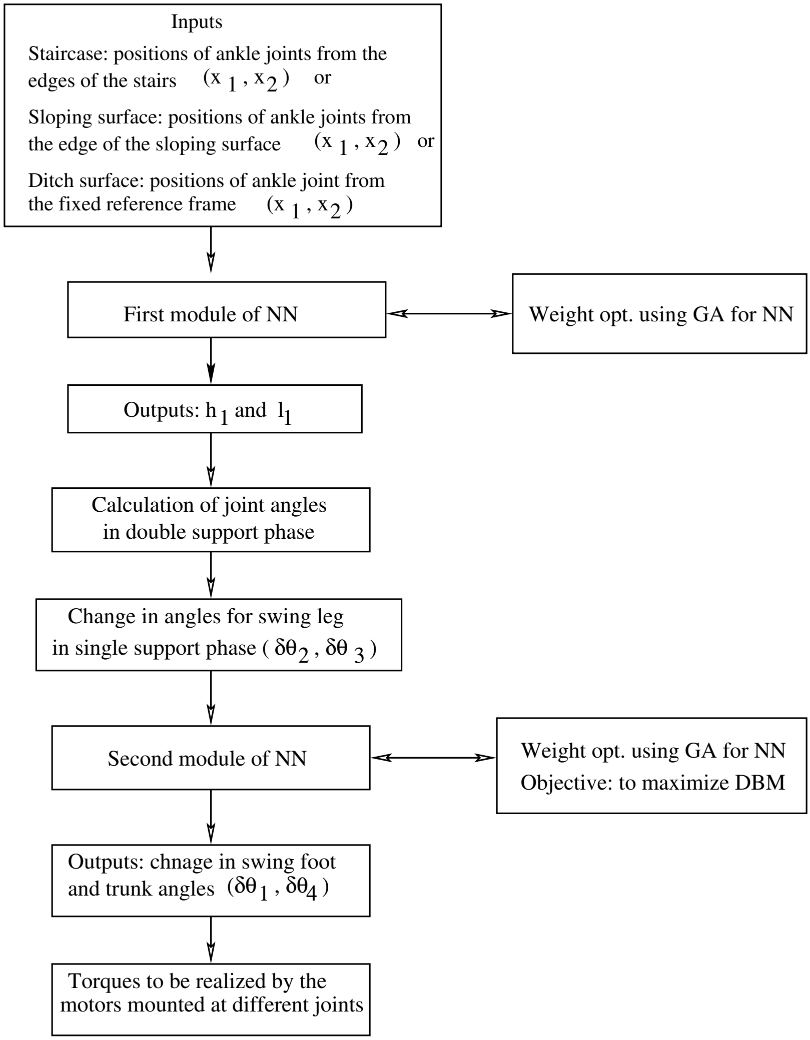


Fig. 5 Flowchart of the proposed NN-based gait planner

train the network. The average of all the objective function (that is, DBM) values is treated as the fitness (f) of a GA-string and it is determined as follows:

$$f = \frac{\sum_{i=1}^{S} DBM_i}{S}, \tag{6}$$

where S represents the number of training cases considered. A high penalty equal to -100 is added to the fitness value, if the NNs represented by the GA-string are unable to generate the dynamically balanced gait.

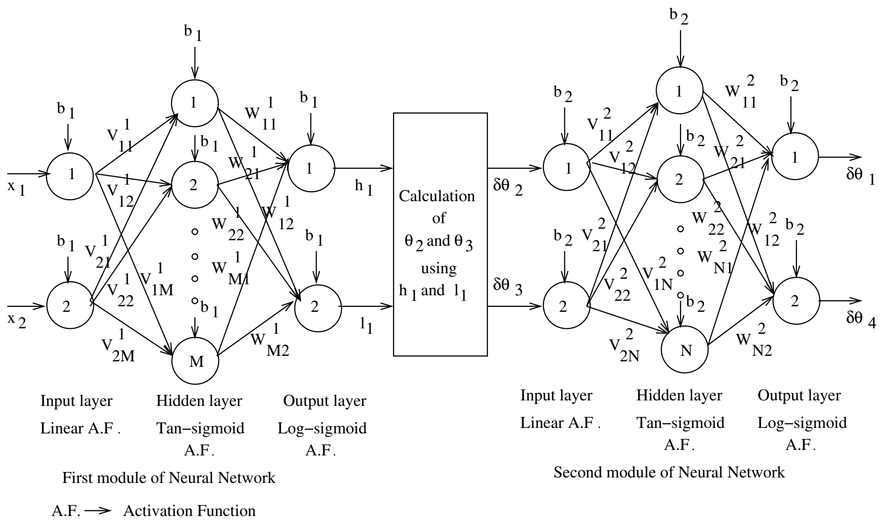


Fig. 6 A schematic view showing the working principle of NN-based gait planner

Approach II: Fuzzy Logic (FL)-Based Gait Planner

An approach for automatic design of the FL-system has been developed for solving the gait generation problems related to ascending and descending the staircase, sloping surface and crossing the ditch. The GA will be used to optimize the Knowledge Base (KB) (that is, data base and rule base) of the FL-system. The flowchart of this approach is similar to that shown in Fig. 5, in which the NN is to be replaced by the FL-system. Here also, the same methodology has been used to solve the ascending and descending gait generation problems of the biped robot moving on the staircase, sloping surface and crossing the ditch. The operating principle of FL-based approach is similar to that of the NN-based approach. Two modules of FL have been used instead of two modules of NN and the GA will be utilized to optimize the KB of FL-system. The membership function distributions of the input and output variables of the first and second modules of FL for ascending the staircase are shown in Figs. 7 and 8, respectively. The *a* values indicate the base-widths of right-angled triangles and half base-widths of the isosceles triangles. The representation of input and output variables' membership function distributions for descending the staircase, sloping surface and crossing the ditch have been kept the similar to those shown in Figs. 7 and 8, respectively. The only difference lies in the fact that the starting values of some of these membership function distributions are different. Each variable of the FL-module is represented with the help of four linguistic terms. For example, the variable: *distance* is represented using Low (L), Medium (M), High (H) and Very High (VH), whereas the *angle* has been denoted by Negative Large (NL), Negative Small (NS), Positive Small (PS) and Positive Large (PL). A batch

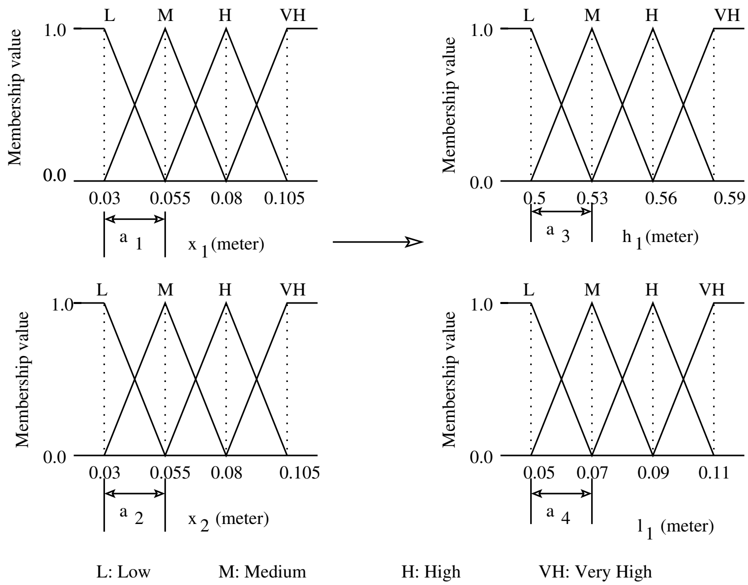


Fig. 7 Membership function distributions for input and output variables of the first module of FL-system - ascending gait generation

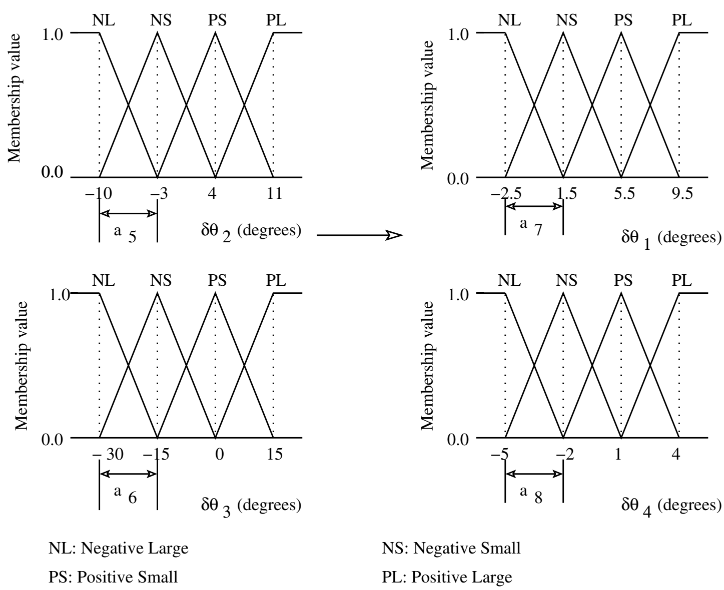


Fig. 8 Membership function distributions for input and output variables of the second module of FL-system - ascending gait generation

mode of training has been adopted as discussed above in the NN-based system and the fitness of a GA-string is calculated utilizing Eq. (6).

4.2.2 Constrained Optimization Using Soft Computing

In this method, the biped robot will have to plan its dynamically balanced gait after optimizing the positions of mass centers on the limbs and hip trajectory, while moving through various terrains, such as the staircases, sloping surfaces and ditch. It is to be noted that the robot should generate the above gaits by consuming the minimum power. Moreover, the rate of change of torque for each joint must be kept below a pre-defined value to ensure smooth walking of the robot. Thus, it may be posed as a constrained optimization problem as given below.

$$Maximize\ \left(\frac{L_7}{2} cos\ \alpha - |x_{ZMP}|\right) + \frac{1}{P}, \tag{7}$$

$$subject\ to \qquad \Delta\tau_{ij} \leq \Delta\tau_{specified},$$

where i and j represent number of joints and time instants, respectively; P is the average power consumption and $\Delta\tau$ represents the change in joint torque value. The remaining terms carry the usual meaning explained earlier.

Approach I: Neural Network (NN)-Based Gait Planner

The methodology and algorithm used in this case, are also similar to the ones utilized to solve the un-constrained optimization problem. The difference lies in the fact that the position of masscenter for each link has been varied within a predefined range on that link, and the hip joint is assumed to follow a cubic polynomial as given below.

$$z_h = k_0 + k_1 x + k_2 x^2 + k_3 x^3, \tag{8}$$

where z_h represents the height of the hip joint at a distance x from the starting point and k_0, k_1, k_2 and k_3 are the coefficients. Among the four coefficients of the polynomial, two (that is, k_0 and k_1) are taken from the GA-string and the remaining two (that is, k_2 and k_3) are calculated from the boundary conditions of the hip joint. Moreover, the positions of mass centers represented by r_1 through r_4 (refer to Fig. 1) are also taken from the GA-string. It is important to note that the values of r_5, r_6 and r_7 are made equal to r_3, r_2 and r_1, respectively, to make the structure of the biped robot symmetric about the central axis. Finally, the optimal NN will be evolved with the help of GA-based training given off-line. In this way, an attempt has been made for the simultaneous optimization of structure and gait of the biped robot moving on uneven terrains.

Fitness Calculation

A batch mode of training has been employed off-line to train the NNs with the help of two hundred and sixteen cases. The fitness (f) of a GA-string is calculated as the average of all objective function values, as given below.

$$f = \frac{\sum_{i=1}^{S} DBM_i + 1/P_i}{S}, \tag{9}$$

where S represents the number of training cases considered. A high penalty equals to -100 is added to the fitness value, if the NNs represented by the GA-string are unable to generate the dynamically balanced gait or the change in torque for a particular joint exceeds the pre-defined value.

Approach II: Fuzzy Logic-Based Gait Planner

The working principle of the FL-based gait planner used for solving this problem is also similar to that of the one utilized to tackle the un-constrained optimization problem. The only variation lies in the introduction of some more variables (such as k_0, k_1, r_1, r_2, r_3 and r_4) into the GA-string for optimization. The fitness of a GA-string is calculated utilizing Eq. (9).

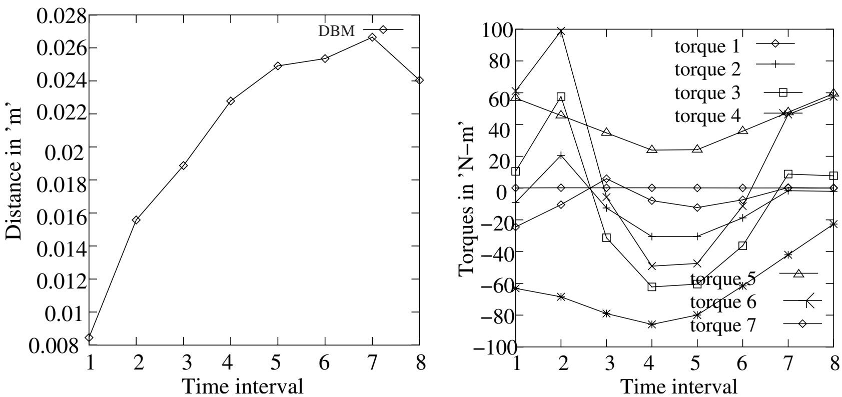
4.3 Results and Discussion

The performances of the above approaches have been tested through computer simulations. The following length L (m), mass m (kg) and moment of inertia I ($kg\ m^2$) parameters of the biped robot have been considered: $L_1 = 0.06$, $L_2 = 0.34$, $L_3 = 0.30$, $L_4 = 0.60$, $L_5 = 0.30$, $L_6 = 0.34$, $L_7 = 0.06$; $m_1 = 0.5$, $m_2 = 2.0$, $m_3 = 5.0$, $m_4 = 30.0$, $m_5 = 5.0$, $m_6 = 2.0$, $m_7 = 0.5$; $I_1 = 0.0006$, $I_2 = 0.021067$, $I_3 = 0.0780$, $I_4 = 1.8720$, $I_5 = 0.0780$, $I_6 = 0.021067$, $I_7 = 0.0006$. The slope of the surface is kept equal to 10^o.

4.3.1 Analytical Approach

Results of computer simulations related to the variations of DBM and joint torques are shown in Fig. 9. The DBM value is found to increase initially and then decrease at the end of the cycle (refer to Fig. 9(a)). It happens due to the fact that the ZMPs move towards the ankle joint during all the intervals except the last one. Fig. 9(b) shows the variation of torques at different joints. It is to be noted that anticlockwise torque has been considered as negative and clockwise torque is assumed to be positive. It is interesting to note that during the second time interval, the torque values of ankle, knee and hip joints of the swing leg and that of the trunk joint are coming out to be relatively more, compared to those found during the other time intervals. It happens due to the fact that the effect of inertia is more at this interval, as the biped robot starts swinging its leg. It is also interesting to note that the knee torque of the supporting leg has come out to be more compared to other joint torques. This characteristic exactly matches with the human-beings' experience of ascending and descending the staircase and sloping surface.

It is impotant to mention that the variation of these quantities on other problems, such as descending the staircase, ascending and descending the sloping surface and crossing the ditch have yielded the similar trends. Results of ascending and descending the staircase, sloping surface and crossing the ditch have been summarized in



(a) Variation of DBM with time (seconds) in x-direction during ascending

(b) Variation of joint torques with time (seconds) during ascending

Fig. 9 Variations of DBM and joint torques during ascending the staircase in one complete cycle

Table 1 Summary of results of Analytical approach

	Staircase		Sloping surface		Ditch
Quantity	Ascending	Descending	Ascending	Descending	surface
DBM (m)	0.021	0.019	0.023	0.022	0.023
Power (W)	47.375	41.933	42.952	16.238	13.547

Table 1. From this table, it can be observed that ascending gaits are dynamically more balanced than the descending ones. It is also important to note that ascending requires more power than the descending does. These characteristics also exactly match with the experiences of human-beings moving in the presence of some staircases and sloping surfaces.

4.3.2 Soft Computing-Based Approaches

An attempt has been made to design suitable NN- and FL-based gait planners of a biped robot, so that it can generate dynamically balanced gaits while moving through uneven terrains containing of staircase, sloping surface and ditch. Five modules have been developed for each of the NN- and FL-based approaches, out of which four deal with ascending and descending gait generations through the staircase and sloping surface and the remaining one is related to ditch-crossing gait generation of the biped robot. These modules are optimized off-line using a GA. Once the optimal modules are obtained, those can be used on-line for some test cases generated at random. It is to be noted that the performances of the developed gait planners have been tested here through the computer simulations only. Fig. 10 shows the flowchart explaining the way simulation has been carried out.

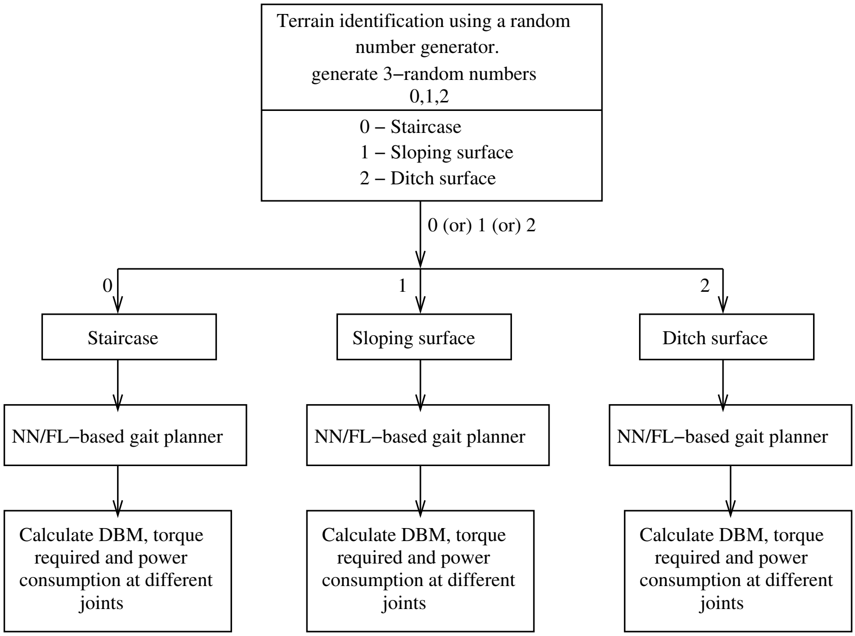


Fig. 10 Flowchart for simulation of the test cases

Three numbers are generated at random to represent the test scenarios. For example, 0, 1 and 2 denote the staircase, sloping surface and ditch, respectively. There are twenty seven possibilities with the generated three numbers, such as 000, ..., 012, ..., and 222, corresponding to staircase-staircase-staircase, ..., staircase-sloping surface-ditch, ..., and ditch-ditch-ditch, respectively. It is to be noted that ascending through the staircase and sloping surface is followed by its descending. Moreover, the inputs to the gait planners (that is, positions of feet placement) have also been generated at random. To verify the feasibility of the proposed scheme, results of one particular test scenario, say 012 (that is, staircase ascending and descending, sloping surface ascending and descending, followed by the ditch crossing) have been presented here for both the un-constrained as well as constrained optimization problems.

Un-constrained Optimization

The plots representing the variations of average DBM and average torque required for ascending and descending the staircase, sloping surface and crossing the ditch for both the NN- and FL-based gait planners are seen to be similar in nature to those of Figs. 9(a) and (b), respectively. The values of average DBM (m) for ascending and descending the staircase, sloping surface and crossing the ditch using NN- and FL-based gait planners are found to be equal to (0.026, 0.025, 0.025, 0.025, 0.026) and (0.026, 0.026, 0.025, 0.025, 0.026), respectively. Moreover, the power consumptions (W) for the said order of modules are seen to be equal to (26.692,

Table 2 Results of comparison for the NN- and FL-based approaches for ten test cases - un-constrained optimization

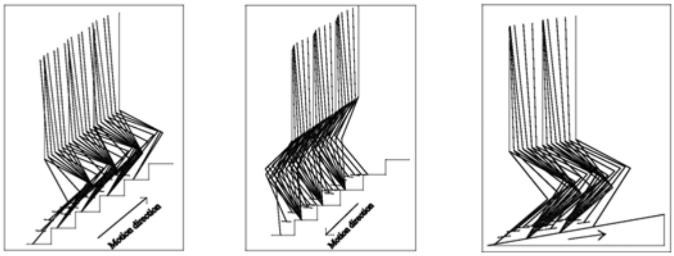
Scene No.	Average DBM in × $10^{-}6$ m									
	Staircase				Sloping surface				Ditch surface	
	Ascending		Descending		Ascending		Descending			
	NN	FL	NN	FL	NN	FL	NN	FL	NN	FL
1	256	259	252	251	254	257	252	254	255	256
2	260	260	254	255	251	254	249	251	252	253
3	260	262	253	254	254	258	253	254	254	256
4	259	259	253	254	253	257	252	255	266	267
5	251	253	252	254	249	252	248	250	258	259
6	271	264	253	254	264	266	261	264	260	262
7	264	264	251	251	252	257	253	255	258	258
8	272	265	253	254	253	257	252	254	256	256
9	256	253	252	251	259	263	258	258	261	262
10	258	247	252	250	256	260	255	257	256	256

13.289, 20.906, 16.527, 11.598) and (29.339, 21.547, 28.306, 25.055, 16.174), respectively. It can be observed from these results that ascending gaits are dynamically more balanced than the descending ones, but at the cost of more power consumption. Moreover, it is also important to mention that knee joint torque of the supporting leg has come out to be more compared to other joint torques. In this method of gait generation also, these characteristics exactly match with the experience of human-beings. Results of comparison for both the NN- and FL-based approaches for 10 random test cases are shown in Table 2. FL-based approach has slightly outperformed the NN-based approach for most of the test cases. Thus, the FL-based approach has performed a slightly better than other approach and it may happen due to the reason that some problem information has been injected into the data base of the FLC before the commencement of its training. The designer uses his/her KB to initially design the membership function distributions of the variables. On the other hand, no problem information is fed to the NN prior to its training. Figures 11 and 12 show the simulation results of the two-legged robot ascending and descending the staircase, sloping surface and crossing a ditch using the NN- and FL-based approaches, respectively.

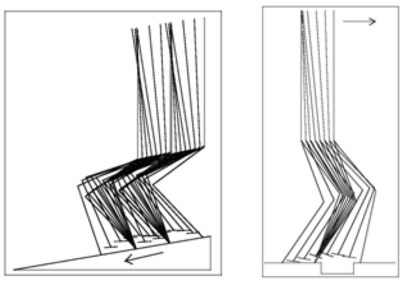
Results of computer simulations show that the biped robot has successfully generated the dynamically balanced gaits for all the test cases using both the NN- and FL-based approaches. It is important to note that the hip joint follows a straight path in all the simulation runs. Moreover, it can be observed from these figures that the lower limbs' gait is periodic for ascending and descending the staircase and sloping surface, whereas the gait is found to be aperiodic for crossing the ditch.

Robustness Test

The NN- and FL-based gait planners of a two-legged robot developed in this study are tested for their performances by allowing a little variation in the values of



(a) Staircase ascending (b) Staircase descending (c) Sloping surface ascending



(d) Sloping surface descending (e) Crossing a ditch

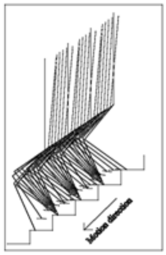
Fig. 11 Simulations of the biped robot using NN-based approach - un-constrained optimization

input variables (that is, x_1, x_2 and x_3). Table 3 shows the results of robustness test conducted on the developed gait planners. It is interesting to note that even $\pm$ 15% variations in the input variables have led to a maximum of 2.20%, 0.062% and 0.428% variations in the DBM values for the staircase, sloping surface and ditch surface gait planners, respectively. Thus, the developed planners are able to generate dynamically balanced robust gaits.

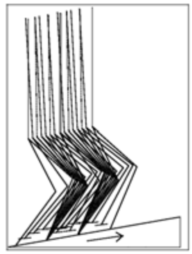
Table 3 Results of robustness test for the developed gait planners - un-constrained optimization

% changes in inputs	% changes in DBM									
	Staircase				Sloping surface				Ditch surface	
	Ascending		Descending		Ascending		Descending			
	NN	FL	NN	FL	NN	FL	NN	FL	NN	FL
-15	0.451	-0.793	1.152	-2.122	0.001	0.062	0.004	0.040	0.007	0.015
-10	0.283	-0.523	0.771	-1.513	0.004	0.035	0.004	0.039	0.004	0.012
-5	0.232	-0.441	0.381	-0.303	0.000	0.012	0.000	0.012	0.004	0.011
5	-0.350	0.891	-0.390	0.220	-0.003	0.004	0.000	-0.016	0.000	-0.012
10	-0.801	1.062	-0.790	0.391	-0.003	0.024	-0.004	-0.039	-0.298	-0.405
15	-2.202	1.492	-1.202	0.782	-0.007	0.051	-0.004	-0.047	-0.310	-0.428

(a) Staircase ascending

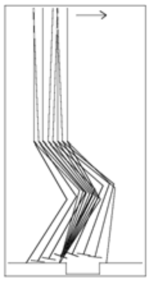


(b) Staircase descending



(c) Sloping surface ascending

(d) Sloping surface descending



(e) Crossing a ditch

Fig. 12 Simulations of the biped robot using FL-based approach - un-constrained optimization

Adaptability Test

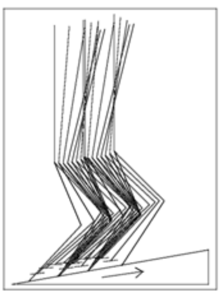
The developed NN- and FL-based gait planners are also tested for their adaptability in handling variations in geometry of the terrains. This test has been conducted by changing the geometry of the terrains (that is, changing the width (s_w) and height (s_h) of the staircase; angle of the sloping surface; and location (d_d) and width (d_w) of the ditch). Results of the adaptability test are shown in Table 4. It is important to mention that the developed NN- and FL-based gait planners have generated the dynamically balanced gaits in all the scenarios. Thus, the developed gait planners are found to be adaptive in nature.

Constrained Optimization

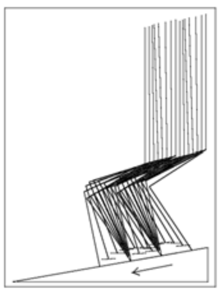
Constrained optimization problem (refer to Sub-section 4.2.2) has been solved using both the NN- and FL-based approaches. The flowchart shown in Fig. 10 has been utilized to test the performance of the developed approaches. The values of average DBM (m) for ascending and descending the staircase and sloping surface, and crossing the ditch using NN- and FL-based gait planners are found to be equal to (0.025, 0.024, 0.024, 0.018, 0.024) and (0.024, 0.023, 0.025, 0.020, 0.025), respectively. Moreover, the power consumptions (W) for the said order of modules are seen to be equal to (24.447, 5.490, 14.037, 7.149, 7.676) and (24.404, 8.928, 11.283, 6.122, 7.561), respectively.

Table 4 Results of adaptability test showing the DBM values for the developed gait planners - un-constrained optimization

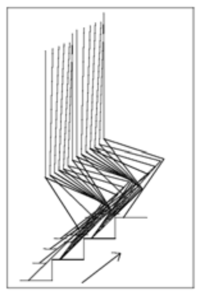
Scen-ario no.	Staircase			Sloping surface			Ditch	
	(s_w, s_h) in m	Ascend in m	Descend in m	slope in deg	Ascend in m	Descend in m	(d_d, d_w) in m	Ditch in m
	NN-based gait planner							
1	(0.13, 0.08)	0.026	0.025	8	0.025	0.025	(0.23, 0.11)	0.026
2	(0.14, 0.09)	0.026	0.025	9	0.025	0.025	(0.25, 0.13)	0.026
3	(0.15, 0.1)	0.025	0.025	10	0.024	0.024	(0.26, 0.15)	0.026
4	(0.16, 0.11)	0.025	0.025	11	0.024	0.024	(0.27, 0.16)	0.025
5	(0.17, 0.12)	0.025	0.025	12	0.024	0.024	(0.28, 0.15)	0.025
	FL-based gait planner							
1	(0.13, 0.08)	0.026	0.026	8	0.026	0.026	(0.23, 0.11)	0.026
2	(0.14, 0.09)	0.026	0.026	9	0.026	0.025	(0.25, 0.13)	0.026
3	(0.15, 0.1)	0.026	0.025	10	0.026	0.025	(0.26, 0.15)	0.026
4	(0.16, 0.11)	0.025	0.025	11	0.026	0.025	(0.274, 0.16)	0.026
5	(0.17, 0.12)	0.025	0.025	12	0.025	0.025	(0.28, 0.15)	0.026



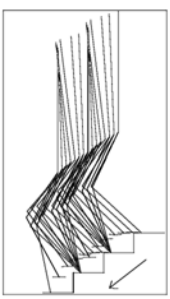
(a) Sloping surface ascending



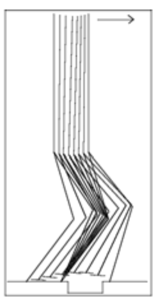
(b) Sloping surface descending



(c) Staircase ascending



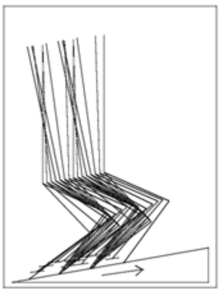
(d) Staircase descending



(e) Crossing a ditch

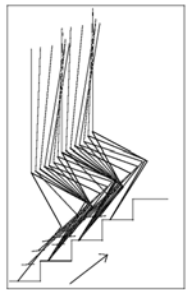
Fig. 13 Simulations of the biped robot using NN-based approach - constrained optimization

Again, for this problem, FL-based gait planner is found to perform better than the NN-based gait planner in most of the test scenarios. It has happened so, due to the reason explained earlier. Simulation results of the biped robot moving on the staircase, sloping surface, and crossing the ditch using NN- and FL-based gait planners are shown in Figs. 13 and 14, respectively. The FL-based gait planner has also been found to generate the gaits similar to that yielded by the NN-based gait planner. Thus, the biped robot is seen to successfully negotiate the staircase, sloping surface and ditch using both the approaches. Moreover, the generated gait by both the NN- and FL-based gait planners is found to be periodic for all the problems except ditch crossing. It can also be observed from the figures that the hip joint follows cubic polynomial trajectory, which is different from the earlier observation.

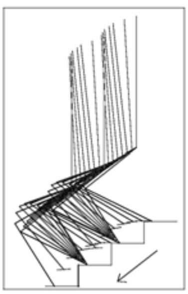


(a) Sloping surface ascending

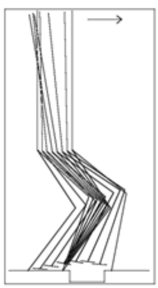
(b) Sloping surface descending



(c) Staircase ascending



(d) Staircase descending



(e) Crossing a ditch

Fig. 14 Simulations of the biped robot using FL-based approach - constrained optimization

For the constrained optimization problems also, both robustness and adaptability have been tested for the developed NN- and FL-based gait planners. Both the NN- as well as FL-based approaches are found to be robust and adaptive in nature.

5 Concluding Remarks

The following conclusions have been made from the above study:

- Both the approaches, namely analytical and soft computing-based approaches are able to generate dynamically balanced gaits for the biped robot moving on various terrains (that is, staircase, sloping surface and ditch). Ascending gaits are found to be more balanced but consume more power compared to the descending gaits. The knee torque of the supporting leg has come out to be the maximum. Both the above observations are in line with the general experience of a human-being ascending and descending the staircase and sloping surface.
- Both the NN- as well as FL-based planners are able to generate dynamically balanced gaits of the biped robot and the solutions of the proposed approaches have shown the similar trends.
- FL-based approach has performed better than the NN-based approach in terms of DBM and power consumption. It might have happened due to the fact that in the FL-based approach, some problem information has been fed into its data base prior to its GA-based optimization, which could not be done in the NN-based approach.
- It may be possible to implement the proposed algorithms on-line, as the CPU time values for solving 20 test cases by the NN- and FL-based approaches are found to be equal to 0.01 and 0.02 seconds, respectively, on a P-IV machine. However, the following issues are to be considered for conducting experiments with the real robot: on-line modeling and recognition of the environment; modeling of foot-ground interaction; movement in the lateral direction; closed-loop control system to check whether the robot has reached the desired location; and others.

6 Scope for Future Study

The present work may be extended in a number of ways, as indicated below.

- The performance of the developed model has been tested on computer simulations. However, it will be more interesting to carry out experiments with the real robot.
- A 7-DOF biped robot has been considered in this study. It may be extended for a two-legged robot having more than 7-DOF, in order to allow it to move in the lateral direction and mimic human walking in a more realistic way.
- Here, an attempt has been made to solve the problems related to adaptive gait generation of a biped robot moving on uneven terrains. However, the problem of path planning can also be integrated, in future, with the present problem of gait planning.

Appendix A: Bounadary Conditions for Swing Foot Trajectory

The trajectory of the swing foot (ankle joint) is assumed to follow a cubic polynomial as follows:

$$z = c_0 + c_1 x + c_2 x^2 + c_3 x^3,$$

where z represents the height of the swing foot (that is, ankle joint) at a distance x from the starting point and c_0, c_1, c_2 and c_3 are the coefficients, whose values are determined with the help of some boundary conditions as given below.

Ascending the staircase (refer to Fig. 1):

- at $x = 0$, $z = 0$,
- at $x = s_w - x_1 - \frac{f_s}{2}$, $z = s_h + \frac{f_s}{2}$,
- at $x = 2s_w - x_1 - \frac{f_s}{2}$, $z = 2s_h + \frac{f_s}{2}$,
- at $x = 2s_w - x_1 + x_3$, $z = 2s_h$,

Ascending the sloping surface (refer to Fig. 2):

- at $x = x_1 \cos\ \alpha$, $z = x_1 \sin\ \alpha$,
- at $x = (x_1 \cos\ \alpha + x_2 \cos\ \alpha)/2$, $z = x_2 \sin\ \alpha + \frac{f_s}{2}$,
- at $x = (x_2 \cos\ \alpha + x_3 \cos\ \alpha)/2$, $z = x_3 \sin\ \alpha + \frac{f_s}{2}$,
- at $x = x_3 \cos\ \alpha$, $z = x_3 \sin\ \alpha$,

Ditch crossing (refer to Fig. 3):

- at $x = x_1$, $z = 0$,
- at $x = d_d - \frac{f_s}{2}$, $z = \frac{f_s}{2}$,
- at $x = d_d + d_w - \frac{f_s}{2}$, $z = \frac{f_s}{2}$,
- at $x = x_3$, $z = 0$,

Descending the staircase (refer to Fig. 4 (a)):

- at $x = 2s_w + x_1 - x_3$, $z = 2s_h$,
- at $x = 2s_w - x_3 + \frac{f_s}{2}$, $z = 2s_h + \frac{f_s}{2}$,
- at $x = s_w - x_3 + \frac{f_s}{2}$, $z = s_h + \frac{f_s}{2}$,
- at $x = 0$, $z = 0$.

Descending the sloping surface (refer to Fig. 4 (b)):

- at $x = x_1 \cos\ \alpha$, $z = x_1 \sin\ \alpha$,
- at $x = (x_1 \cos\ \alpha + x_2 \cos\ \alpha)/2$, $z = x_1 \sin\ \alpha + \frac{f_s}{2}$,
- at $x = (x_2 \cos\ \alpha + x_3 \cos\ \alpha)/2$, $z = x_2 \sin\ \alpha + \frac{f_s}{2}$,
- at $x = x_3 \cos\ \alpha$, $z = x_3 \sin\ \alpha$.

where s_w and s_h are the width and height of the staircase, respectively; f_s indicates the length of swing foot (that is, L_1); α is the angle of slope; d_d and d_w denote the location and width of the ditch; x_1, x_2 and x_3 are the feet placements.

Appendix B: Determination of Joint Torques of the Biped Robot

The joint torques of the robot can be determined as given below.

$$\tau_i = \sum_{k=1}^{n} D_{ik}\ddot{q}_k + \sum_{k=1}^{n}\sum_{m=1}^{n} h_{ikm}\dot{q}_k\dot{q}_m + C_i, \qquad i = 1,2,\cdots,n$$

where n = number of joints,

$$D_{ik} = \sum_{j=max(i,k)}^{n} Tr(U_{jk}J_jU_{ji}^T) \qquad i,k = 1,2,\cdots,n,$$

$$h_{ikm} = \sum_{j=max(i,k,m)}^{n} Tr(U_{jkm}J_jU_{ji}^T) \qquad i,k,m = 1,2,\cdots,n,$$

$$C_i = \sum_{j=1}^{n}(-m_jgU_{ji}\bar{r}_j^j) \qquad i = 1,2,\cdots,n,$$

where D_{ik} represents inertia terms, h_{ikm} indicates Coriolis and centrifugal terms and C_i carries information of the gravity terms.

D, h and C terms of joint torques for ascending gait generation:
The D terms are found to be as follows:

- $D_{11} = D_{22} + a_{11} + (a_{24} + a_{34} + a_{44} + a_{54} + a_{64} + a_{74})\, L_2^2 + (2\, ((a_{34} + a_{44} + a_{54} + a_{64} + a_{74})\, L_3$ - $2\, a_{22})\, L_2 \cos(q_2)$ - $2\, a_{32}\, L_2 \cos(q_2 + q_3)$ - $2\, a_{42}\, L_2 \cos(q_2 + q_3 + q_4) + (2\, a_{74}\, L_5 + 2\, a_{64}\, L_5$ - $2\, a_{52})\, L_2 \cos(q_2 + q_3 + q_4 + q_5) + (2\, a_{74}\, L_6$ - $2\, a_{62})\, L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6)$ - $2\, a_{72}\, L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $D_{12} = D_{21} = D_{22}$ - $a_{22}\, L_2 \cos(q_2)$ - $a_{32}\, L_3 \cos(q_3)$ - $a_{42}\, L_3 \cos(q_3 + q_4)$ - $a_{52}\, L_3 \cos(q_3 + q_4 + q_5)$ - $a_{62}\, L_3 \cos(q_3 + q_4 + q_5 + q_6)$ - $a_{62}\, L_5 \cos(q_6) + a_{64}\, L_3\, L_5 \cos(q_3 + q_4 + q_5)$ - $a_{72}\, L_3 \cos(q_3 + q_4 + q_5 + q_6 + q_7) + a_{74}\, L_3\, L_5 \cos(q_3 + q_4 + q_5) + a_{74}\, L_3\, L_6 \cos(q_3 + q_4 + q_5 + q_6) + a_{74}\, L_5\, L_6 \cos(q_6)$ - $a_{32}\, L_2 \cos(q_2 + q_3) + a_{34}\, L_2\, L_3 \cos(q_2)$ - $a_{42}\, L_2 \cos(q_2 + q_3 + q_4) + a_{44}\, L_2\, L_3 \cos(q_2)$ - $a_{52}\, L_2 \cos(q_2 + q_3 + q_4 + q_5) + a_{54}\, L_2\, L_3 \cos(q_2)$ - $a_{64}\, L_2\, L_6 \cos(q_2 + q_3 + q_4 + q_5 + q_6) + a_{64}\, L_2\, L_3 \cos(q_2) + a_{64}\, L_2\, L_5 \cos(q_2 + q_3 + q_4 + q_5)$ - $2\, a_{72}\, L_5 \cos(q_6 + q_7)$ - $a_{72}\, L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6 + q_7) + a_{74}\, L_2\, L_3 \cos(q_2) + a_{74}\, L_2\, L_5 \cos(q_2 + q_3 + q_4 + q_5) + a_{74}\, L_2\, L_6 \cos(q_2 + q_3 + q_4 + q_5 + q_6) + (a_{34} + a_{44} + a_{54} + a_{64} + a_{74})\, L_3\, L_4$;
- $D_{13} = D_{31} = D_{23}$ - $a_{32}\, L_2 \cos(q_2 + q_3)$ - $a_{42}\, L_2 \cos(q_2 + q_3 + q_4) + (a_{64}\, L_5 + a_{74}\, L_7$ - $a_{52})\, L_2 \cos(q_2 + q_3 + q_4 + q_5) + (a_{74}\, L_6$ - $a_{62})\, L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6)$ - $a_{72}\, L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $D_{14} = D_{41} = D_{24}$ -$a_{42}\, L_2 \cos(q_2 + q_3 + q_4) + (a_{64}\, L_5 + a_{74}\, L_7$ - $a_{52})\, L_2 \cos(q_2 + q_3 + q_4 + q_5) + (a_{74}\, L_6$ - $a_{62})\, L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6)$ - $a_{72}\, L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$ - $2\, a_{72}\, L_6 \cos(q_7)$;

- $D_{15} = D_{51} = D_{25} + (a_{74}\ L_7 + a_{64}\ L_5 - a_{52})\ L_2 \cos(q_2 + q_3 + q_4 + q_5) + (a_{74}\ L_6 - a_{62})\ L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6) - a_{72}\ L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6 + q_7) - a_{62}\ L_5 \cos(q_7)$;
- $D_{16} = D_{61} = D_{26} - a_{62}\ L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6)$;
- $D_{17} = D_{71} = D_{27} - a_{72}\ L_2 \cos(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $D_{22} = D_{33} + a_{21} - (a_{33} + a_{32})\ L_3 \cos(q_3) + (a_{34} + a_{44} + a_{54} + a_{64} + a_{74})\ L_3^2 - (a_{43} + a_{42})\ L_3 \cos(q_3 + q_4) + (a_{53} + a_{52} + 2a_{64}\ L_5 + 2a_{74}\ L_5)\ L_3 \cos(q_3 + q_4 + q_5) + (2a_{74}\ L_6 - a_{63} - a_{62})\ L_3 \cos(q_3 + q_4 + q_5 + q_6) - (a_{73} + a_{72})\ L_3 \cos(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $D_{23} = D_{32} = D_{33} - a_{32}\ L_3 \cos(q_3) - a_{42}\ L_3 \cos(q_3 + q_4) + (a_{64}\ L_5 + a_{74}\ L_7 - a_{52})\ L_3 \cos(q_3 + q_4 + q_5) + (a_{74}\ L_6 - a_{62})\ L_3 \cos(q_3 + q_4 + q_5 + q_6) - a_{72}\ L_3 \cos(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $D_{24} = D_{34} - a_{42}\ L_3 \cos(q_3 + q_4) + (a_{64}\ L_5 + a_{74}\ L_5 - a_{52})\ L_3 \cos(q_3 + q_4 + q_5) + (a_{74}\ L_6 - a_{62})\ L_3 \cos(q_3 + q_4 + q_5 + q_6) - a_{72}\ L_3 \cos(q_3 + q_4 + q_5 + q_6 + q_7) + a_{62}\ L_5 \cos(q_6)$;
- $D_{25} = D_{52} = D_{35} + (a_{64}\ L_5 + a_{74}\ L_5 - a_{52})\ L_3 \cos(q_3 + q_4 + q_5) + (a_{74}\ L_6 - a_{62})\ L_3 \cos(q_3 + q_4 + q_5 + q_6) - a_{72}\ L_3 \cos(q_3 + q_4 + q_5 + q_6 + q_7) + a_{62}\ L_5 \cos(q_6)$;
- $D_{26} = D_{62} = D_{36} + (a_{74}\ L_6 - a_{62})\ L_3 \cos(q_3 + q_4 + q_5 + q_6) - a_{72}\ L_3 \cos(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $D_{27} = D_{72} = D_{37} - a_{72}\ L_3 \cos(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $D_{33} = D_{44} + a_{31}$;
- $D_{34} = D_{43} = D_{44} = a_{41} + D_{55}$;
- $D_{35} = D_{53} = D_{45} = D_{54} = D_{55}$;
- $D_{36} = D_{63} = D_{46} = D_{64} = D_{56}$;
- $D_{37} = D_{73} = D_{47} = D_{74} = D_{57}$;
- $D_{55} = D_{66} + a_{51} - 2a_{63}\ L_5 \cos(q_6) + l_5^2\ a_{64} + l_5^2\ a_{74} - 2a_{73}\ L_5 \cos(q_6 + q_7) + 2a_{74}\ L_5\ L_6 \cos(q_6)$;
- $D_{56} = D_{65} = D_{66} - a_{62}\ L_5 \cos(q_6) - a_{72}\ L_5 \cos(q_6 + q_7) + a_{74}\ L_5\ L_6 \cos(q_6)$;
- $D_{57} = D_{75} = D_{67} - a_{72}\ L_5 \cos(q_6 + q_7)$;
- $D_{67} = D_{76} = a_{71} - a_{72}\ L_6 \cos(q_7)$;
- $D_{66} = a_{71} + a_{61} - L_6^2\ a_{74} - 2a_{72}\ L_6 \cos(q_7)$;
- $D_{77} = a_{71}$;

The h terms are seen to be as follows:

- $h_{111} = 0$;
- $h_{112} = h_{121} = h_{122} - 2\ a_{23}\ L_2 \sin(q_2) + a_{74}\ L_3\ L_5 \sin(q_2 + q_3 + q_4)$;
- $h_{113} = h_{123} = h_{131} = h_{132} = h_{133} = h_{114} + a_{32}\ L_3 \sin(q_3) + a_{32}\ L_2 \sin(q_2 + q_3)$;
- $h_{114} = h_{124} = h_{134} = h_{141} = h_{142} = h_{143} = h_{144} = h_{155} + a_{42}\ L_3 \sin(q_3 + q_4) + a_{42}\ L_2 \sin(q_2 + q_3 + q_4)$;
- $h_{115} = h_{125} = h_{135} = h_{145} = h_{151} = h_{152} = h_{153} = h_{154} = h_{155} = a_{72}\ L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7) + a_{62}\ L_3 \sin(q_3 + q_4 + q_5 + q_6) - a_{74}\ L_3\ L_6 \sin(q_3 + q_4 + q_5 + q_6) + a_{52}\ L_3 \sin(q_3 + q_4 + q_5) - (a_{64} + a_{74})\ L_3\ L_5 \sin(q_3 + q_4 + q_5) - (a_{64} + a_{74})\ L_2\ L_5 \sin(q_2 + q_3 + q_4 + q_5 + q_6) + a_{52}\ L_2 \sin(q_2 + q_3 + q_4 + q_5) + a_{62}\ L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6) - a_{74}\ L_2\ L_6 \sin(q_2 + q_3 + q_4 + q_5 + q_6)$;

- $h_{116} = h_{161} = h_{126} = h_{136} = h_{146} = h_{156} = h_{162} = h_{163} = h_{164} = h_{165} = h_{166} = a_{62} L_5 \sin(q_6) - a_{74} L_6 L_5 \sin(q_6) + a_{72} L_5 \sin(q_6 + q_7) + a_{72} L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7) + a_{62} L_3 \sin(q_3 + q_4 + q_5 + q_6) - a_{74} L_3 L_6 \sin(q_3 + q_4 + q_5 + q_6) + a_{62} L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6) + a_{72} L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7) - a_{74} L_2 L_6 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{117} = h_{127} = h_{137} = h_{147} = h_{157} = h_{167} = h_{171} = h_{172} = h_{173} = h_{174} = h_{175} = h_{176} = h_{177} = a_{72} L_5 \sin(q_6 + q_7) + a_{72} L_6 \sin(q_7) + a_{72} L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7) + a_{72} L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{122} = a_{23} L_2 \sin(q_2) - (a_{74} L_5 + a_{64} L_5 - a_{53}) L_2 \sin(q_2 + q_3 + q_4 + q_5) + (a_{74} L_6 - a_{63}) L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6) - a_{73} L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7) - a_{33} L_2 \sin(q_2 + q_3) - a_{43} L_2 \sin(q_2 + q_3 + q_4) + (a_{34} + a_{44} + a_{54} + a_{64} + a_{74}) L_2 L_3 \sin(q_2) - a_{74} L_3 L_5 \sin(q_2 + q_3 + q_4)$;
- $h_{211} = (a_{74} L_5 + a_{64} L_5 - a_{53}) L_2 \sin(q_2 + q_3 + q_4 + q_5) + (a_{74} L_6 - a_{63}) L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6) - a_{73} L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7) - a_{33} L_2 \sin(q_2 + q_3) + a_{43} L_2 \sin(q_2 + q_3 + q_4) + (a_{34} + a_{44} + a_{54} + a_{64} + a_{74}) L_2 L_3 \sin(q_2)$;
- $h_{212} = h_{221} = h_{222} = 0$;
- $h_{213} = h_{223} = h_{231} = h_{232} = h_{233} = h_{244} + a_{32} L_3 \sin(q_3)$;
- $h_{214} = h_{224} = h_{234} = h_{241} = h_{242} = h_{243} = h_{244} = h_{251} + a_{42} L_3 \sin(q_3 + q_4)$;
- $h_{215} = h_{225} = h_{235} = h_{245} = h_{251} = h_{252} = h_{253} = h_{254} = h_{255} = a_{72} L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7) + a_{62} L_3 \sin(q_3 + q_4 + q_5 + q_6) - a_{74} L_3 L_6 \sin(q_3 + q_4 + q_5 + q_6) + a_{52} L_3 \sin(q_3 + q_4 + q_5) - (a_{64} + a_{74}) L_3 L_5 \sin(q_3 + q_4 + q_5)$;
- $h_{216} = h_{226} = h_{236} = h_{246} = h_{256} = h_{261} = h_{262} = h_{263} = h_{264} = h_{265} = h_{266} = a_{62} L_5 \sin(q_6) - a_{74} L_6 L_5 \sin(q_6) + a_{72} L_5 \sin(q_6 + q_7) + a_{72} L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7) + a_{62} L_3 \sin(q_3 + q_4 + q_5 + q_6) - a_{74} L_3 L_6 \sin(q_3 + q_4 + q_5 + q_6)$;
- $h_{217} = h_{227} = h_{237} = h_{247} = h_{257} = h_{267} = h_{271} = h_{272} = h_{273} = h_{274} = h_{275} = h_{276} = h_{277} = a_{72} L_5 \sin(q_6 + q_7) + L_6 \sin(q_7) + L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{311} = h_{322} + (a_{74} L_5 + a_{64} L_5 - a_{53}) L_2 \sin(q_2 + q_3 + q_4 + q_5) + (a_{74} L_6 - a_{63}) L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6) - a_{73} L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7) - a_{33} L_2 \sin(q_2 + q_3) - a_{43} L_2 \sin(q_2 + q_3 + q_4)$;
- $h_{312} = h_{321} = h_{332} = - a_{33} L_3 \sin(q_3) - a_{43} L_3 \sin(q_3 + q_4) + (a_{74} L_5 + a_{64} L_5 - a_{53}) L_3 \sin(q_3 + q_4 + q_5) + (a_{74} L_6 - a_{63}) L_3 \sin(q_3 + q_4 + q_5 + q_6) - a_{73} L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{313} = h_{314} = h_{315} = h_{323} = h_{324} = h_{325} = h_{331} = h_{332} = h_{333} = h_{334} = h_{335} = h_{341} = h_{342} = h_{343} = h_{344} = h_{345} = h_{351} = h_{352} = h_{353} = h_{354} = h_{355} = 0$;
- $h_{316} = h_{326} = h_{336} = h_{346} = h_{356} = h_{361} = h_{362} = h_{363} = h_{364} = h_{365} = h_{366} = a_{62} L_5 \sin(q_6) + a_{74} L_6 L_5 \sin(q_6) + a_{72} L_5 \sin(q_6 + q_7$;
- $h_{317} = h_{327} = h_{337} = h_{347} = h_{357} = h_{367} = h_{371} = h_{372} = h_{373} = h_{374} = h_{375} = h_{376} = h_{377} = a_{72} L_5 \sin(q_6 + q_7) + L_6 \sin(q_7)$;
- $h_{411} = h_{422} = (a_{74} L_5 + a_{64} L_5 - a_{53}) L_2 \sin(q_2 + q_3 + q_4 + q_5) + (a_{74} L_6 - a_{63}) L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6) - a_{73} L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{412} = h_{421} = h_{422} = - a_{43} L_3 \sin(q_3 + q_4) + (a_{74} L_5 + a_{64} L_5 - a_{53}) L_3 \sin(q_3 + q_4 + q_5) + (a_{74} L_6 - a_{63}) L_3 \sin(q_3 + q_4 + q_5 + q_6) - a_{73} L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{413} = h_{414} = h_{415} = h_{423} = h_{424} = h_{425} = h_{431} = h_{432} = h_{433} = h_{434} = h_{435} = h_{441} = h_{442} = h_{443} = h_{444} = h_{445} = h_{451} = h_{452} = h_{453} = h_{454} = h_{455} = 0$;

- $h_{416} = h_{426} = h_{436} = h_{446} = h_{456} = h_{461} = h_{462} = h_{463} = h_{464} = h_{465} = h_{466} = a_{62}\, L_5 \sin(q_6) + a_{74}\, L_6\, L_5 \sin(q_6) + a_{72}\, L_5 \sin(q_6 + q_7)$;
- $h_{417} = h_{427} = h_{437} = h_{447} = h_{457} = h_{467} = h_{471} = h_{472} = h_{473} = h_{474} = h_{475} = h_{476} = h_{477} = a_{72}\, L_5 \sin(q_6 + q_7) + L_6 \sin(q_7)$;
- $h_{511} = h_{522} + (a_{74}\, L_5 + a_{64}\, L_5 - a_{53})\, L_2 \sin(q_2 + q_3 + q_4 + q_5) + (a_{74}\, L_6 - a_{63})\, L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6) - a_{73}\, L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{512} = h_{521} = h_{522} = (a_{74}\, L_5 + a_{64}\, L_5 - a_{53})\, L_3 \sin(q_3 + q_4 + q_5) + (a_{74}\, L_6 - a_{63})\, L_3 \sin(q_3 + q_4 + q_5 + q_6) - a_{73}\, L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{513} = h_{514} = h_{515} = h_{523} = h_{524} = h_{525} = h_{531} = h_{532} = h_{533} = h_{534} = h_{535} = h_{541} = h_{542} = h_{543} = h_{544} = h_{545} = h_{551} = h_{552} = h_{553} = h_{554} = h_{555} = 0$;
- $h_{516} = h_{526} = h_{536} = h_{546} = h_{556} = h_{561} = h_{562} = h_{563} = h_{564} = h_{565} = h_{566} = a_{62}\, L_5 \sin(q_6) + a_{74}\, L_6\, L_5 \sin(q_6) + a_{72}\, L_5 \sin(q_6 + q_7)$;
- $h_{517} = h_{527} = h_{537} = h_{547} = h_{557} = h_{567} = h_{571} = h_{572} = h_{573} = h_{574} = h_{575} = h_{576} = h_{577} = a_{72}\, L_5 \sin(q_6 + q_7) + L_6 \sin(q_7)$;
- $h_{611} = h_{622} - a_{63}\, L_3 \sin(q_2 + q_3 + q_4 + q_5 + q_6) + (a_{74}\, L_2\, L_6 - a_{73}\, L_2) \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{612} = h_{621} = h_{622} = h_{623} + (a_{74}\, L_6 - a_{63})\, L_3 \sin(q_3 + q_4 + q_5 + q_6) - a_{73}\, L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{613} = h_{614} = h_{615} = h_{623} = h_{624} = h_{625} = h_{631} = h_{632} = h_{633} = h_{634} = h_{635} = h_{641} = h_{642} = h_{643} = h_{644} = h_{645} = h_{651} = h_{652} = h_{653} = h_{654} = h_{655} = a_{74}\, L_6\, L_5 \sin(q_6) - a_{63}\, L_5 \sin(q_6) - a_{73}\, L_5 \sin(q_6 + q_7)$;
- $h_{616} = h_{626} = h_{636} = h_{646} = h_{656} = h_{661} = h_{662} = h_{663} = h_{664} = h_{665} = h_{666} = 0$;
- $h_{617} = h_{627} = h_{637} = h_{647} = h_{657} = h_{667} = h_{671} = h_{672} = h_{673} = h_{674} = h_{675} = h_{676} = h_{677} = a_{72}\, L_6 \sin(q_7)$;
- $h_{711} = h_{722} - a_{73}\, L_2 \sin(q_2 + q_3 + q_4 + q_5 + q_6 + q_7)$;
- $h_{712} = h_{721} = h_{722} = -\, a_{73}\, L_3 \sin(q_3 + q_4 + q_5 + q_6 + q_7) - a_{73}\, L_5 \sin(q_6 + q_7) - a_{73}\, L_6 \sin(q_7)$;
- $h_{713} = h_{714} = h_{715} = h_{723} = h_{724} = h_{725} = h_{731} = h_{732} = h_{733} = h_{734} = h_{735} = h_{741} = h_{742} = h_{743} = h_{744} = h_{745} = h_{751} = h_{752} = h_{753} = h_{754} = h_{755} = -\, a_{73}\, L_5 \sin(q_6 + q_7) - a_{73}\, L_6 \sin(q_7)$;
- $h_{716} = h_{726} = h_{736} = h_{746} = h_{756} = h_{761} = h_{762} = h_{763} = h_{764} = h_{765} = h_{766} = -\, a_{73}\, L_6 \sin(q_7)$;
- $h_{717} = h_{727} = h_{737} = h_{747} = h_{757} = h_{767} = h_{771} = h_{772} = h_{773} = h_{774} = h_{775} = h_{776} = h_{777} = 0$;

where
$a_{11} = I_1 + m_1\, r_1^2$; $a_{12} = a_{13} = m_1\, r_1$; $a_{14} = m_1$; $a_{21} = I_2 + m_2\, (L_2 - r_2)^2$; $a_{22} = a_{23} = m_2\, (L_2 - r_2)$; $a_{24} = m_2$; $a_{31} = I_3 + m_3\, (L_3 - r_3)^2$; $a_{32} = a_{33} = m_3\, (L_3 - r_3)$; $a_{34} = m_3$; $a_{41} = I_4 + m_4\, r_4^2$; $a_{42} = a_{43} = m_4\, r_4$; $a_{44} = m_4$; $a_{51} = I_5 + m_5\, (L_5 - r_5)^2$; $a_{52} = a_{53} = m_5\, (L_5 - r_5)$; $a_{54} = m_5$; $a_{61} = I_6 + m_6\, (L_6 - r_6)^2$; $a_{62} = a_{63} = m_6\, (L_6 - r_6)$; $a_{64} = m_6$; $a_{71} = I_7 + m_7\, r_7^2$; $a_{72} = a_{73} = m_7\, r_7$; $a_{74} = m_7$.

The C terms are found to be as follows:

- $C_1 = m_1\, r_1 \cos(q_1)$;
- $C_2 = -m_2\, g\, (L_2 - r_2) \sin(q_2) - m_1\, g\, (L_1 \sin(q_2) - r_1 \cos(q_1))$;

- C_3 = -(-m_3 g (L_3-r_3) sin (q_3) + m_2 g (-L_3 sin(q_3) + (L_2-r_2) sin (q_2)) + m_1 g (-L_3 sin (q_3) + L_2 sin(q_2) - r_1 cos(q_1)));
- $C_4 = c_3 + m_4\, g\, r_4$ sin (q_4);
- $C_5 = c_4 + m_5\, g$ (L_4-r_4) sin (q_5);
- C_6 = - m_6 g (L_6-r_6) sin (q_6) - m_5 g r_5 sin (q_5) - m_4 g (-r_4 sin (q_4) + L_5 sin(q_5)) - m_3 g (-(L_3-r_3) sin(q_3) + L_5 sin(q_5)) - m_2 g (L_5 sin(q_5) - L_3 sin(q_3) + (L_2-r_2) sin(q_2)) - m_1 g (L_5 sin(q_5) - L_3 sin(q_3) + L_2 sin(q_2)- r_1 cos(q_1);
- $C_7 = m_7\, r_7$ cos(q_7).

***D*, *h* and *C* terms of joint torques for descending gait generation:**
The D and h terms are found to be the same as that of the ascending case and C terms have come out to be as follows:

- $C_1 = m_1\, r_1$ cos (q_1);
- $C_2 = m_2\, g$ (-(L_2-r_2)) sin (q_2) + m_1 g (-L_1 sin(q_2) + r_1 cos (q_1));
- C_3 = (m_3 g (L_3-r_3) sin (q_3) + m_2 g (L_3 sin(q_3) - (L_2-r_2) sin (q_2)) + m_1 g (L_3 sin (q_3) - L_2 sin(q_2) + r_1 cos(q_1)));
- $C_4 = c_3 + m_4\, g\, r_4$ sin (q_4);
- $C_5 = c_4 + m_5\, g$ (L_4-r_4) sin (q_5);
- $C_6 = m_6\, g$ (-(L_6-r_6)) sin (q_6) - m_5 g r_5 sin (q_5) + m_4 g (r_4 sin (q_4) - L_5 sin(q_5)) + m_3 g ((L_3-r_3) sin(q_3) - L_5 sin(q_5)) + m_2 g (-L_5 sin(q_5) + L_3 sin(q_3) - (L_2-r_2) sin(q_2)) + m_1 g (-L_5 sin(q_5) + L_3 sin(q_3) - L_2 sin(q_2) + r_1 cos(q_1);
- $C_7 = m_7\, r_7$ cos(q_7).

References

1. Shih, C.: Ascending and descending stairs for a biped. IEEE Transactions on Systems, Man and Cybermatics - PART-A: Systems and Humans 29(3), 255–268 (1998)
2. Vukobratovic, M., Frank, A.A., Juricic, D.: On the stability of biped locomotion. IEEE Trans. Biomedical Engineering BME 17(1), 25–36 (1970)
3. Mousavi, P.N., Bagheri, A.: Mathematical simulation of a seven link biped robot on various surfaces and ZMP consideration. Applied Mathematical Modeling 31, 18–37 (2007)
4. Yagi, M., Lumelsky, V.: Biped robot locomotion in scenes with unknown obstacles. In: Proc. of IEEE Int'l Conf. on Robotics & Automation, Detroit, Michigan, pp. 375–380 (1999)
5. Juricic, D., Vukobratovic, M.: Mathematical modeling of biped walking systems. ASME Publication 72-WA/BHF13 (1972)
6. Lee, T., Chen, Y.: Minimum-fuel path planning of a 5-link biped robot. In: Proc. of Southern Symposium on System Theory, pp. 20–22 (1988)
7. Channon, P.H., Hopkins, S.H., Phan, D.T.: Derivation of optimal walking motions for a biped walking robot. Robotica 10(2), 165–172 (1992)
8. Roussel, L., Wit, C.: Generation of energy optimal complete gait cycles for biped robots. In: Proc. of IEEE Int'l Conf. on Robotics & Automation, Leuven, Belgium, pp. 2036–2041 (1998)
9. Zheng, Y.F., Shen, J.: Gait synthesis for the SD-2 biped robot to climb sloping surface. IEEE Trans. Robot. Automat. 6, 86–96 (1990)

10. Takanishi, A., Ishida, M., Yamazaki, Y., Kato, I.: The realization of dynamic walking robot WL-10RD. In: Proc. of Int. Conf. Advanced Robotics, pp. 459–466 (1985)
11. Hirai, K., Hirose, M., Haikawa, Y., Takenada, K.: The development of honda humanoid robot. In: Proc. of IEEE Int'l Conf. on Robotics & Automation, pp. 1321–1326 (1998)
12. Dasgupta, A., Nakamura, Y.: Making feasible walking motion of humanoid robots from human motion capture data. In: Proc. of IEEE Int'l Conf. on Robotics & Automation, pp. 1044–1049 (1999)
13. Tsujita, K., Tsuchiya, K., Kawano, Y.: A study on optimal motion of a biped locomotion machine. Artificial Life and Robotics 3(2), 55–60 (1999)
14. Park, J.H., Haan, J., Park, F.C.: Convex optimization algorithms for active balancing of humanoid robots. IEEE Transactions on Robotics 23(4), 817–822 (2007)
15. Pratihar, D.K.: Soft Computing. Alpha Science International, Oxford (2008)
16. Salatian, A.W., Zheng, Y.F.: Gait synthesis for a biped robot climbing sloping surfaces using neural networks, part II: Dynamic learning. In: Proc. of IEEE Int'l Conf. on Robotics and Automation, Nice, France, pp. 2607–2611 (1992)
17. Benbrahim, H., Franklin, J.A.: Biped dynamic walking using reinforcement learning. Robotics and Autonomous Systems 22, 283–302 (1997)
18. Juang, J.G.: Intelligent locomotion control on sloping surfaces. Information Sciences (147), 229–243 (2002)
19. Sabourin, C., Bruneau, O.: Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks. Robotics and Autonomous Systems (51), 81–99 (2005)
20. Yilei, W., Qing, S., Xulei, Y.: Robust recurrent neural network control of biped robot. Journal of Intelligent and Robotic Systems 49, 151–169 (2007)
21. Capi, G., Nasu, Y., Borolli, L., Mitobi, K.: Real time gait generation for autonomous humanoid robots: A case study for walking. Robotics and Autonomous Systems 42, 107–116 (2003)
22. Shih, C.L., Gruver, W.A., Zhu, Y.: Fuzzy logic force control for a biped robot. In: Proc. of IEEE Int'l Symposium on Intelligent Control, Virginia, USA, pp. 269–274 (1991)
23. Park, J.H.: Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots. Fuzzy Sets and systems 134(1), 189–203 (2003)
24. Cuevas, E.V., Zaldivar, D., Rojas, R.: Incremental fuzzy control for a biped robot balance. In: Proc. of Int. Conf. on Robotics and Applications, Cambridge, USA (2005)
25. Zhou, C., Jagannathan, K.: Adaptive network based fuzzy control of a dynamic biped walking robot. In: IEEE Int'l joint symposia on Intelligence and Systems, pp. 109–116 (1996)
26. Zhou, C., Meng, Q.: Dynamic balance of a biped robot using fuzzy reinforcement learning agents. Fuzzy Sets and Systems 134(1), 169–187 (2003)
27. Gu, D., Hu, H.: Evolving fuzzy logic controllers for SONY legged robots. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, p. 356. Springer, Heidelberg (2002)
28. Jha, R.K., Singh, B., Pratihar, D.K.: On-line stable gait generation of a two-legged robot using a Genetic-Fuzzy system. Robotics and Autonomous Systems 53, 15–35 (2005)

5

On Design and Development of an Intelligent Mobile Robotic Vehicle for Stair-Case Navigation

Ranjit Ray, Bikash Bepari, and Subhasis Bhaumik

Abstract. The present chapter focuses on the study of different features of stair-climbing mechanisms for transporting materials autonomously. The authors finally concentrate on arm wheel-based design for stair climbing. An attempt is made to design, develop and construct a mobile robotic vehicle for stair navigation with the following functional attributes: (i) motion on plane surface, (ii) navigation on stair- case, i.e., ascending and descending stairs keeping payload carrier platform parallel to the ground, (iii) autonomous navigation and control including obstacle avoidance and stair detection. The authors attempt to design a skid steer model incorporating the slip-friction for motion on plane surface and sensor-based navigation on stair-case and plane surface. It encompasses necessary controls including vision-assisted controller for stair-case detection and sonar data-based two layered fuzzy logic technique for obstacle avoidance.

1 Introduction

Stairs are the means to ascend and descend from one datum to another, which may be the most common type of burden in the real world as confronted by the robotic vehicle. From the very inception to today's sate of the art flamboyant research, many attempts have been made to mitigate the need of the hour. Still stairs are the main hindrance for smooth navigation by mobile vehicles. This motivates the researcher to develop a means to adroitly navigate stairs.

Autonomous mobile robots are the robots in demand for pioneering the new fields and also most effective means to rejuvenate advanced carrier technologies for future industrial robots. Mobile manipulation is an interesting technique because it allows the robot to enact over a distance. Concurrent advancement in sensor

Ranjit Ray
Central Mechanical Engineering Research Institute, Durgapur (CSIR), India
e-mail: ranjitray@cmeri.res.in

Bikash Bepari
Haldia Institute of Technology, Haldia, India
e-mail: bikashbepari@yahoo.co.in

Subhasis Bhaumik
Bengal Engineering and Science University, Shibpur, Howrah, India
e-mail: sbhaumik_besu@yahoo.co.in

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 87–122.
springerlink.com

and drive technologies enhances the capability of the robots to work in inaccessible or hostile environments.

The prima facie concern to develop an autonomous robotic vehicle is of manifold considerations. They are namely:

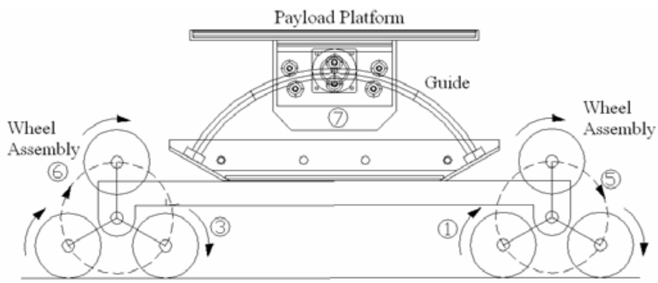
(a) Modes of locomotion: This phenomenon deals with the diversification of the system and the researcher has got an opportunity to cope with the choice of the system, e.g., crawler type, wheel type, legged type or hybrid type.
(b) Kinematics, dynamics and control: This consideration deals with the synthesis of proper mechanisms, which in turn makes the robotic vehicle capable of climbing the stairs after ensuring the dynamic stability of the system. It also aims to determine suitable control strategies of the system.
(c) Incorporation of appropriate intelligence depending on degree of autonomy.
(d) Development of an efficient algorithm for the navigation on both stair-case as well as plane surface.

1.1 Literature Survey

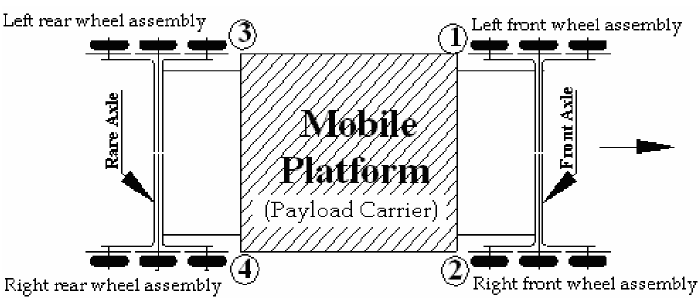
There are various stair climbing mechanisms, which had been implemented and some of them were also patented by several researchers during the past decade. Depending on the type of locomotion, they can be classified into four groups, namely wheeled type, Crawler type, legged type and Hybrid type. A number of stair climbing vehicles had been developed in the past, for example, Terrain Adaptive Quadru-Track (TAQT) Vehicle [1], HELIOS-V[2], and others. In 2000, Häussler et al. [3] developed and patented a four wheel vehicle having a pair of climbing arms. Cox et al. [4] invented a single three wheel cluster stair-climber with articulated front and rear sliders to maintain balance during stair navigation. Tamagawa University, Japan in collaboration with Tomo Co. Ltd, Japan developed a dual cluster front articulated stair-climber [5]. Michaelsen and Popp of the Institute für Robotik (IFR), Germany [6] could develop an autonomous four wheeled stair climber with lift mechanism based on a slider-crank linkage on both front and rear sides. Morales et al. [7] in the School of Industrial Engineering, University of Castilla-La Mancha, Spain, manufactured a four wheeled reconfigurable stair climber with two single slider-crank linkage on the front and rear sides for support and balancing. Arm wheel-based stair climber was proposed by Takano and Odawara [8], Ray et al. [9], and Dalvand et al. [10]. Crawler type vehicle with two active shorter track sections arms for extra gripping force while negotiating the stair-case had been developed by Helmick et al. [11], Hirose et al., [12], Matthies et al. [13]. Tank-like vehicle to climb multiple flights of stairs had been implemented by Steplight et al. [14], Fair and Miller [15]. Quigg designed and patented an innovation vehicle with modified tracks [16]. A number of hexapod robots with stair climbing ability have been developed in the recent past. Among them, some of the robots are very much popular like RHex [17] at McGill University, Canada and ASTERISK [18] at Osaka University, Japan. A number of biped robots could successfully climb the stairs. The Honda bipeds: *ASIMO* could climb quasi-statically [19]. Some other renowned biped robots are QRIO of Sony [20], WABIAN RIII of Waseda University, Tokyo [21], Johnnie of Technische Universitaet Muenchen, Munich [22]. Small robots called *Mini-Whegs* was developed by Nelson et al. [23] at Case Western Reserve University, Cleveland, Ohio, USA. This was derived from larger Whegs series of robots having three spoked appendages, called *whegs*, which benefit from abstracted cockroach locomotion principles. Murray and Ishimatsu [24] of Nagasaki University, Japan proposed a stair-climbing wheelchair mechanism with high single step capability. *Zero Carrier*, a most rugged but simple type of legged wheel stair climbing mobile vehicle, was developed by Yuan and Hirose [25] at Tokyo Institute of Technology. Moreover, a four-leg-wheel hybrid locomotion had been developed at Tokyo Institute of Technology, Japan [26].

1.2 Genesis

From the literature survey, it was felt that wheeled vehicles have greater movability on plane and if by any engineering means, the wheel motion is modified to easily negotiate uneven surfaces without using any complex control system then the superior mobility and maneuverability can be achieved. It has also been found that whenever the robotic vehicle moves over a stair-case, its payload carrier platform remains oriented along the inclination of stair angle, which ultimately creates problems related to orientation and stability. In case of carrying disabled persons, it creates discomfort to the passenger. This has encouraged the researchers to re-investigate the wheel-type motion for stair-case navigation.



(a)



(b)

Fig. 1 Arm Wheel-based Mobile Robot

In the present chapter, the authors have concentrated on arm wheel-based design for stair climbing (refer to Fig. 1) and vision-based stair navigation. The three wheels of each wheel cluster denoted as 1-4 in Fig. 1(b), are powered simultaneously from a single actuator, i.e., four individual drives have been employed for the wheel clusters of front and rear axle. During motion on flat surface, eight wheels are in contact with the floor through which the required tractive power is transmitted and the rest four wheels rotate freely. However, in order to navigate stairs and/or steps, additional mechanisms are provided. Another two drive arms at the front and rear sides (5-6 as shown in Fig. 1(a)) are actuated for stair ascending and descending motion. Besides the stair climbing features, an additional mechanism is also incorporated to make the platform horizontal. For accomplishing this task, an additional actuator (7) is also employed, as shown in Fig. 1(a).

The mechanism of stairs ascending motion has been decomposed into three distinguished steps:

(a) Detection of staircase using sensors' input and navigate the vehicle towards the stair,
(b) Vehicle would be in running mode until it is aligned with the staircase,
(c) After aligning with the staircase, both the actuators (that is, 5 and 6 in Fig. 1) for the arm rotation start exhaling until vehicle reaches the top of the stair.

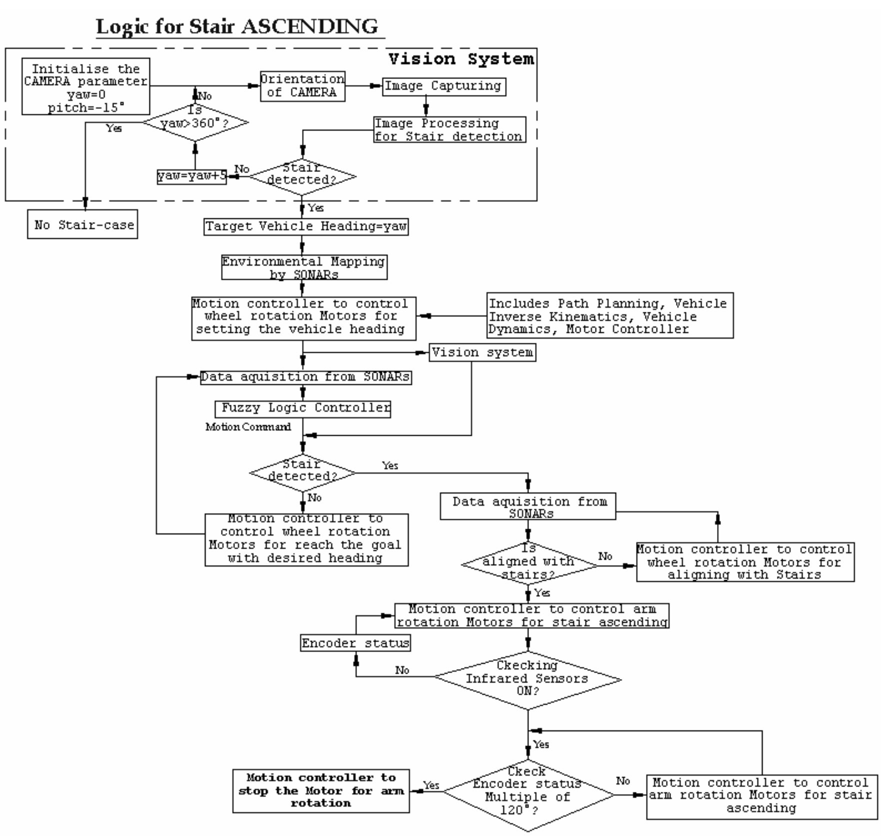


Fig. 2 Control logic flow-chart for stair climbing

The mechanism of stair-descending is quite similar with that of stair-ascending except the third step. Actual stair descending motion starts due to the motion of the actuators for arm rotation, but simultaneously, four wheel actuators rotate in reverse direction with a very slow speed to impart anti-gravitational pull, which in turn, helps the vehicle to get rid of sudden fall phenomenon.

The authors developed a stair-case recognition technique from an image acquired through a camera and a fuzzy logic-based obstacle avoidance scheme for the autonomous navigation on both plane surface as well as stairs. Flow-charts for ascending and descending the stairs are shown in Figs. 2 and 3, respectively.

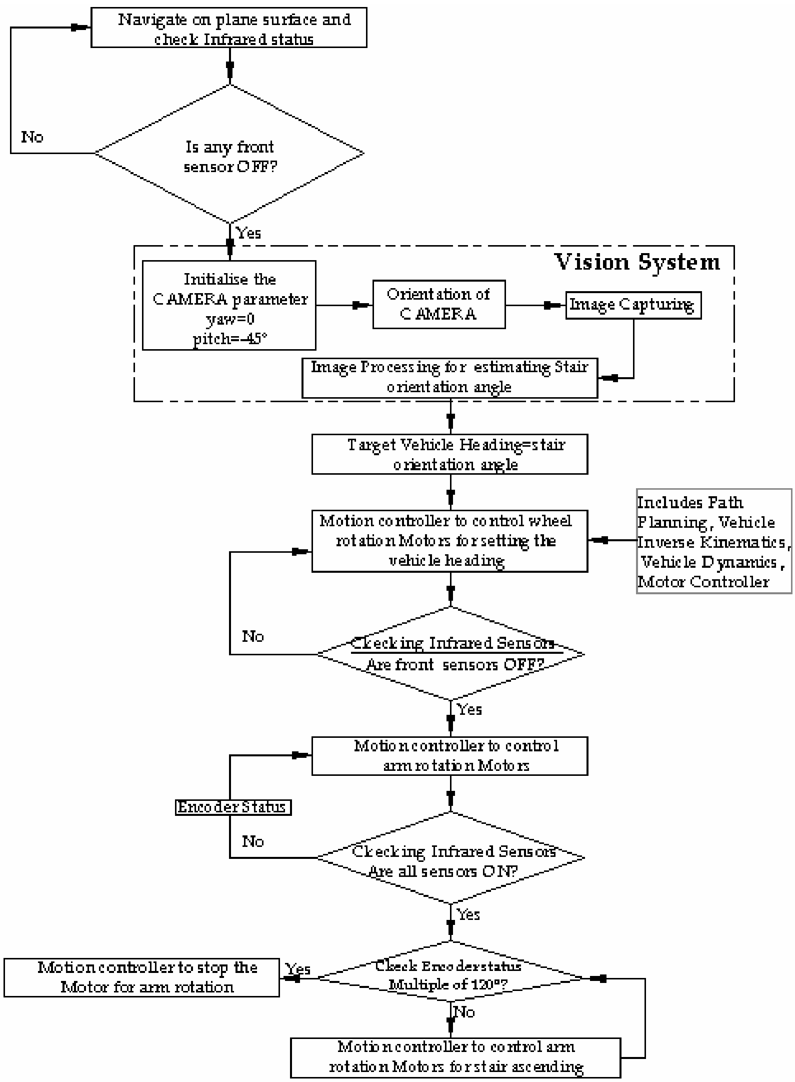


Fig. 3 Control logic flow-chart for stair descending

2 Kinematics, Dynamics and Control

Due to lateral skidding, velocity constraints occurring in Skid Steer Vehicles (SSV) are quite different from the other mobile robots [46], which have no skidding. Control of these types of robots is yet in infancy, which demands for development of an adept control algorithm to mitigate the problems associated with dynamic stability of the system.

A fixed frame (X, Y) and a robot body frame (x, y) have been defined as depicted in Fig. 4. Let $(\dot{x}, \dot{y}, \dot{\varphi})$ be the longitudinal, lateral and angular velocities of the robot about the body frame. It has been revealed from literature [27-29,47] that the instantaneous center must shift forward to the vehicle centroid by an amount of dCx that depends on the vehicle's lateral acceleration $\ddot{y}$. This longitudinal shift is required to achieve a net lateral force to accelerate the vehicle to move forward the instantaneous center of rotation and also to reduce the resisting yaw moments. It has also been shown in literature [27-28] that x-components of the wheel velocities must be constant along each side wheels, and that the y-components of the velocities of wheels depend only on the longitudinal position of the wheels (x-dCx).

Since the center of mass of the vehicle is above the plane of the wheels' forces, the moment of the centrifugal forces during turning, will increase the wheel contact pressure for the front wheels. Thus, the contact pressure of the front wheels

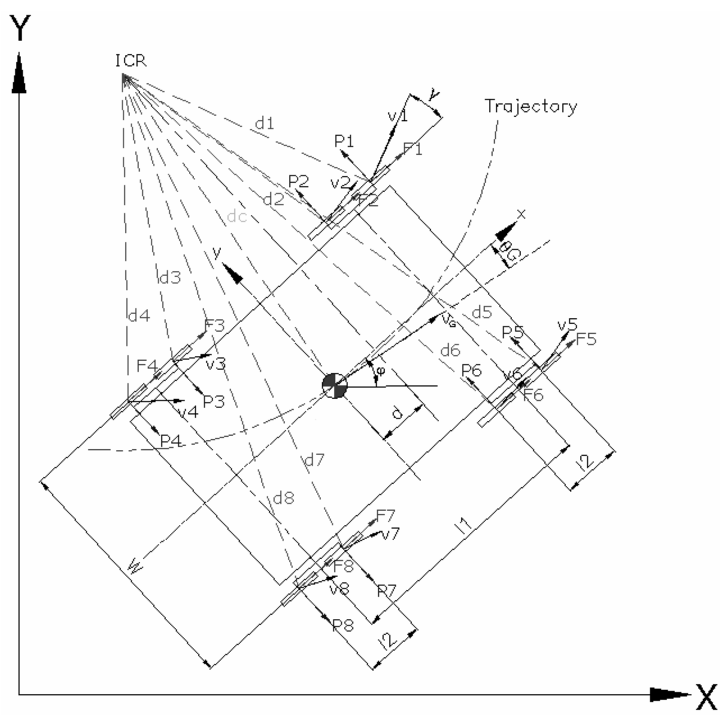


Fig. 4 Model for the Skid Steer Vehicle (SSV)

will be more compared to that of the rear wheels. Similarly, the contact pressure for the inside wheels will be more compared to that for the outside wheels. Hence, without the loss of generality, the following assumptions have been made:

(a) Center of mass of the robot is located at the geometric center of the body frame.
(b) The thickness of the wheel is neglected and it is assumed to be in contact with the ground at a point.
(c) The resistive force is negligible.
(d) Four wheels of each side rotate at the same speed.
(e) The normal forces at the wheel/ground contact points are equally distributed among eight wheels during motion.
(f) The robot is running on a flat surface and eight wheels are always in contact with the ground surface.

As the vehicle is symmetrical in shape, the geometrical parameters, such as distance between the front and rear axles along x direction (l1), that between two wheels in a cluster (l2) and that between inner and outer wheels of the robotic vehicle (W) are sufficient to describe the vehicle geometry for kinematic analysis. The angular wheel velocity is denoted by ω_i and the velocities of the wheel contact points are denoted by v_i, where, i = 1, 2,…,8. Since each wheel of the same cluster is powered by the same actuator, it implies that $\omega_1 = \omega_2$, $\omega_3 = \omega_4$, $\omega_5 = \omega_6$ and $\omega_7 = \omega_8$. According to the assumption number (d), it can be written that $\omega_{inner} = \omega_1 = \omega_2 = \omega_3 = \omega_4$ and $\omega_{outer} = \omega_5 = \omega_6 = \omega_7 = \omega_8$. The longitudinal and lateral forces at each wheel's contact point are F_i and P_i, respectively, where i = 1, 2,· · · , 8. The velocity of the robot mass center is denoted as v_G. It is straightforward to calculate the relationship of the robot velocities and accelerations in both frames, as given below.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \end{bmatrix} = R^T(\varphi) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \;\&\; \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{\varphi} \end{bmatrix} = R^T(\varphi) \begin{bmatrix} \ddot{x} - \dot{y}\dot{\varphi} \\ \ddot{y} - \dot{x}\dot{\varphi} \\ \ddot{\varphi} \end{bmatrix}, \text{ where } R(\varphi) = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

In Fig. 4, the radius vector for each wheel and CG are defined as $d_i = [\, d_{ix} \;\; d_{iy} \,]$ and $d_C = [\, d_{Cx} \;\; d_{Cy} \,]$, respectively. Since the motion of the vehicle can be described by $(\dot{\varphi}, v_G)$, the condition for the y-velocity of the point (dCx, 0, 0) in global frame to vanish leads to the relation as shown below [27,29].

$$d_{Cx} = -\frac{\dot{y}}{\dot{\varphi}} \text{ and } d_{Cy} = \frac{\dot{x}}{\dot{\varphi}} \quad (2)$$

The coordinates of vectors d_i satisfy the following relationships:

$$d_{1x} = d_{5x} = \frac{l1 + l2}{2} - d_{Cx} \qquad d_{2x} = d_{6x} = \frac{l1 - l2}{2} - d_{Cx}$$
$$d_{3x} = d_{7x} = -\frac{l1 - l2}{2} + d_{Cx} \qquad d_{4x} = d_{8x} = -\frac{l1 + l2}{2} + d_{Cx} \quad (3)$$

and

$$d_{1y} = d_{2y} = d_{3y} = d_{4y} = d_{Cy} - \frac{W}{2}$$
$$d_{5y} = d_{6y} = d_{7y} = d_{8y} = d_{Cy} + \frac{W}{2} \tag{4}$$

For a given mass center velocity v_G and yaw rate $\dot{\varphi}$, the longitudinal velocities v_{ix} of the wheel/ground contact points are

$$v_{1x} = v_{2x} = v_{3x} = v_{4x} = \dot{x} - \frac{W}{2}\dot{\varphi}$$
$$v_{5x} = v_{6x} = v_{7x} = v_{8x} = \dot{x} + \frac{W}{2}\dot{\varphi} \tag{5}$$

Taking r as wheel radius, the longitudinal wheel slips s_i can be defined as

$$s_i = \frac{r\omega_i - v_{ix}}{r\omega_i} = \frac{\Delta v_{ix}}{r\omega_i}, \text{ where i=1,2,....,8 and } \Delta v_{ix} = r\omega_i - v_{ix} \tag{6}$$

Based on the geometry, the following equations can be derived as

$$\Delta v_{ix} = \left(d_{Cy} - \frac{W}{2}\right)\dot{\varphi} \text{ and } \Delta v_{jx} = \left(d_{Cy} + \frac{W}{2}\right)\dot{\varphi} \tag{7}$$

where i=1, 2, 3, 4 and j=5, 6, 7, 8.

From Eq.s (6) and (7), it can be obtained that

$$\omega_{inner} = \frac{1}{r} \cdot \left[\dot{x} + \left(d_{Cy} - W\right)\dot{\varphi}\right] \text{ and } \omega_{outer} = \frac{1}{r} \cdot \left[\dot{x} + \left(d_{Cy} + W\right)\dot{\varphi}\right]. \tag{8}$$

Note that $s_{inner} = s_1 = s_2 = s_3 = s_4$ and $s_{outer} = s_5 = s_6 = s_7 = s_8$ as per the assumption mentioned at (d).

To complete the kinematics model of the SSV, the non-holonomic constraint as written in Eq. (2) is rewritten in the Pfaffian form:

$$\left[-sin(\varphi)\ cos(\varphi)\ d_{Cx}\right] \cdot \left[\dot{X}\ \ \dot{Y}\ \ \dot{\varphi}\right]^T = A(q)\dot{q} = 0 \tag{9}$$

Since the generalized velocity $\dot{q}$ is always in the null space of A, it can be written as [33] $\dot{q} = S(q)\eta$, where $S^T(q) \cdot A^T(q) = 0$, $S(q) = \begin{bmatrix} cos\varphi & d_{Cx} sin\varphi \\ sin\varphi & -d_{Cx} cos\varphi \\ 0 & 1 \end{bmatrix}$ (10)

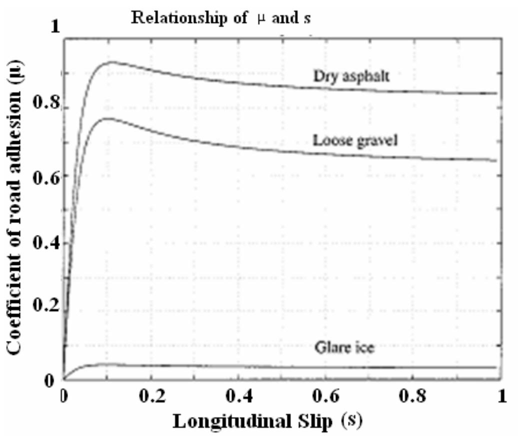
and the new control input $\eta = \begin{bmatrix} \dot{x} \\ \dot{\varphi} \end{bmatrix}$.

Longitudinal and lateral reaction forces are acting against the motion as shown in Fig. 4. It has been considered that the longitudinal friction forces $F_i = N_i.\mu_i$ for the i^{th} wheel, where μ_i is the friction coefficient and N_i is the normal force. It is also considered that the friction coefficient μ is a function of the longitudinal slips [31-33].

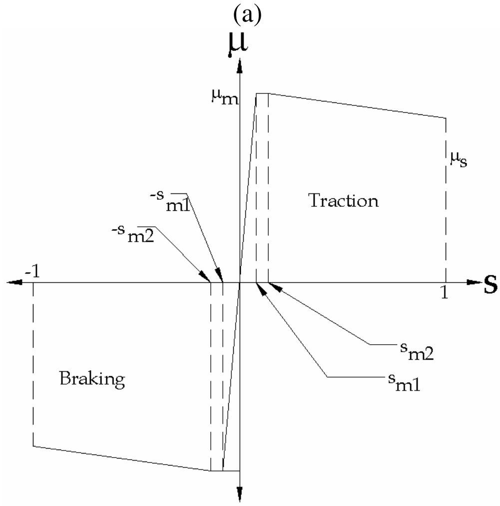
The μ-s curve is obtained as shown in Fig. 5(a) by fitting the experimental data [33]. Here, a linear approximation of the μ-s curve has been considered as shown in Fig. 5(b). For the traction case, the frictional coefficient μ is approximated by the following functions:

$$\mu(s)=\begin{cases} Ks & s\in[0,s_{m1}] \\ \mu_m & s\in[s_{m1},s_{m2}] \\ \mu_m-\dfrac{\mu_m-\mu_s}{1-s_{m2}}(s-s_{m2}) & s\in[s_{m2},1] \end{cases} \tag{11}$$

where K is the friction stiffness coefficient, $s_{m1}$and s_{m2} are the two longitudinal slip values which are the two extreme ends of the zone, where maximum wheel/ground



(a)



(b)

Fig. 5 (a) Variations between coefficient of road adhesion μ and longitudinal slips [8], (b) linear approximation of μ-S relationship friction coefficient μ_m occurs, and μ_s is the longitudinal wheel/ground sliding friction coefficient. Eq. (11) can still be used to calculate the magnitude of the friction coefficients for the braking case while the longitudinal slip s < 0 and $\mu < 0$ [30]. With such simplifications, Eq. (11) is rewritten as $\mu(s) = \mu(s)\cdot sgn(s)$, where function sgn(x)=1 if $x \geq 0$ and sgn(x)= −1 if x< 0.

The longitudinal friction force F_i and lateral friction force P_i are dependent on each other and their magnitudes form a friction force circle, namely, $F_i = F_{ir} cos\gamma_i$, $P_i = F_{ir} sin\gamma_i$, where F_{ir} is the resultant maximum friction force and γ_i is the slip angle at the i[th] wheel (Fig. 4). It is to be noted that the longitudinal friction force $F_i = N_i . \mu_i(s_i)$ and the lateral friction force P_i is written as

$$P_i = F_i \tan\gamma_i \qquad \text{where } i=1,\ldots\ldots,8 \tag{12}$$

For the eight-wheel robot, the normal load was assumed at each wheel $N_i = \frac{mg}{8}$, which is constant and the ground soil conditions are the same for the eight wheels. Due to the fact that $s_1 = s_2 = s_3 = s_4$, $s_5 = s_6 = s_7 = s_8$, it is obtained that $F_1 = F_2 = F_3 = F_4$, $F_5 = F_6 = F_7 = F_8$. Using the relationship given by Eq. (3), (4) and (12), it has been obtained that

$$P_i = (-1)^n F_i sgn(s_i) \frac{d_{ix}}{d_{iy}},$$

where i=1,…,8 and n=1 for i=3,4,7,8 and n=2 for i=1,2,5,6.

Hence, the dynamic model of the vehicle is written in the frame (x, y) and in continuous time as

$$\begin{aligned}
&m\ddot{x} = 4\cdot[sgn(s_1)\cdot F_1 + sgn(s_5)\cdot F_5] \\
&m\ddot{y} = \sum_{i=1}^{8} P_i \, sgn(s_i) \\
&I\ddot{\varphi} = 4\cdot[-sgn(s_1)\cdot F_1 + sgn(s_5)\cdot F_5]\frac{W}{2} - \\
&\left[(-P1 + P4 - P5 + P8).\frac{l1+l2}{2} + (-P2 + P3 - P6 + P7).\frac{l1-l2}{2}\right]
\end{aligned} \tag{13}$$

Using the Lagrange-Euler formula, the dynamic equation of the robotic vehicle is obtained as

$$M(q)\ddot{q} + R(q,\dot{q}) = B(q)\tau, \tag{14}$$

where $M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}$

$$R(q,\dot{q}) = [F_{rx}(\dot{q}) \; F_{ry}(\dot{q}) \; M_r(\dot{q})]^T$$

$$F_{rx}(\dot{q}) = m\ddot{X} = m\ddot{x}\cdot cos\varphi - m\ddot{y}\cdot sin\varphi$$

$$F_{ry}(\dot{q}) = m\ddot{Y} = m\ddot{x}\cdot sin\varphi + m\ddot{y}\cdot cos\varphi$$

$$M_r(\dot{q}) = I\ddot{\varphi}$$

$$B(q)=\frac{1}{r}\begin{bmatrix} cos\varphi & cos\varphi \\ sin\varphi & sin\varphi \\ -\frac{W}{2} & \frac{W}{2} \end{bmatrix} \text{ and } \tau=\left[\tau_{a1}+\tau_{a2}\ \ \tau_{a3}+\tau_{a4}\right]^T=\begin{bmatrix} \tau_1+\tau_2+\tau_3+\tau_4 \\ \tau_5+\tau_6+\tau_7+\tau_8 \end{bmatrix},$$

where $\left[\tau_{a1}\ \ \tau_{a2}\ \ \tau_{a3}\ \ \tau_{a4}\right]$ denotes the torques produced by cluster wheels on the left front and rear, and right front and rear sides of the vehicle, respectively.

It should be noted that Eq.(14) describes the dynamics of a free body only and does not include the nonholonomic constraint Eq.(9). Therefore, a constraint has to be imposed on Eq.(14). A vector of Lagrange multipliers λ is introduced as follows:

$$M(q)\ddot{q}+R(q,\dot{q})=B(q)\tau+A^T(q)\lambda \tag{15}$$

For control purposes, it would be more suitable to express Eq. (15) in terms of the internal velocity vector η. Therefore, Eq. (15) is multiplied from the left by $S(q)$, which results in

$$S^T(q)M(q)\ddot{q}+S^T(q)R(\dot{q})=S^T(q)B(q)\tau+S^T(q)A^T(q)\lambda \tag{16}$$

After taking the time derivative of Eq. (16), it can be obtained

$$\ddot{q}=\dot{S}(q)\eta+S(q)\dot{\eta} \tag{17}$$

Next, using (17) and (10) in (16), the dynamic equations become

$$\overline{M}\dot{\eta}+\overline{C}\eta+\overline{R}=\overline{B}\tau \tag{18}$$

where $\overline{M}=S^T(q)\cdot M(q)\cdot S(q)=\begin{bmatrix} m & 0 \\ 0 & md_{Cx}^2+I \end{bmatrix}$

$$\overline{C}=S^T(q)\cdot M(q)\cdot \dot{S}(q)=md_{Cx}\begin{bmatrix} 0 & \dot{\varphi} \\ -\dot{\varphi} & d_{Cx} \end{bmatrix}$$

$$\overline{R}=S^T(q)R(\dot{q})=\begin{bmatrix} F_{rx}(\dot{q}) \\ d_{Cx}F_{ry}(\dot{q})+M_r \end{bmatrix}$$

$$\overline{B}=S^T(q)B(q)=\frac{1}{r}\begin{bmatrix} 1 & 1 \\ -\frac{W}{2} & \frac{W}{2} \end{bmatrix}$$

Using this dynamic model (18), control law has been evaluated as described in [44].

3 Dynamic Model for Stair Climbing

The algorithms as described indicate that both the actuators for arm rotation rotate simultaneously. The vehicle's kinematic configurations can be described as a simple four-bar mechanism as shown in Fig. 6. L_i defines the cluster radius where l_1 is the length of body, θ_i defines the vector space angle of the ith link and l is the distance between two wheels which are in contact with the stairs. Generally, this distance remains theoretically constant unless the step heights of the stairs are unequal. Here, the step height has been assumed to be constant and hence $l_1 = l$. Since the whole system has only one degree of freedom, in true sense, one actuator would act as a redundant input. Basically, two actuators share the single degree of freedom with the same velocity and acceleration.

A dynamic problem can be solved by static analysis, when inertia forces and moments are taken into account. All the members of a mechanism can be considered to be in equilibrium under the combined effect of all the applied and reactive forces, including its own inertia [34].

Neglecting the dynamic effects of the payload orientation mechanism, the free-body diagram of all the links is shown in Fig. 6(b). m_i is the mass of the i^{th} link, a_{ix} and a_{iy} are the x and y components of acceleration of the i-th link. P_{ax}, P_{ay}, P_{bx}, P_{by}, N_1, F_1, N_2, F_2 are considered as the reaction forces.

Total torque required for satisfying the kinematics is divided into two parts: T_1 and T_2, such that forces

$$P_{ax} = P_{bx}\text{, as } -P_{ax} + P_{bx} = 0 \tag{19}$$

The dynamic equation for the complete system is derived as

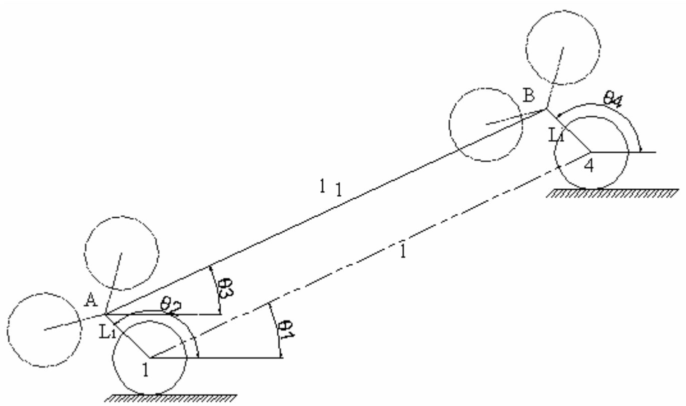
$$\{F\} + [Q]\{P\} = \{H\} \tag{20}$$

where, $$\{F\} = \begin{bmatrix} 0 \\ -m_2 g \\ 0 \\ 0 \\ -m_3 g \\ 0 \\ 0 \\ -m_4 g \\ 0 \\ 0 \end{bmatrix}; \quad \{P\} = \begin{bmatrix} F_1 \\ N_1 \\ P_{ax} \\ P_{ay} \\ T_1 \\ P_{bx} \\ P_{by} \\ F_2 \\ N_2 \\ T_2 \end{bmatrix}; \quad \{H\} = \begin{bmatrix} m_2 a_{2x} \\ m_2 a_{2y} \\ m_2 \ddot{\theta}_2 k_2^2 \\ m_3 a_{3x} \\ m_3 a_{3y} \\ m_3 \ddot{\theta}_3 k_3^2 \\ m_4 a_{4x} \\ m_4 a_{4y} \\ m_4 \ddot{\theta}_4 k_4^2 \\ 0 \end{bmatrix}$$

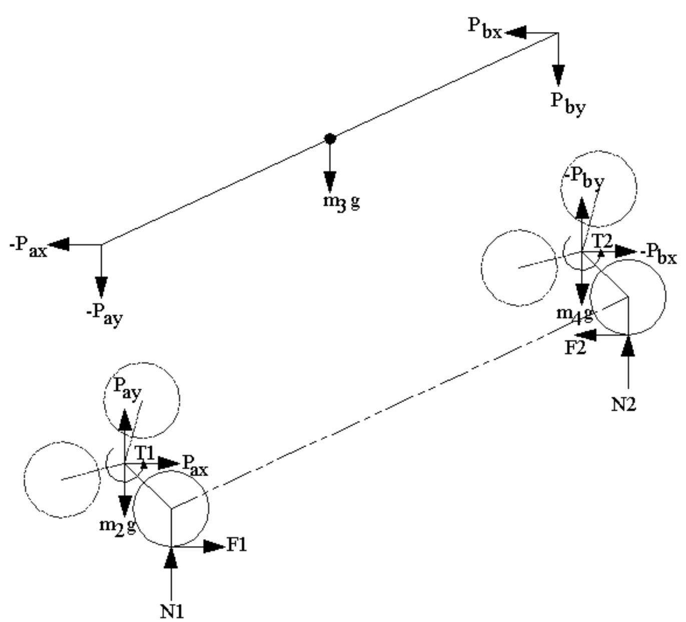
and

$$Q = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ Li\sin\theta_2 & -Li\cos\theta_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -l_1\sin\theta_3/2 & l_1\cos\theta_3/2 & 0 & l_1\sin\theta_3/2 & -l_1\cos\theta_3/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -Li\sin\theta_4 & -Li\cos\theta_4 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

By solving equation (20), torque T_1 and T_2 are derived, where the inputs to the systems are $\theta_2, \dot{\theta}_2, \ddot{\theta}_2$.



(a)



(b)

Fig. 6 Dynamic Model: (a) Kinematic Model, (b) Free-Body diagram during stair climbing

4 Modeling of the Payload Platform Orientation Mechanism

The payload carriage is moving over a curved guide using belt pulley drive (refer to Fig. 7). For finding out the geometry of the guide, the following assumptions are made:

(a) Unwrapped / wrapped angle of left hand side idler pulley is almost equal to wrapped / unwrapped angle of right hand side idler pulley.
(b) The two ends of the timer belt are connected with the table at a point (Apex).

From the assumption (a),

$$LL1 + LL2 = KK(constant) \tag{21}$$

From the geometry of the carriage,

$$KK = 2 \times \sqrt{PH^2 + Len^2} \tag{22}$$

Now for any position of the Apex (x,z)

$$LL1 = \sqrt{(Len + x)^2 + \left(z^2 - R_p^2\right)} \tag{23}$$

and

$$LL2 = \sqrt{(Len - x)^2 + \left(z^2 - R_p^2\right)} \tag{24}$$

where R_p is the radius of the idler pulley.

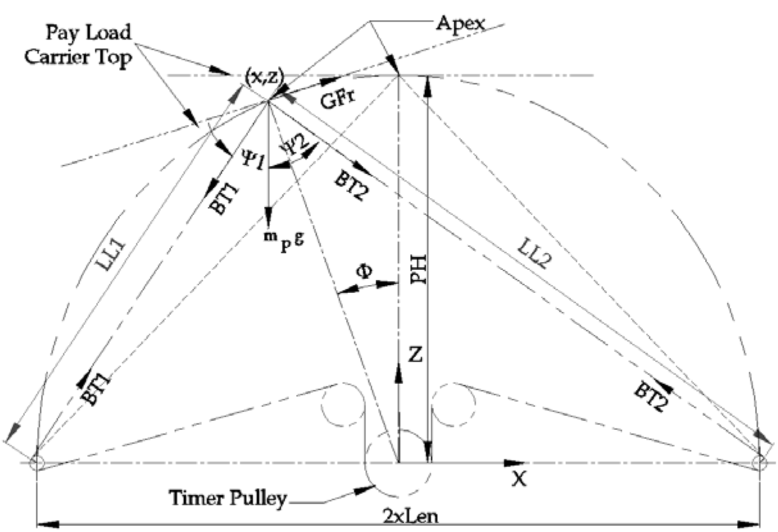


Fig. 7 Geometrical Diagram of the Payload Carrier Turning Mechanism

From Eqs. (21), (22), (23) and (24), the equation of the guide curve is derived as

$$\frac{X^2}{A^2}+\frac{Z^2}{B^2}=1 \tag{25}$$

where $A^2=\dfrac{KK^4+4\times KK^2\times R_p{}^2-4\times KK^2\times len^2}{16\times PH^2}$

and $B^2=\dfrac{KK^4+4\times KK^2\times R_p{}^2-4\times KK^2\times Len^2}{4\times KK^2}$

The above Eq. (25) describes the curved profile, which follows an elliptical expression. Hence, for rotating the payload platform to a particular angle, the angle measured by the tilt sensor is the direct input command that has to be executed by the actuator. Obviously, this simplifies the control algorithm.

For calculating the required torque for aligning the pay load table at an angle Φ (considering statically stable equilibrium), the forces acting at the Apex (x, z) is resolved, where BT1 and BT2 are the pulley tensions in slack and tight side respectively, Fr is the friction force exerted on the guide roller, m is the mass of payload table including payload. From the geometry, ψ1 and ψ2 are calculated as

$$tan\psi a=\frac{Bsin\varphi}{Len-Acos\varphi} \text{ and } tan\psi a=\frac{Bsin\varphi}{Len+Acos\varphi} \tag{26}$$

Forces acting in x and z directions are derived by resolving the different forces as

$$\begin{bmatrix} cos\psi 1 & cos\psi 2 \\ sin\psi 1 & -sin\psi 2 \end{bmatrix}\begin{bmatrix} BT1 \\ BT2 \end{bmatrix}=\begin{bmatrix} GFrsin\varphi - mg \\ GFrcos\varphi \end{bmatrix}, \text{where } GFr=\mu mg \tag{27}$$

Hence, the required torque is $T=(BT2 - BT1)r_p$, where r_p is the radius of the timer pulley.

5 Fuzzy Logic Controller

Fuzzy logic has become an important technique in machine control. However, the term itself inspires certain skepticism. But, fuzzy logic is actually very straightforward. It is a way of interfacing inherently analog processes, which move through a continuous range of values, to a digital computer, that likes to see things as well-defined discrete numeric values.

An intelligent navigation control system [45] has been developed for the mobile robot. The controller is capable of dealing with the uncertainties of the working environment. A Fuzzy Logic Controller (FLC) processes the environmental data acquired by the ultrasonic range finders and finds the safe path towards the goal by avoiding obstacles. The decision of the FLC is conveyed to the vision-based controller for controlling the speed of the motors.

The proposed FLC controller has two layers. The first layer has two inputs and one output. The two input variables are (i) *obstacle distance* from the vehicle and (ii) *obstacle angle*, i.e., the angle of obstacle from the robotic vehicle. The second layer fuzzy logic controller receives the deviation angle along with the three other sides collision possibilities as inputs and the outputs are the translational velocity and rotational velocity, which are responsible for moving the vehicle in forward direction. Depending on the physical dimension of the vehicle, placement of eight ultrasonic sensors and their related fuzzy partitions have been considered judiciously as shown in Fig. 8. One sensor each is placed straight at the front and rear (sensor number 2 at front side and sensor number 6 at rear side) and two sensors are at an angle of -25° (at sensor numbers 1 and 5) and +25° (at sensor numbers 3 and 7), on the right hand side. The other two sensors are placed on the two sides of the vehicle at the symmetric positions (sensor numbers 4 and 8).

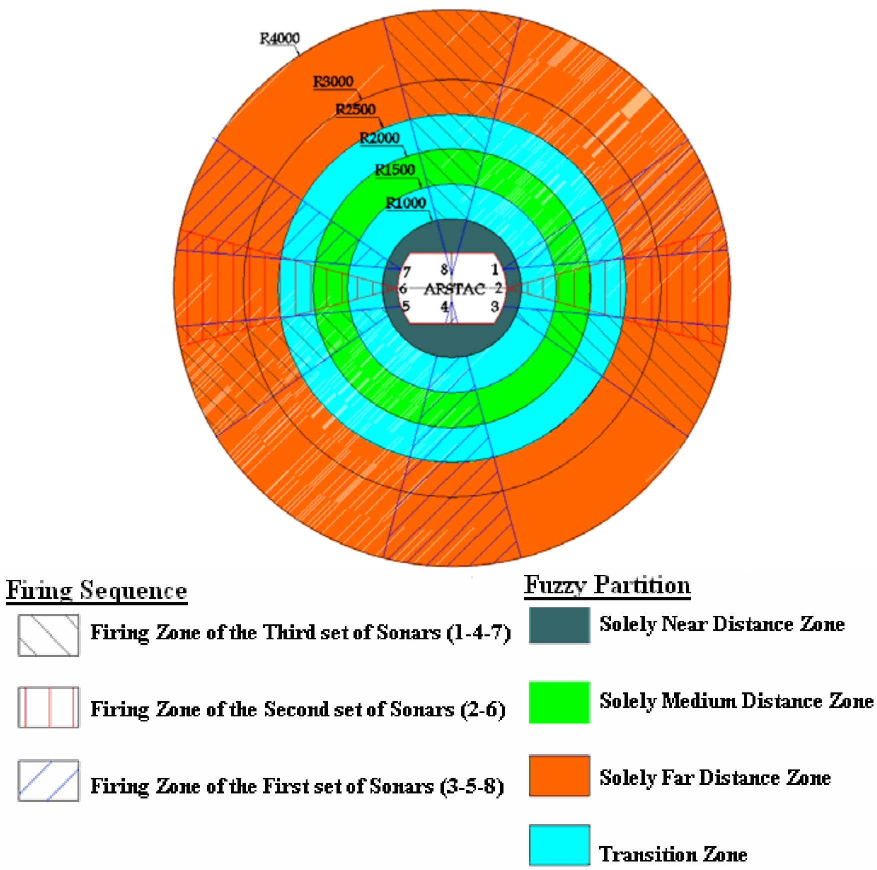


Fig. 8 Sonar firing sequence and fuzzy partition zone

To minimize the problem of interference, all the sensors are not fired simultaneously. One switch circuit has been designed for switching the sensors located at 1, 4 and 7 at a time and then, switching on the two sensors at 2 and 6 followed by switching for rest of the sensors 3, 5 and 8. Hence, at any instant, only one set remains in working condition and the chances of mutual interference are minimized. The data of the ultrasonic range finders are noisy and erroneous. Hence, the sensory data are filtered before their use. Through prolonged experimentation, filter has been designed for elimination of spurious data.

Obstacle distance and *obstacle angle* are fuzzified as shown in Figs. 9 and 10, respectively. The output of the fuzzy controller: *deviation* is fuzzified as shown in Fig. 11. After fuzzy partitioning of the input and output variables, a rule base for the first layer has been developed as shown in Table 1. The rule base and the database are correlated. The designed rule base is tested by computer simulations and is modified along with the database.

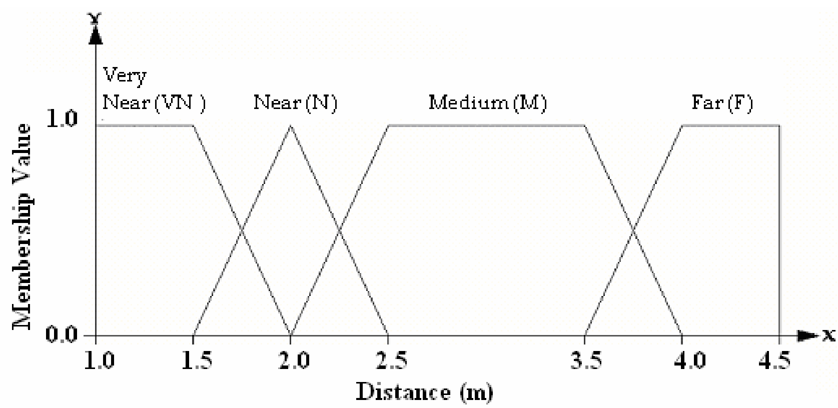


Fig. 9 Fuzzy partition of obstacle distance

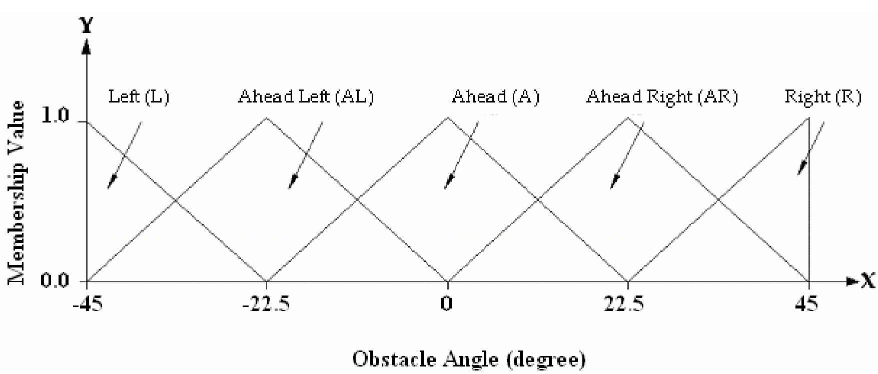


Fig. 10 Fuzzy partition of obstacle angle

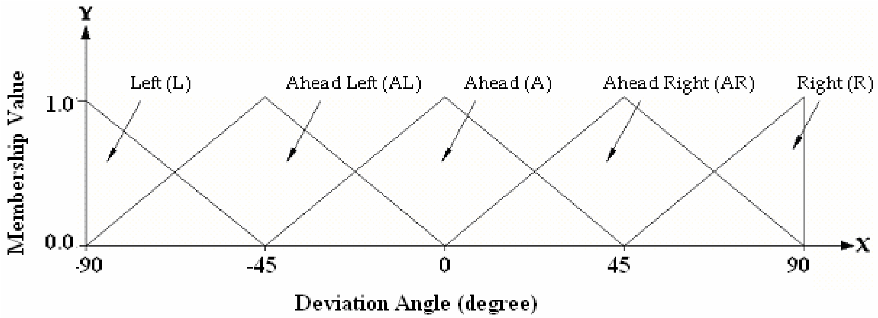


Fig. 11 Fuzzy partition of deviation angle

First layer of the FLC provides the deviation angle of the vehicle from input sensory data. But, the FLC does not take into account the current orientation of the vehicle. The deviation actually required to avoid the obstacle is calculated based on the output of the FLC, orientation of the robot in the previous position measured by the compass and the current position of the robot. The speed of the vehicle is decided and accordingly, the *delay* is also decided. The sampling time is taken as 100ms. Hence, within this sampling time, the vehicle practically deviates by an angle ($\delta|\theta|$). From this, the current position and orientation of the robot have been calculated.

Table 1 Rule base for the first layer of fuzzy logic controller

	Distance			
Angle	**VN**	**N**	**M**	**F**
L	AR	A	A	A
AL	R	AR	AR	A
A	L	AL	AL	A
AR	L	AL	AL	A
R	AL	A	A	A

The translational velocity (m/sec) and rotational velocity (rad/sec) are fuzzified as shown in Figs. 12 and 13, respectively.

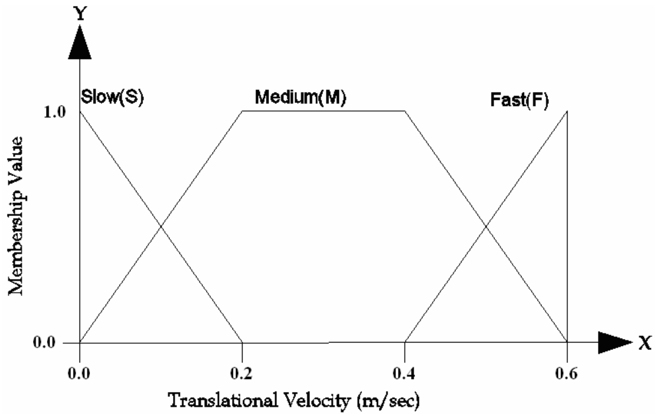


Fig. 12 Membership function distributions of translational velocity

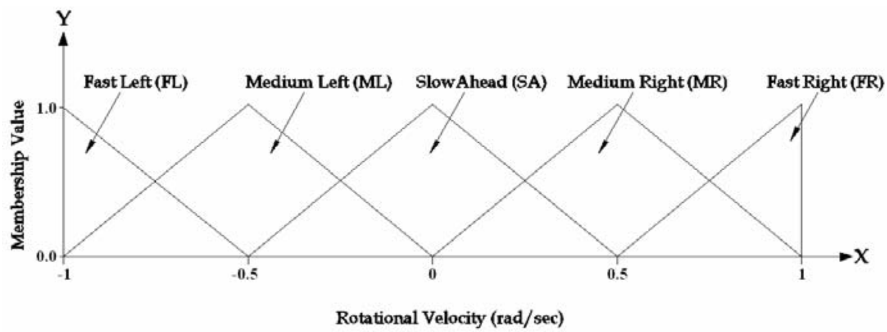


Fig. 13 Membership function distributions of rotational velocity

Fuzzy rule base for the second layer is displayed in Table 2.

Table 2 Rule base for the second layer of fuzzy logic controller

Deviation Angle	**L**	**AL**	**A**	**AR**	**R**
Translational Velocity	S	M	F	M	S
Rotational Velocity	FL	ML	SA	MR	FR

6 Vision System

In order for the vehicle to climb the stairs automatically, one critical parameter must be estimated robustly and continuously, that is, offset angle (α) between the stair orientation and vehicle heading direction.

Staircase edges are generally parallel to each other in 3D space, hence for simplicity, other spiral types of staircases are not considered in this work. Therefore, the projected stair edges on the images should intersect at a vanishing point provided that the edges are not front-to-parallel to the image plane [13][42]. But, in real cases, when a staircase is seen from a distance, the lines are quite parallel to each other, because the vanishing point is far from the image. It is, therefore, logical to search for concurrent lines when looking for a structure that originally consists of parallel lines. Hence, the automatic detection of stair-cases can be decomposed as shown in Fig. 14.

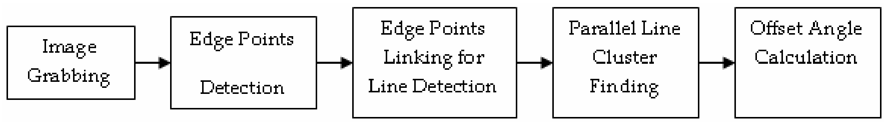


Fig. 14 Block diagram for stair angle detection

A PC-based image processing system has been developed for detection of stair orientation angle. A CCD camera has been employed for grabbing the image. Compressed JPEG image has been transferred to PC through faster ethernet interface. This has been used to build powerful PC-based imaging system for stair detection purposes.

Feature detection from the camera images itself requires high processing time. While selecting a particular feature detection technique for the online application in a navigational system, particularly for autonomous stair navigation problem, the issues that have been considered are: (1) less processing time and (2) edges should be thinner enough resulting less number of edge points, which in fact affect the time for edge linking. From the literature survey, it has been found out that there are different edge detection techniques like (i) Canny edge detector [35]), (ii) traditional Sobel edge detector (which is supplemented with hysteresis thresholding as used in the Canny), (iii) Nalwa–Binford edge detector [37] (which was chosen to represent the surface fitting approach to the edge detection), and (iv) Sarkar–Boyer edge detector [38] (which is considered for representing the current state of the art in the *zero crossing* approach). According to the review of [36], the Sobel is the baseline historical standard and the Canny is a modern standard, in the sense that papers describing new edge detectors often compare the results to those of the Canny.

In this work, edge points are detected by Canny Edge detector algorithm. For the implementation of the above edge detection technique, one crucial point is the fixing of required input parameters which also depends on the image resolution and the size of the object of interest. For the implementation of Canny Edge detector, the required three parameters are: (i) low hysteresis threshold, (ii) high hysteresis threshold, and (iii) ¾ of a Gaussian, which controls the amount of smoothing.

Edge points linking procedure is the immediate step after edge points detection for extracting the meaningful edges. Among the so far adopted methods, linear regression [13] is used to recognize all possible line segments. This requires evaluation of the relative positions of edge data points to determine if they belong to a line segment. This method is appropriate for those cases where numbers of edge data points are very few and also the data points are assumed to be gathered in a geometrically sequential order. In this method, a newly acquired data point is only tested against the present line. If it does not fit into the present line, a new line is started. But, for the case of large edge data processing obtained from a camera image, detection of lines by this method is really cumbersome. For n number of edge points extracted from the image, it involves finding of $n(n-1)/2 \approx n^2$ lines and then comparing with $n(n(n-1)/2) \approx n^3$ lines. This approach is computationally prohibitive due to its large processing time.

Another approach is based on the theory of linear contrasts [43]. By incorporating the F-statistic and the shape test, this algorithm detects and locates line features in a correlated noise environment. There is another approach which is based on representing edge segments in the form of a graph and searching the graph for low-cost paths that corresponds to significant edges. This approach may be a rugged one, but is considerably complicated and takes large computational time.

By using Hough transformation technique [39-41], the problem of finding n^2 lines and then comparing with n^3 lines can be solved within the minimum stipulated time period. In this work, Hough Transform line fitting algorithm has been adopted for stair edge line detection from the images. According to Hough Transformation**,** a line L can be represented by the equation as shown in Fig. 15., $p = x\cos\beta + y\sin\beta$, where p is the distance from the origin of the (x,y) coordinate system and β is the angle rotated. For any point (x,y) of the image (size = $row \times col$), the quantized Hough space is (p, β) as shown in Fig. 15, such that $p = \left(x - \frac{col}{2}\right)\cos\beta + \left(\frac{row}{2} - y\right)\sin\beta$

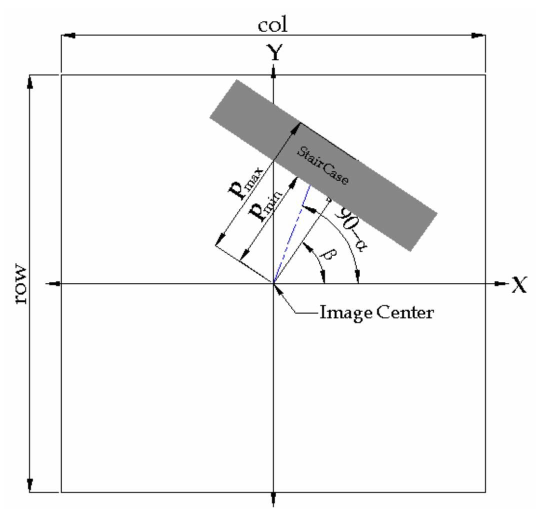


Fig. 15 A schematic showing definition of p, β

The Hough algorithm is used to locate the unique (p, β) coordinate of a straight line. After obtaining the straight lines using Hough Transform, the clusters of at least five parallel lines are searched which are close together, and return (β_m, p_{min}, p_{max}) for the kerb region perpendicular to the slope tan(β_m) extending from p_{min} to p_{max} as shown in Fig. 15. This provides information relevant to the estimated position of the stair region. Now, the offset angle (α) between the stair orientation and vehicle heading direction is calculated with respect to the image center.

7 Results and Discussion

Hardware of the stair climbing vehicle has been developed and mechanisms for navigation on both plane surface as well as stair, and that for pay-load orientation platform have been tested through a suitable graphical user interface (GUI), which is a command driven interface between the user and the vehicle. GUI also displays the sensory feedback like camera image, distance data etc., which in fact assists the user to predict the environment and also updates internal status of the vehicle. Here, an attempt has been taken to compare the results of the theoretical model with the performance of the actual system. The performance testing of the FLC-based system and vision-based automatic detection of the stairs have also been carried out with the instruction necessary for automatic navigation.

7.1 Motion Simulation and Experimentation on Stair

The motion on plane surface based on the skid-steering has been tested and compared with simulation based on theoretical model. Fig. 16 shows the actual trajectory

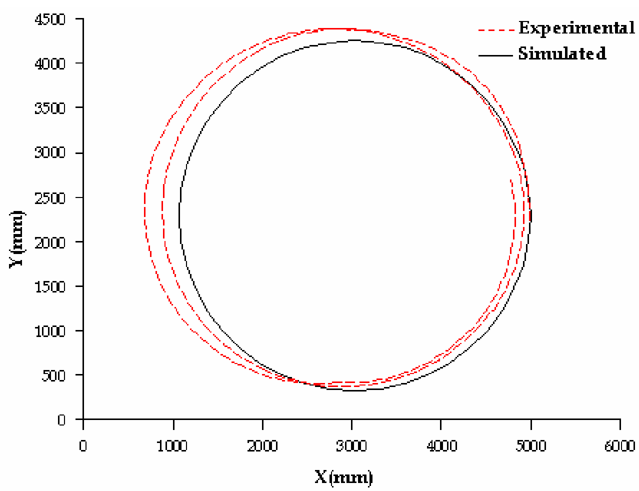


Fig. 16 Experimental and simulated trajectory for a circular motion with ω_1 = 54 rpm and ω_5 = 108 rpm

of the center of mass of the robot while operating under constant angular velocities of $\omega_1=\omega_2=\omega_3=\omega_4=54$ rpm and $\omega_5=\omega_6=\omega_7=\omega_8=108$ rpm on a concrete surface. The curve indicates that the vehicle almost follows the nature of the simulated trajectory but with certain offset. This is mainly due to the mechanical misalignment and geometrical variation of the different components of the vehicle.

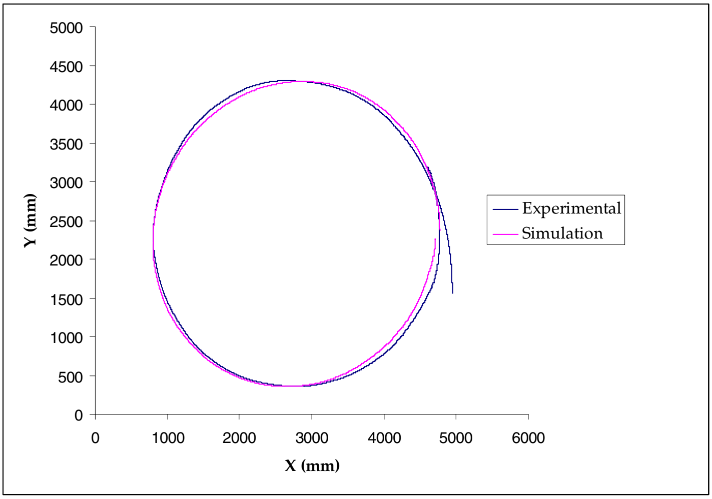


Fig. 17 Simulation and experimental tracking results

It is also necessary to test the performance of the vehicle while it is tracking a particular trajectory. In this study, a circular trajectory is considered having the trajectory parameters as $X = r\cos(\theta)$ and $Y = r\sin(\theta)$, where r=2 m. Thus, the ICR of the robot is located at a distance of X= 2.778 m, Y= 2.349 m. The slope of the wheel/ground friction coefficient μ-λ curve is approximated and assumed to be a constant at K = 5 during the entire maneuver. The other parameters in the μ-λ curve during modeling are μ_s= 0.6, μ_m = 0.75, s_{m1}=0.15, s_{m2} = 0.25. For testing of the vehicle, it starts from the location at (4773, 2367) with horizontal velocity $\dot{x} = 0.2$ m/s and yaw angle φ(t=0) = 0. The initial wheel velocities are $\omega_5 = \omega_6 = \omega_7 = \omega_8$ (at t=0)=50 rpm. Figure 17 shows the comparison curve of robot trajectory during simulation and experimentation. The experimental result also shows that the vehicle takes the desired trajectory quickly from its initial starting position.

By monitoring the current of the drive motor amplifiers, the power and torque used to drive each wheel has been estimated to ascertain the performance of the vehicle. It is necessary to compare the power and torque requirement both

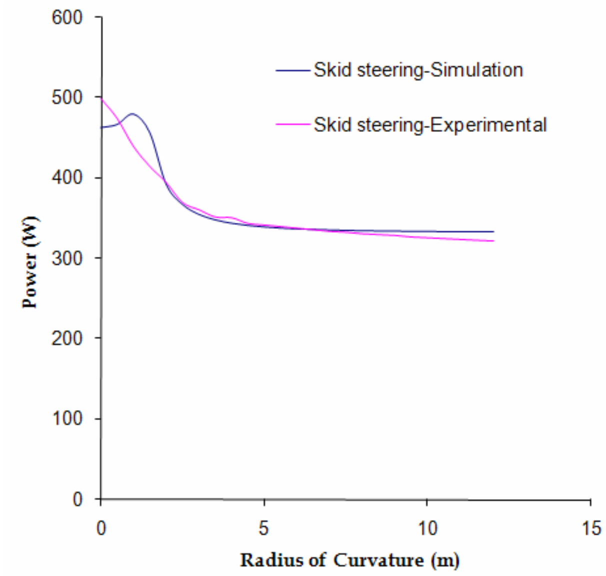
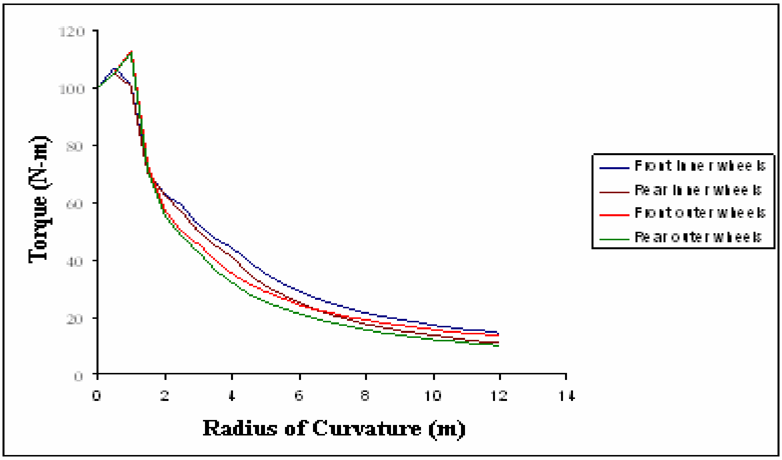


Fig. 18 Simulation and experimental power requirement at different radii of curvature

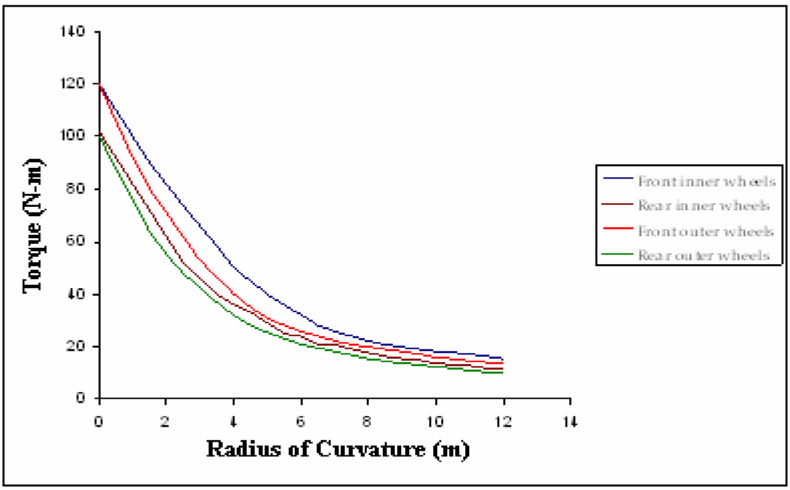
theoretically and experimentally. The torque constant for the drive motors is given as 0.5 Nm/Amp. Using the gear reduction of 63:1, the torque of the wheel has been evaluated.

Figure 18 shows the simulation and experimental results of the requirement of power with respect to radius of curvature for skid steering. As the radius of curvature decreases from straight driving to a point turning, power requirement for all the cases increases due to the increase of sideslip angle. It has been found from theoretical study that the power requirement has been increased gradually to its maximum, when the radius of curvature increases form zero to half of the width of the vehicle and then the curve gradually decreases up to a certain radius of curvature, after which it is kept almost constant. The minimum radius of curvature requirement for skid steering is zero due to the physical configuration.

Figure 19 depicts the comparison of the torque values between simulation and experimentation. Quantification of wheel torques during experimentation at different radii of curvature (0, 1, 2, 4, 6, 8, 10 and 12 m) has been carried out. For straight driving, the mean torque of four wheels has come about 98 Nm. This matches the trend of the increased radius of curvature converging to straight line motion. From the theoretical study, it has been found that power and torque requirement are the highest, when the radius of curvature is kept equal to half of the width of the vehicle rather than point steering, because of high skidding angle at the outer wheels compared to the inner wheels.



(a)



(b)

Fig. 19 Torque versus Radius of Curvature - (a) Simulation results, (b) Experimental results

7.2 Motion Simulation and Experimentation on Stair

Stair climbing operation is carried out by the rotation of wheel cluster. For climbing a single stair-case height, the required rotational angle of each wheel cluster is 120°. But, if the stair-case width is not long enough to hold two wheels simultaneously, the wheel cluster assembly cannot stay on a single stair case.

As shown in Fig. 20 (i), wheel cluster rotates from position ‘a’ to ‘b’ where two wheels of the wheel cluster are on two different stair-cases. Then, the rest of the motion as shown by ‘b’, ‘c’ and ‘d’ repeat during the course of motion. Therefore,

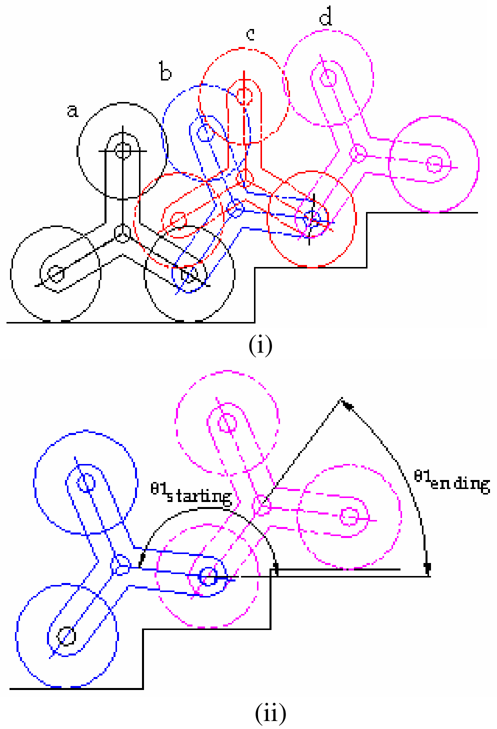


Fig. 20 Stair climbing motion of a single cluster

it is obvious that from the position ‘b’ to position ‘d’, the variation of required torque characteristic is also repetitive in nature. An attempt has been made to quantify the required torque for climbing a single stair-case, as shown in Fig. 20 (ii).

Based on the design, θ_1 will vary from 160° (starting) to 40° (ending). From the theories and the experimental data as plotted in Fig. 21, it reveals that the

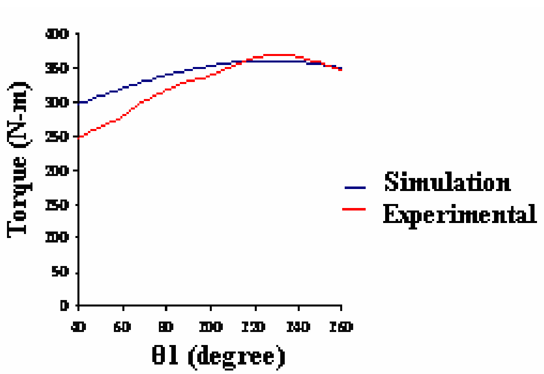


Fig. 21 Torque Vs. θ_1 for 30° stair angle

requirement of torque varies with rotational angle of the wheel clusters. As each wheel is a combination of three wheels of similar configuration, the nature of the curve repeats every 120° rotation of wheel assembly.

7.3 Stability Margin

Stability analysis is an important criterion for a dynamic system and particularly for stair climbing vehicles as they move against gravity. The stability margin has been simulated for stair climbing operation as shown in Fig. 22, which shows the quasi-static stability margin of the vehicle during stair navigation. Two types of stability margin have been depicted: i) payload orientation platform in active condition and ii) with inactive payload orientation platform. The comparison indicates that the stability margin improves with adjustable payload platform.

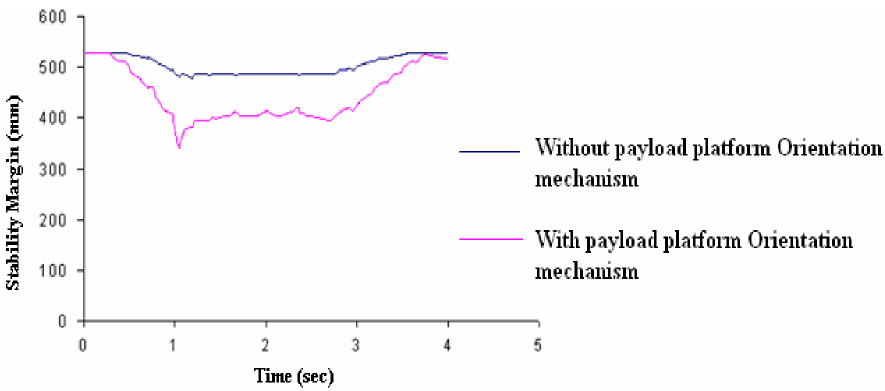


Fig. 22 Simulated quasi-static stability margin

7.4 Simulation Results of Fuzzy Logic Controller

In the beginning, Fuzzy Logic Controller (FLC) developed for the purpose of obstacle avoidance has been tested as an individual module by computer simulation, which is an iterative process. The fuzzy partitions of the input and output variables and the rule base have been continuously modified and tested through simulations for finding out a suitable motion trajectory. This process is continued until a satisfactory FLC decision has been achieved. The trajectory followed by it is not found to be optimum in any sense and this is also not an objective in this work. The simulation results are shown in Fig. 23 for various cases. Mainly two types of deficiencies have been observed in the controller. One of the deficiencies is related to left biasing or right biasing. The problem of bias arises due to the following reasons:

- When any obstacle is detected at almost straight ahead, after fuzzification, the effect of the ahead fuzzy set has become much predominant than that of ahead left or ahead right. Hence, among the rules that are fired in this situation, the rule related to the ahead contributes much predominantly.
- For example, the obstacle at 10° left of ahead would have membership value of 0.7 for ahead and 0.3 for ahead left. As a result, the decision of FLC should be influenced by the ahead part and output of FLC should be ahead left. This indicates that the vehicle should deviate towards left direction although the obstacle was located at 10° left side.

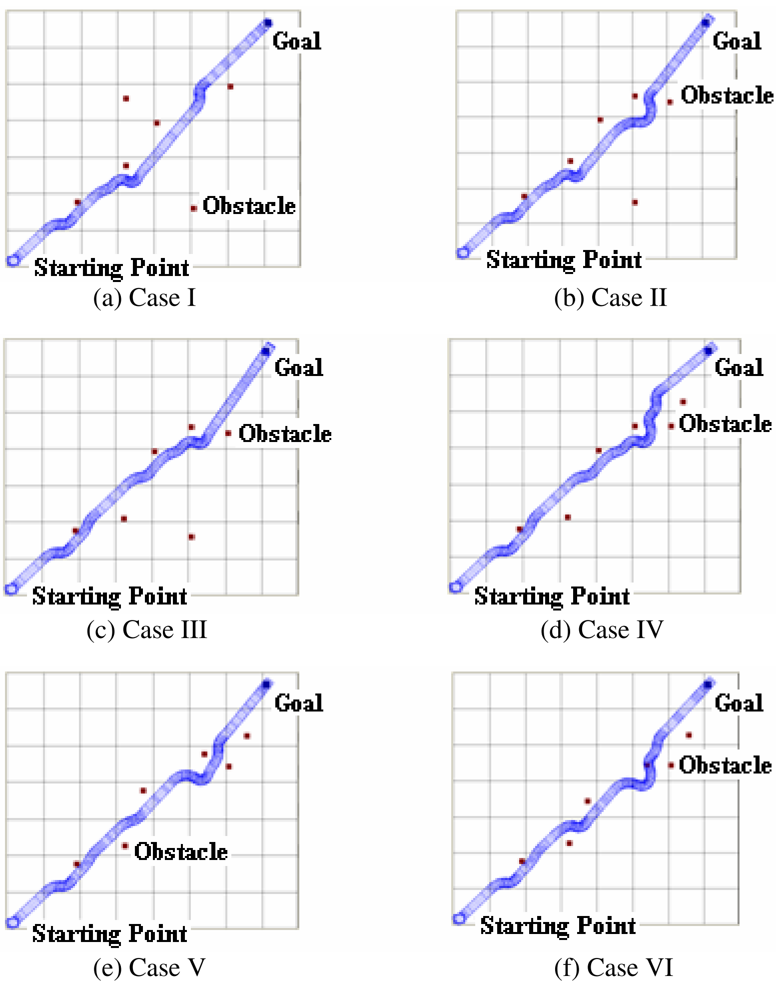


Fig. 23 Results of computer simulations for various cases

To get rid of these problems, there are two alternatives –

- During fuzzification, the universe of discourse (in this case, angle) should be fuzzified with unequal and asymmetrical divisions. The middle part, say from –5° to +5° is to be fine grained; whereas the two sides may be divided by courser divisions.
- The other alternative is to modify the rule base which could take care of this problem. In the developed FLC, the bias problem has been tackled using the second alternative.

Table 3 Rules for AHEAD and LEFT

Sensor LEFT	Sensor AHEAD			
	VN	**N**	**M**	**F**
VN	-D2	-D2	-D2	-D1
N	-D2	-D2	-D2	-D1
M	-D2	-D2	-D2	-D2
F	-D2	-D2	-D2	-D2

Table 4 Rules for AHEAD and RIGHT

Sensor RIGHT	Sensor AHEAD			
	VN	**N**	**M**	**F**
VN	D2	D2	D2	D3
N	D2	D2	D2	D3
M	D2	D2	D2	D2
F	D2	D2	D2	D2

Table 5 Rules for RIGHT and LEFT

Sensor RIGHT	Sensor LEFT			
	VN	**N**	**M**	**F**
VN	-D1	-D1	D3	D3
N	-D1	-D1	D3	D3
M	D1	D1	D3	D3
F	D1	D1	D3	D3

Another major deficiency observed through simulation is that the robot is unable to avoid the obstacles if these are closely placed. This is because of the fact that the rules of the FLC take ultimate decision on the basis of only one obstacle, which is the most critical at that instant. If obstacles are very closely spaced, more than one obstacles may be found to be critical in practice, though mathematically only one will be the nearest. In this case, the robot will avoid the most critical obstacle, but at the same time there is every possibility that it will hit the next critical obstacle, as shown in Fig. 23(f). This problem has been solved by designing new heuristic rules, which consider two or more critical obstacles for making the right decision as follows:

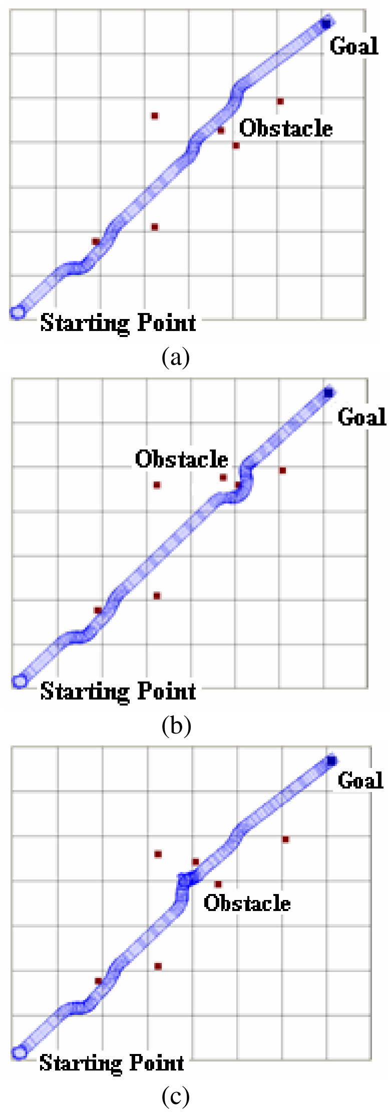


Fig. 24 Simulation results when (a) LEFT and AHEAD sensors' data are critical, (b) AHEAD and RIGHT sensors' data are critical, and (c) LEFT and RIGHT sensors' data are critical

Since there are three sensors, namely LEFT, AHEAD and RIGHT which are responsible for detecting the obstacle in the forward or backward direction, following cases may arise for the detection of obstacle - a) AHEAD and LEFT or b) AHEAD and RIGHT or c) LEFT and RIGHT or d)LEFT, AHEAD and RIGHT. Tables 3, 4 and 5 show the rules for the first three cases, where D1, D2 and D3 denote the output of fuzzy deviation angle considering the data of sensor LEFT, sensor AHEAD and sensor RIGHT, respectively. When the obstacles sensed by the three sensors are found to be critical, then the deviation of the vehicle will be - D3, provided it updates its data regarding presence / absence of obstacle during the course of turning. Figs. 24 (a), (b) and (c) show the simulation considering the most critical three cases.

7.5 Results for Staircase Detection Using Vision Sensor

Image processing techniques have already been explained in section 6 for detection of stairs using Canny Edge Detector, Hough Transform and Clustering techniques. The following series of images (refer to Table 6) reveal the above techniques applied to images for detection of stairs and its orientation angle.

Table 6 Sequences of processed images for the detection of stairs

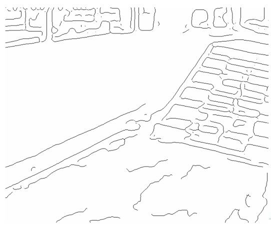
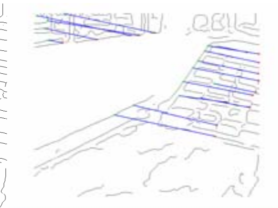
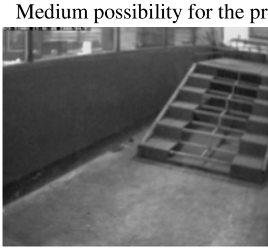
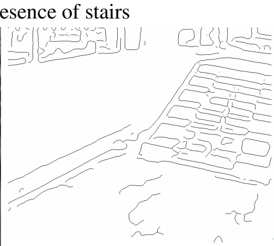
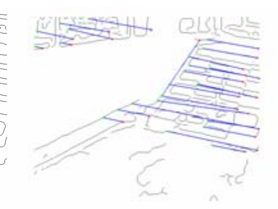
Sl. No	Original Image	Image after edge detection	Image after HT and Clustering
1			
Remarks	No of parallel lines=11 at orientation angle ≈ 79°, gap is uniform and no clear indication of two side edges Medium possibility for the presence of stairs		
2			
Remarks	No of parallel lines=18 at orientation angle ≈ 78°, gap is uniform and clear indication of one side edges High possibility for the presence of stairs		

Table 6 (*continued*)

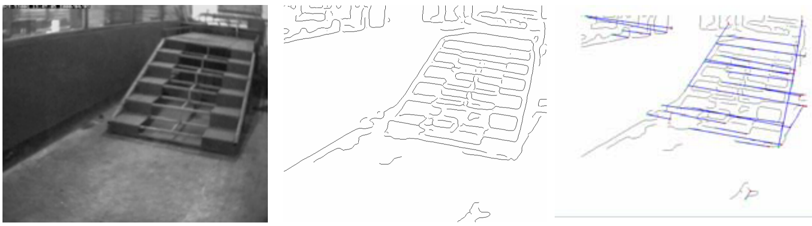
3	
Remarks	No of parallel lines=17 at orientation angle ≈ 78°, gap is uniform and side edges are not totally cleared High possibility for the presence of stairs
4	
Remarks	No of parallel lines=18 at orientation angle ≈ 79°, gap is uniform and clear indication of two side edges Highest possibility for the presence of stairs

Figure 25 shows different steps for stair climbing. Figure 26 indicates the experimental path followed by the vehicle during navigation on plane surface and climbing on stair-case. It incorporates different sensory devices for stair detection and obstacle avoidance. It also reveals that for stair alignment, the vehicle will adjust its posture by slightly moving in backward direction and then move in forward direction.

Fig. 25 Navigation on stair-case

Fig. 25 (*continued*)

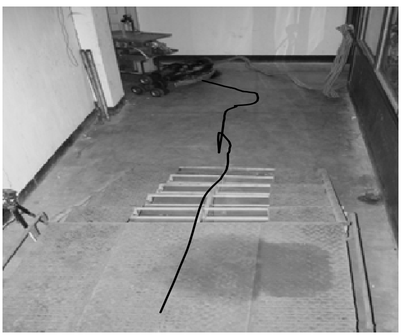


Fig. 26 Navigation on plane surface and on stairs

8 Conclusion

The present chapter deals with the issues of how a sensor integrated arm-wheel design with suitable navigational algorithm can transform a wheel type vehicle for its suitability to overcome barriers like stairs. Hence, the design and development of the robotic vehicle with stair climbing capability, thorough analysis of the vehicle, sonar-based obstacle and vision-based stair detection and their implementation in the obstacle avoidance algorithm are claimed as important contributions in compliance with the autonomous stair navigation.

Acknowledgments. The authors express their gratitude to the Department of Science and Technology, Govt. of India, New Delhi, for providing the necessary financial assistance for development of the vehicle.

References

[1] Hirose, S., Sensu, T., Aoki, S.: The TAQT Carrier: A Pratical Terrain-Adaptive Quadru-Track Carrier Robot. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Tokyo, pp. 2068–2073 (1992)
[2] Uchida, Y., Furuichi, K., Hirose, S.: Evaluation of Wheel Performance on Rough Terrain and Development of HS Wheel. Journal of Robotics and Mechatronics 12(5), 593–602 (2000)
[3] Häussler, C., Salig, M., Ullrich, R.: Stair Climbing. US Patent 6,328,120 B1 (2001)
[4] Cox, K.: Battery Powered Stair-Climbing Wheelchair. US Patent 6,484,829 B1, November 26 (2002)
[5] RH-2001A (2002), `http://www.runsoft.com.cn/fwzx/fwzx4wtjd.htm` (May 20, 2002)
[6] Michaelsen, A., Popp, K.: Kinematics and Control of a Stair Climbing Robot. In: Proc. of Int. Conf. on Motion and Vibration Control, St Louis (2004)
[7] Morales, R., Felie, V., Gonzalez, A., Pintado, P.: Kinematic Model of a New Staircase Climbing Wheelchair and its Experimental Validation. Int. Journal of Robotics Research 25(9), 825–841 (2006)
[8] Takano, M., Odawara, G.: Development of new type of Mobile Robot TO-ROVER. In: Proc. of Int. Symposium on Industrial Robots, vol. 2(20), pp. 81–89 (1983)
[9] Ray, R., Barai, N., Bhaumik, S.: Design and Development of an Intelligent Mobile Vehicle for Stair Navigation. In: Proc. of All India Manufacturing Technology Design and Research Conference (AIMTDR), BIT, Mesra, India (2002)
[10] Dalvand, M.M., Moghadam, M.M.: Stair Climber Smart Mobile Robot (MSRox). Autonomous Robots, Research 20, 3–14 (2006)
[11] Helmick, D.M., Roumeliotis, S.I., McHenry, M.C., Matthies, L.: Multi-Sensor, High Speed Autonomous Stair Climbing. In: Proc. of Int. Conf. on Robotics and Systems, EPFL, Lausanne, Swizerland, pp. 733–742 (2000)
[12] Hirose, S., Fukushima, E., Damoto, R., Nakamoto, H.: Design of terrain adaptive versatile crawler vehicle HELIOS-VI. In: Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp. 1540–1545 (2001)
[13] Matthies, L., Xiong, Y., Hogg, R., Zhu, D., Rankin, A., Kennedy, B.: A Portable, Autonomous, Urban Reconnaissance Robot. Robotics and Autonomous Systems 40, 163–172 (2002)
[14] Steplight, S., Egnal, G., Jung, S.-H., Walker, D.B., Taylor, C.J., Ostrowski, J.P.: A Mode Based Sensor Fusion Approach to Robotic Stair Climbing. In: Proc. of Int. Conf. on Intelligent Robots and Systems, pp. 1113–1118 (2000)
[15] Fair, M., Miller, D.P.: Automated Staircase Detection, Alignment, and Traversal. IASTED. In: Proc. of Int. Conf. on Robotics and Manufacturing, Cancun, Mexico, pp. 218–222 (2001)
[16] Quigg, R.T.: Stair Climbing Wheelchair. US Patent 6,857,490 B2, February 22 (2005)
[17] Saranli, U., Buehler, M., Koditschek, D.E.: RHex: A Simple and Highly Mobile Hexapod Robot. Int. J. Robotics Research 20(7), 616–631 (2001)
[18] Fujii, S., Inoue, K., Takubo, T., Arai, T.: Climbing Up onto Steps for Limb Mechanism Robot "ASTERISK". In: Proc. of ISARC, pp. 225–230 (2006)

[19] Yoshiaki, S., Ryujin, W., Chiaki, A., Shinichi, M., Nobuo, H., Kikuo, F.: The Intelligent ASIMO: System Overview and Integration. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2478–2483 (2002)
[20] Gutmann, J.S., Fukuchi, M., Fyjita, M.: Stair Climbing for Humanoid Robots Using Stereo Vision. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2478–2483 (2004)
[21] Fumio, K., Hirohisa, H., Kenji, K., Shuuji, K., Kiyoshi, F., Kensuke, H., Kazuhito, Y.: Locomotion Planning of Humanoid Robots to Pass Through Narrow Spaces. In: Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 604–609 (2004)
[22] Pfeiffer, F., Löffler, K., Gienger, M.: Humanoid Robots. In: Proc. of Int. Conf. on Climbing and Walking Robots, pp. 505–516 (2003)
[23] Nelson, G.M., Quinn, R.D., Bachmann, F.W.C., Ritzmann, R.E., Watson, J.T.: Design and Simulation of a Cockroach-like Hexapod Robot. In: Proc. of IEEE Int. Conf. on Robotics and Automation, Albuquerque, NM (1997)
[24] Lawn, M.J., Ishimatsu, T.: Modeling of A Stair-climbing Wheelchair Mechanism with High Single Step Capability. IEEE Transactions on Neural Systems And Rehabilitation Engineering 11(3), 323–332 (2003)
[25] Yuan, J., Hirose, S.: Zero Carrier: A Novel Eight Leg-Wheels Hybrid Stair Climbing Mobile Vehicle. Journal of Robotics and Mechatronics 17(1), 44–51 (2005)
[26] Takahashi, M., Yoneda, K., Hirose, S.: Rough Terrain Locomotion of a Leg-Wheel Hybrid Quadruped Robot. In: Proc. of IEEE Int. Conference on Robotics and Automation, Orlando, pp. 1090–1095 (2006)
[27] Anh Tuan, L., Rye, D.C., Durrant-Whyte, H.F.: Estimation of Track-soil Interactions for Autonomous Tracked Vehicles. In: Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1388–1393 (1997)
[28] Caracciolo, L., De Luca, A., Iannitti, S.: Trajectory Tracking Control of a Four-Wheel Differentially Driven Mobile Robot. In: Proc. of IEEE Int. Conf. Robotics and Automation, Detroit, MI, pp. 2632–2638 (1999)
[29] Wong, J.Y.: Theory of Ground Vehicles. Wiley, New York (1993)
[30] Weiss, K.R.: Skid-Steering. Automobile Engineer., 22–25 (1971)
[31] Yi, J., Alvarez, L., Horowitz, R.: Adaptive Emergency Braking Control with Underestimation of Friction Coefficient. IEEE Transactions on Control Syst. Technol. 10(3), 381–392 (2002)
[32] Balakrishna, R., Ghosal, A.: Modeling of Slip for Wheeled Mobile Robots. IEEE Transactions on Robotics and Automation 11(1), 126–132 (1995)
[33] Kozowski, K., Pazderski, D., Rudas, I., Tar, J.: Modeling and Control of a 4-Wheel Skid-steering Mobile Robot: From Theory to Practice. In: Budapest Tech. Polytechnical Institution's Jubilee Conf., Hungary, pp. 317–330 (2004)
[34] Shigley, J.E., Uicker, J.J.: Theory of Machines and Mechanisms. McGraw Hill, New York (1995)
[35] Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 679–714 (1986)
[36] Heath, M., Sarkar, S., Sanocki, T., Bowyery, K.: Comparison of Edge Detectors: A Methodology and Initial Study. Computer Vision and Image Understanding 69(1), 38–54 (1998)
[37] Nalwa, V.S., Binford, T.O.: On Detecting Edges. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 699–714 (1986)
[38] Sarkar, S., Boyer, K.L.: Optimal Infinite Impulse Response Zero Crossing based Edge Detectors. Computer Vision and Image Understanding 54(2), 224–243 (1991)

[39] Hough, P.V.: Method and Means for Recognizing Complex Patterns. U. S. Patent 3, 069 654, December 18 (1962)

[40] Yun, X., Latt, K., Scott Glennon, J.: Mobile Robot Localization using the Hough Transform and Neural Networks. In: Proc. of IEEE Int. Symposium on Intelligent Control, vol. 14(17), pp. 393–400 (1998)

[41] Dutta, S., Ray, R.: AMR Vision System for Perception, Job Detection and Identification in Manufacturing. In: Vision Systems: Applications, Austria, EU, ch. 27, pp. 517–538. I-Tech Publications (2007)

[42] Xiong, Y., Matthies, L.: Vision-guided Autonomous Stair Climbing. In: IEEE Int. Conf. on Robotics and Automation, San Francisco, CA, vol. 2, pp. 1842–1847 (2000)

[43] Behar, D., Cheung, J., Kurz, L.: Contrast Techniques for Line Detection in a Correlated Noise Environment. IEEE Transactions on Image Processing 6(5), 625–641 (1997)

[44] Dixon, W.E., Dawson, D.M., Zergeroglu, E., Behal, A.: Nonlinear Control of Wheeled Mobile Robots. Springer, Heidelberg (2001)

[45] Dutta, S., Ray, R., Banerji, D.: Development of Autonomous Mobile Robot with Manipulator for Manufacturing Environment. Int. Journal of Manufacturing Technology 38, 536–542 (2007)

[46] Ray, R., Bhaumik, S., Shome, S.N.: A Skid-Steer Robotic Vehicle with Stair Climbing Ability. Int. Journal of Advanced Robotics System, Communicated (2009)

6

Ensemble Learning for Multi-source Information Fusion

Jörg Beyer, Kai Heesche, Werner Hauptmann, Clemens Otte, and Rudolf Kruse

Abstract. In this chapter, we propose a new ensemble learning method. The main objective of this approach is to jointly use data-driven and knowledge-based submodels, like mathematical equations or rules, in the modeling process. The integration of knowledge-based submodels is of particular interest, since they are able to provide with information not contained in the data. On the other hand, data-driven models can complement the knowledge-based models with respect to input space coverage. For the task of appropriately integrating the different models, a method for partitioning the input space for the given models is introduced. Using that kind of ensembles, the advantages of both models are combined, i.e., robustness and physical transparency of the knowledge-based models and approximation abilities of the data-driven learning. The benefits of this approach are demonstrated for a real-world application.

Jörg Beyer
Siemens AG - CT IC 4, Otto-Hahn-Ring 6, 80200 Munich, Germany
Otto-von-Guericke-University Magdeburg - School of Computer Science, Universitätsplatz 2, 39106 Magdeburg, Germany
e-mail: joerg.beyer.ext@siemens.com

Kai Heesche
Siemens AG - CT IC 4, Otto-Hahn-Ring 6, 80200 Munich, Germany

Werner Hauptmann
Siemens AG - CT IC 4, Otto-Hahn-Ring 6, 80200 Munich, Germany

Clemens Otte
Siemens AG - CT IC 4, Otto-Hahn-Ring 6, 80200 Munich, Germany

Rudolf Kruse
Otto-von-Guericke-University Magdeburg - School of Computer Science, Universitätsplatz 2, 39106 Magdeburg, Germany

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 123–141.
springerlink.com

1 Introduction

Modern technical systems are characterized by an increasing degree of sophisticated behavior. The traditional way of modeling has been by mathematical equations representing the physical behavior. However, the identification of parameters is time-consuming and expensive. Data-driven models, like artificial neural networks, can be used to approximate physical phenomena. But, the data-driven modeling approach usually suffers from the lack of physical understanding of the model parameters. The resulting model only relies on the training data and does not use any other information source available. Thus, it is desirable to combine available information in terms of knowledge-based models, i.e., models which are based on domain or process knowledge, designed without training data and to complement this information with the data-driven approach.

The integration of knowledge-based submodels has several advantages:

- enhancing the interpretability, i.e., a domain expert can easily comprehend the decisions,
- providing information not contained in the training data and
- reducing the amount of required training data.

For these reasons, an important factor for the generation of adequate models of a technical system is the use of available information in terms of knowledge-based models and the supplementation of this information by data-driven models learnt on the training data. Since a knowledge-based model represents a particular subsystem, information with respect to its validity has to be included in the overall model.

The objective of the proposed approach is to generate an ensemble that is able to integrate the available knowledge-based submodels and to complement these submodels by data-driven ones. Using that kind of ensembles the advantages of both models are combined, i.e., the robustness and physical transparency of the knowledge-based models and the approximation abilities of data-driven learning.

The use of multiple models is also motivated by the paradigm that different partial models can complement each other by appropriate compensation of weaknesses and strengths of the individual models. Much of the work on ensemble techniques has strong parallels with the research on information fusion (IF) systems. In common with the research on IF, several architectures exist and different combination schemes have been developed. Later in this chapter, we give a review of IF.

The chapter is organized as follows: In Secton 2, an introduction of IF is given and Section 3 describes different methods for creating ensembles. In Section 4, two ensemble models for combining data-driven and knowledge-based models are proposed. In Section 5, some experiments on a real-world application are outlined. Section 6 concludes the study.

2 Multiple Source Fusion

Information fusion is an important technique in different application domains, such as sensor fusion [9], identity verification [2], signal and image processing [5], and others. Due to the heterogeneity of the applications, several definitions of the term: *information fusion* exist. In Section 2.1, some definitions of IF are stated and the adopted definition of IF is given. A classification of IF is described in Section 2.2.

2.1 Definition

There exist many definitions of IF or data fusion. In [25], IF is described as a "multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from multiple sources." This is a general definition that suggests the combination of data or information without specifying its objective. Wald considers data fusion as formal framework that formulates means and tools for the combination of data from different sources [23]. In this definition, the focus lies on the framework used to fuse data. Wald also states that data fusion "aims at obtaining information of greater quality". The term *quality* means that the fused information is somehow more appropriate to the application than the original information. The most general definition comprising any type of source, knowledge, and resource used to fuse different pieces of information is given by Dasarathy [7], which states that IF "encompasses the theory, techniques and tools created and applied to exploit the synergy in the information acquired from multiple sources (sensors, databases, information gathered by humans beings, etc.) in such a way that the resulting decision or action is in some sense better (qualitatively or quantitatively, in terms of accuracy, robustness, etc.) than would be possible, if any of these sources were used individually without such synergy exploitation."

Fusion implies the combination of information from more than one source. There are different reasons for fusion of multiple sources:

- The combined solution is able to attain more accurate, transparent, and robust results, since the different information sources can complement each other with respect to their strengths and weaknesses.
- A model that depends on a single source is not robust in the sense that if the single source is erroneous, the whole model is affected. Models based on fused information sources are more robust, since other sources are able to compensate erroneous information.

Here, we chose to use IF as the process of merging and integrating heterogeneous information components from multiple sources, for instance, in the form of sensors, human experts, symbolic knowledge, or physical process models.

2.2 Classification of Information Fusion

In this study, complementary and cooperative information sources are distinguished. They are discriminated with respect to the relationships among the information sources. In complementary fusion, each source provides with information from a different region of the input space, i.e., their responsibilities do not overlap. Locally, these sources provide with a high performance. However, outside their regions, the results are not valid. Cooperative fusion means that the information is shared among several information sources in the same region of the input space and has to be fused for a more complete modeling of the underlying process.

3 Ensemble Models

There exist many approaches, which address the issue of learning and combining local models. The resulting model, referred to as ensemble, is generally more accurate than any of the submodels generating the ensemble. Both empirical [14] and theoretical [13], [17] research has demonstrated that in a good ensemble, the submodels are accurate on different parts of the input space, so that they complement one another. Figure 1 shows a common ensemble model.

The algorithms for learning local models can be discriminated with respect to several aspects: the architecture (parallel or sequential learning of the submodels), the way they divide the training data into subsets, or how they fuse the outputs of the local models.

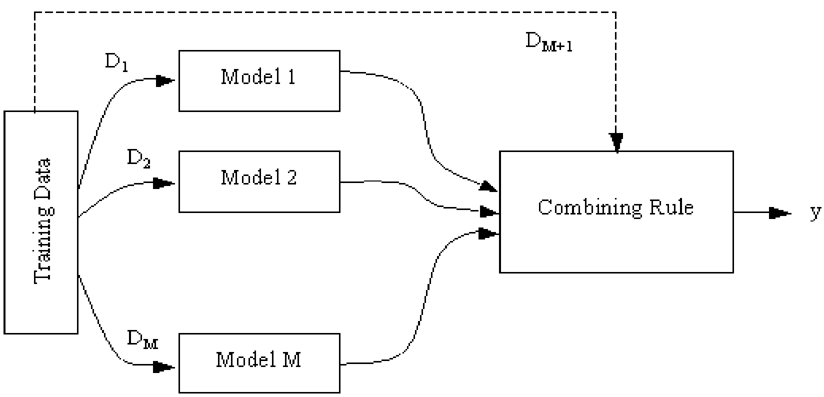


Fig. 1 A common ensemble model. The ensemble members are trained on possibly different data sets D_j. The dashed line indicates that the Combining Rule can be trainable dependent on the combining method

3.1 Stacked Generalization

Stacked generalization (or stacking) creates an ensemble of submodels, whose outputs are used as inputs to a second level combiner to learn the mapping between the ensemble member outputs and the target values [26].

The basic idea is that the outputs of the ensemble members have information that can be used to construct good combinations of the members and a procedure is sought for combining them. In the first step, a set of submodels is trained (with possibly different training data sets, submodel parameters, etc.). The outputs of these submodels and their corresponding true target values are then used as input/output training pairs for the second level combiner.

3.2 Boosting

Boosting is a method for improving the accuracy of any given learning algorithm [19]. The submodels in a boosting ensemble are trained sequentially on training data that has been filtered by the previously trained submodels in the ensemble. The boosting procedure is as follows: the first submodel is trained with a random subset of the training data. The training data set for the second submodel is chosen as the most *informative* given the first submodel. In case of classification, the submodel is trained on training data, the half of which is misclassified by the first submodel, and the other half is correctly classified. The third submodel is trained with input patterns, on which the first two submodels disagree. The submodels are combined using either averaging or a voting scheme.

A popular variation of the original boosting algorithm is AdaBoost (*adaptive boosting*) [11]. In AdaBoost, submodels can be added until some desired training error has been achieved. For that purpose, each training pattern receives a weight that determines its probability of being selected for the training of a new submodel. If a training pattern is misclassified, its probability of being selected in a subsequent submodel is increased. In this way, training data of consecutive submodels are focused on hard to separate patterns. This process can be repeated to form an ensemble, whose joint decision has arbitrarily high accuracy on the training set.

3.3 Mixture-of-Experts

The mixture-of-experts (ME) model consists of a set of models, also called experts, that perform a local function approximation [15]. ME models can be described as input-dependent mixture models, which solve problems by the divide-and-conquer strategy, i.e., they learn to decompose complex problems into simpler, easier to solve subproblems. This decomposition is learned by a gate function by partitioning the input space and assigning submodels to these regions. The output y of the ME model for an input vector $\mathbf{x}$ is computed as the combination of the weighted outputs $y(\mathbf{x}, \boldsymbol{\theta}_j)$ of the M submodels

$$y = \sum_{j=1}^{M} \pi_j y(\mathbf{x}, \boldsymbol{\theta}_j) \,, \tag{1}$$

where $\boldsymbol{\theta}_j$ are the parameters of model j, π_j is the j-th output of the gate model and is constrained to $\sum_{j=1}^{M} \pi_j = 1$.

There exist several variations of the ME model, which differ in the kind of training algorithm [16], [24], [4] and gate function [27].

3.4 Piecewise Linear Regression Models

Piecewise linear models assume a different linear behavior of the true function in different regions of the input space. The model described in [10] assumes that the input space can be divided into disjoint regions characterized by different (linear) behavior of the function to be approximated. The model learns the local linear models by an appropriate clustering of the input space.

A switching regression model assumes that the target values are generated by a number of distinct processes [21]. Quandt developed a method for estimating the switching point, i.e., the point where the processes switch, by searching through all possible switch points and finding the maximum of an appropriate likelihood function [18].

4 Fusion of Locally Valid Heterogeneous Models

The fusion of locally valid heterogeneous models is a crucial process during the training and affects the reliability and performance of the results of the integrated model. To assign the available knowledge-based models to the regions of input space, for which they are defined, the fusion rule has to take into account their validity ranges. For this purpose, a gate function, similar to the ME approach, is used.

In Section 4.1, validity functions defined for knowledge-based models, are described. In Section 4.2, we propose an adaptation of the ME model, called heterogeneous mixture-of-experts (HME). The HME model uses the validity functions during the partitioning process to assign knowledge-based models to the correct regions of the input space. In Section 4.2.2, we use a clustering algorithm as gate function that considers the data density and predictive performance of the local models for separation of the input space.

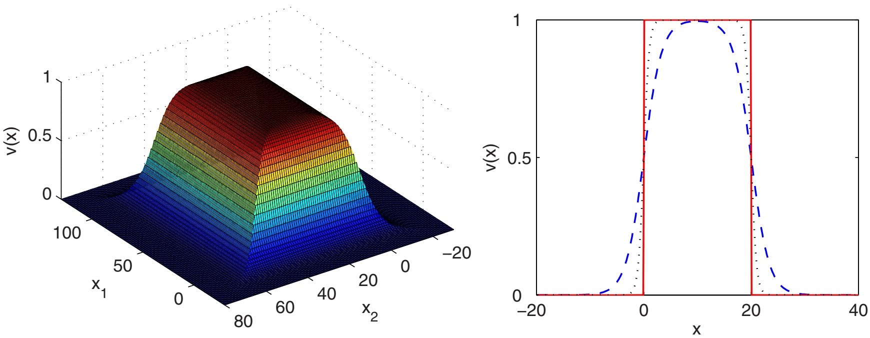


Fig. 2 Left: This figure shows an example of a validity range with the following parameter setting: $l_1 = 0$ and $u_1 = 100$ in one dimension and $l_2 = 0$ and $u_2 = 50$ in the second dimension and a slope $s = 8$. Right: The slope of the borders depends on s for $s = 1$ (*dashed line*), $s = 4$ (*dotted line*), and $s = 100$ (*solid line*)

4.1 Validity Function

The validity function of a knowledge-based model represents the region of input space the model is designed for. The validity function of model j is defined as

$$v_j\left(x^{(n)}\right) = \left(\frac{1}{1+\exp\left(s_j\left(x^{(n)}-u_j\right)\right)} - \frac{1}{1+\exp\left(s_j\left(x^{(n)}-l_j\right)\right)}\right) , \quad (2)$$

where l_j and u_j determine the lower and the upper bounds of the validity range and s_j defines the slope of the border. The effect of different values of s_j is shown in Fig. 2. The larger s_j is, the steeper is the slope of the border. In this way, the transition between the local models can be controlled. These parameters are determined by domain expert.

4.2 Heterogeneous Mixture-of-Experts

This sub-section discusses about hetero-geneous mixture-of-experts.

4.2.1 Introduction

In this section, we define heterogeneous mixture-of-experts (HME) to fuse information of multiple information sources [3]. The basic idea of this approach is to additionally include information about the specific validity ranges of the predefined knowledge-based models to be used for the partitioning of the input space. Thus, it is ensured that the predefined models are assigned to those domains of the input space they are explicitly designed for. On the other hand, data-driven models are used to close the gaps between different knowledge-based models with respect to input space coverage.

The HME model can be interpreted as a generating one, i.e., the data are generated by a set of M independent processes, which are randomly selected. Fig. 3 shows an example of an HME model. The introduction of a latent variable $Z = \left\{ z_j^{(n)} : \; j = 1, \ldots, M, \; n = 1, \ldots, N \right\}$ where $z_j^{(n)}$ is 1 if input vector $\mathbf{x}^{(n)}$ was generated by model j and 0 otherwise, and the data set $D = \left\{ \mathbf{x}^{(n)} \in \Re^k, \; t^{(n)} \in \Re, \; n = 1, \ldots, N \right\}$, allows the HME model to be trained with the Expectation-Maximization (EM) algorithm [8]. The probabilistic model can be seen in Fig. 4, which shows the belief network of the HME model. This expresses the assumption that the target $t^{(n)}$ is dependent on the input $\mathbf{x}^{(n)}$ and the multinomial random variable $z^{(n)}$. We define the conditional scalar output $t^{(n)}$ given the input vector $\mathbf{x}^{(n)}$ and the parameter of the model as:

$$P\left(t^{(n)} \middle| \mathbf{x}^{(n)}, \Theta \right) = \sum_{j=1}^{M} P\left(z_j^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_g \right) P\left(t^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_j \right), \tag{3}$$

where Θ comprises the parameter of the gate $\boldsymbol{\theta}_g$, and of the models $\boldsymbol{\theta}_j$, $j = 1, \ldots, M$. The probability $P\left(t^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_j \right)$ represents the conditional

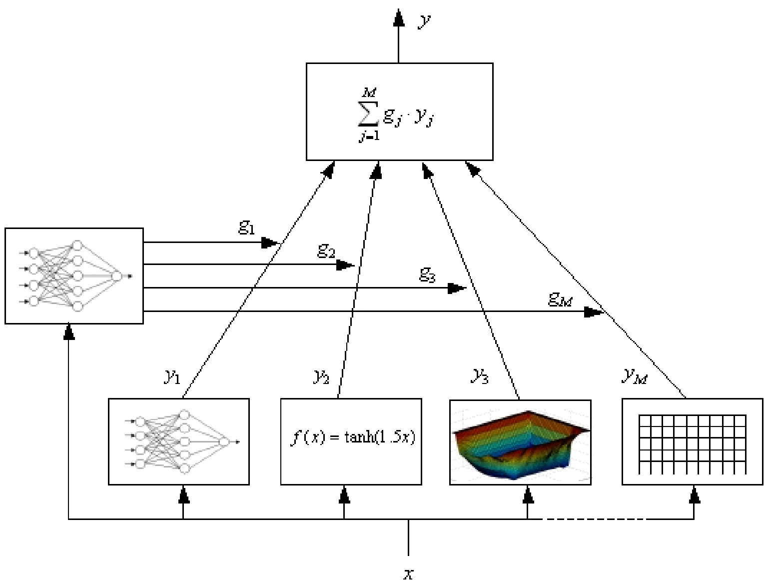


Fig. 3 Architecture of the HME model. The selection of the local models depends on **x**. The gate and the local models may use different feature subsets of the input vector

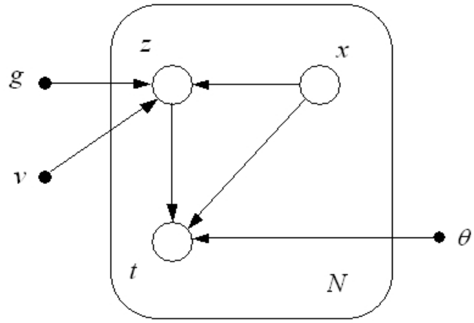


Fig. 4 Graphical representation of the HME model. Random variables will be represented by open circles and deterministic parameters will be denoted by smaller solid circles. The box contains N copies of the nodes shown inside it

densities of target $t^{(n)}$ for model j and $P\left(z_j^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_g\right)$ is the weighting coefficient of model j. The negative log likelihood function is used as error function:

$$\begin{aligned}\log L = &-\sum_{n=1}^{N}\sum_{j=1}^{M} h_j^{(n)} \log P\left(z_j^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_g\right) \\ &-\sum_{n=1}^{N}\sum_{j=1}^{M} h_j^{(n)} \log P\left(t^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_j, z_j^{(n)}\right) ,\end{aligned} \tag{4}$$

where $h_j^{(n)}$ represents the posterior probability of selecting model j for input vector $\mathbf{x}^{(n)}$. The first term of the right-hand side of equation (4) is the part of the error that the gate contributes to the overall error. It can be interpreted as the entropy of the distribution of the input vectors among the models. The second term of equation (4) presents the error component, which the individual models contribute. This is the cross-entropy among the posterior probability $h_j^{(n)}$ and the probability that model j has generated the target value $t^{(n)}$.

To guide the partitioning of the gate, the posterior probability $h_j^{(n)}$ is computed in the E step:

$$h_j^{(n)} = \frac{v_j P\left(z_j^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_g\right) P\left(t^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_j, z_j^{(n)}\right)}{\sum_{l=1}^{M} v_l P\left(z_l^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_g\right) P\left(t^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_l, z_l^{(n)}\right)} . \tag{5}$$

The role of the gate's output and the mapping v_j can be interpreted as follows: Based on the performance the gate's task is to assign a selection probability to the submodel, whereas the mapping v_j evaluates the validity of the model for an input vector. By doing so in (5), the gate will be enforced to decrease the weights of model outputs, if the input vectors are located outside their domains. The particular amount of decrease in weight is dependent on v_j.

The M step then involves finding the optimal set of parameters of the gate and the local models. It decomposes into the following separate optimization problems:

- For the parameter of the gate:

$$L_g = -\sum_{n=1}^{N}\sum_{j=1}^{M} h_j^{(n)} \log P\left(z_j^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_g\right) \tag{6}$$

- For the parameter of the data-driven local models:

$$L_j = -\sum_{n=1}^{N}\sum_{j=1}^{M} h_j^{(n)} \log P\left(t^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\theta}_j, z_j^{(n)}\right) \tag{7}$$

The parameters found on the M step are then used to begin another expectation step. These two steps are repeated until an appropriate convergence criterion, e.g., previously determined number of training iterations, is fulfilled.

4.2.2 Clustering Gating Function

In this section, we use a clustering algorithm to partition the input space and to assign local models to these partitions. It corresponds to the gate function in the HME model. To generate an appropriate partitioning of the input space, not only the data density in the input space is considered but also the performance of local models in the output space and the validity ranges of knowledge-based models.

For each local model one cluster is used. The training process comprises two main steps. First, the assignment of data to cluster prototypes is dependent on the distance between the data and the prototypes, the validity ranges and the predictive performance of the corresponding local models:

$$r_j^{(n)} = \frac{\exp\left(\left\|\mathbf{x}^{(n)} - \boldsymbol{\mu}_j\right\|^2\right) \exp\left(-1/v_j \left\|t^{(n)} - y\left(\mathbf{x}^{(n)}, \boldsymbol{\theta}_j\right)\right\|^2\right)}{\sum_{l=1}^{M} \exp\left(\left\|\mathbf{x}^{(n)} - \boldsymbol{\mu}_l\right\|^2\right) \exp\left(-1/v_l \left\|t^{(n)} - y\left(\mathbf{x}^{(n)}, \boldsymbol{\theta}_l\right)\right\|^2\right)}\ , \tag{8}$$

Second, both the cluster prototypes and the data-driven submodels are trained according to their weighted error for the training data:

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} r_j^{(n)} \mathbf{x}^{(n)}}{\sum_{n=1}^{N} r_j^{(n)}}\ , \tag{9}$$

and

$$\Delta\boldsymbol{\theta}_j = \sum_{n=1}^{N} r_j^{(n)} \left(t^{(n)} - y\left(\mathbf{x}^{(n)}, \boldsymbol{\theta}_j\right)\right) \frac{\partial y\left(\mathbf{x}^{(n)}, \boldsymbol{\theta}_j\right)}{\partial \boldsymbol{\theta}_j}\ . \tag{10}$$

At the end of the training process, the clustering algorithm has partitioned the input space among the submodels. While the knowledge-based models are only active inside their validity ranges, the data-driven models are responsible for the remaining input space.

5 Applications of Information Fusion

In Section 5.1, some applications for fusion of analytical and data-driven models are described. In Section 5.2, we describe the deployment of the HME model in a real-world application.

5.1 Combinations of Analytical and Data-Driven Models

In the field of machine learning, several approaches address the combination of analytical and data-driven models. Data-driven models can be either used to approximate nonlinear parts of the process to model parts of the process that are not observable, or as a state or disturbance estimator.

In [20], an RBF-network and an analytical model of the rolling mill process control system are combined. For unknown inputs, the RBF-network produces a correction factor close to one, thus, in these cases, the output of the overall model is determined by the analytical model alone. The advantage of this approach is that a baseline performance can be guaranteed by the analytical model.

Abonyi et al. describe an approach for using first principles models and data-driven ones, e.g., artificial neural networks (ANNs), for Generic Model Control [1]. The first principle model determines the dominant structure of a controller while data-driven models are used as a state or disturbance estimator. Van Lith et al. combine a physical framework, which builds the basis structure and complement it with fuzzy models derived from data [22].

In [12], a partial analytical model is combined with an ANN for dynamic modeling of an industrial fed-batch crystallization process. Since the target outputs of the ANN are not measured, the network outputs are fed to the analytical part of the hybrid model and the hybrid model's output are compared with available data. The network parameters are updated depending on the observed error.

The objective of our approach is to use of a combining rule that is data-generated and does not need manual adaption. The combining rule decides which submodel or submodels is/are responsible for generating the output depending on the particular input vector. Furthermore, it must be able to train data-driven submodels for parts of the input space not covered by knowledge-based models.

5.2 Modeling of Energy Flow in a Hybrid Electric Vehicle

The application addresses the simulation of electrical energy flow in the electrical system of a hybrid electric vehicle. Four distinct driving modes can be defined and represented by the available expert knowledge: a pure electric drive mode, a hybrid drive mode, a brake mode, and a drag mode. Depending on the current drive mode, electric energy is either used to drive the electric motor or produced by the generator. In pure electric drive mode and hybrid drive mode, energy is provided by the battery to drive the electric motor. In brake mode and drag mode, the electric motor is operating as a generator to regenerate the kinetic energy used for charging the battery. Domain experts designed specific models for each mode. These models represent complementary information sources because they are defined for different regions of the input space and each model provides with information for different mutually exclusive driving modes. Furthermore, the battery must maintain certain chemical limits. These limits determine the maximum charge and discharge capabilities of the battery depending on its state of charge and temperature.

The training data set in this example consists of about 10.00 input patterns, where each input pattern is 5-dimensional. The validation data set comprises approximately 100.00 input patterns. The target is one-dimensional and represents the electrical energy in kW.

Both ensemble methods: HME with a multi-layer perceptron (MLP) gate and HME with a cluster gate, are compared with a standard ME, an ensemble of MLPs, a single MLP, and a radial basis function (RBF) network. Two HME models use four local models each. Two characteristic maps and a mathematical model represent the pure electric drive mode, brake, and drag mode. However, since the hybrid drive mode is too complex to provide with a simple mathematical model, for this mode a two-layer MLP with 5 input units, 6 hidden units and one output unit were trained. Each mode uses different input features for the modeling. As gate, an MLP with 4 hidden units was applied.

The ME consists of 4 MLPs with 8 hidden units and as gate an MLP with 6 hidden units were used. The single MLP comprises 12 hidden units. The RBF network comprises 14 Gaussian basis functions. In the ensemble, 10 members are combined. All members have the same architecture, i.e., MLPs with a single hidden layer of 6 units. The output of the ensemble is computed as follows:

$$y_{ens} = \frac{1}{K} \sum_{j=1}^{K} y_j \left(\mathbf{x}^{(n)} \right) , \tag{11}$$

where $y_j \left(\mathbf{x}^{(n)} \right)$ is the output of the j member and K is the number of ensemble members.

As predictive measure, the means absolute error is used:

$$e = \frac{1}{N} \sum_{n=1}^{N} \left| t^{(n)} - f\left(\mathbf{x}^{(n)}\right) \right| . \tag{12}$$

Table 1 summarizes 10-fold-crossvalidation runs that are performed to estimate the predictive error of the regression models on previously unseen data. The HMEs with both MLP gate and clustering gate have achieved superior performance due to the incorporation of available information sources. Figs. 5 and 6 show the outputs of the gate (the activation of the submodels) of the two HME models. In most of the cases, the MLP gate selects only one submodel for each input vector. In case of clustering gate, the activations of different submodels are distributed slightly more than the MLP gate. This behavior is consistent with the knowledge of the domain expert that the submodels are defined for different slightly overlapping modes. Against the background of domain knowledge, the ME model is not able to identify the driving modes and has divided the input space in a technically non-plausible way. This is illustrated in Fig. 7. The overall output is composed of the outputs of the submodels.

Table 1 Error of the models on the hybrid vehicle data set

model	predictive error	
	training	testing
HME with MLP gate	2.10	2.14
HME with cluster gate	2.25	2.31
ME	2.78	2.88
ensemble	2.39	2.47
RBF	3.85	3.96
MLP	2.51	2.61

Table 2 shows the distribution of the responsibilities of the mode models for data of the corresponding driving mode. The values indicate that the mode models are correctly assigned to the partitions of the driving modes. These responsibilities are depicted as shaded background in Figs. 5 and 6 and confirm the results in Table 2.

Further, the incorporation of available knowledge requires fewer training data. The smaller the size of the training data set, the less robust are the results of data-driven models. In Table 3 and Fig. 8, the results for different sizes of the training data sets are shown. The results indicate that the proposed models require fewer training data compared to other regression methods to yield good predictive performance. For training data set sizes of $D/2$, $D/4$, and $D/8$ (where D indicates the original training data set), the predictive

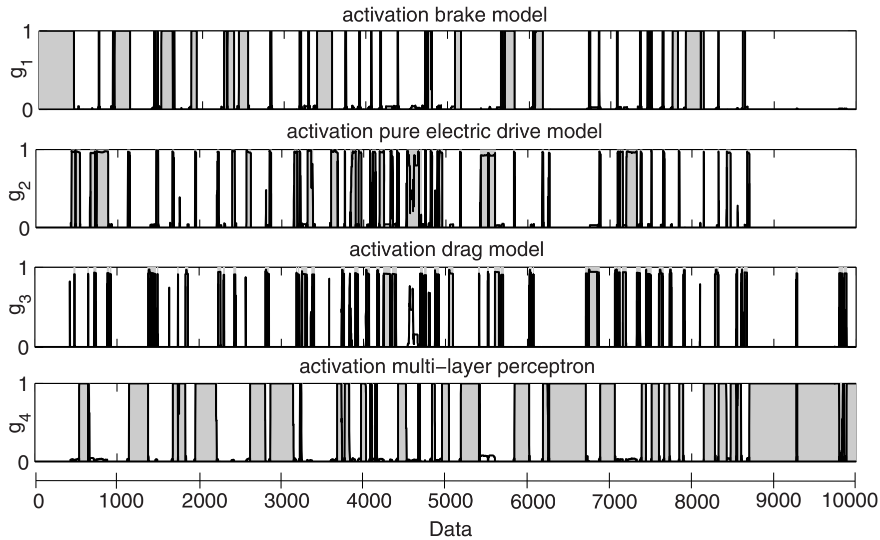


Fig. 5 The figure shows the activations of four submodels by the MLP gate of the HME model. The shaded background indicates data that correspond to the driving mode, which is represented by the mode model. The HME model has correctly identified the different driving modes and the mode models are responsible for data of the corresponding driving mode

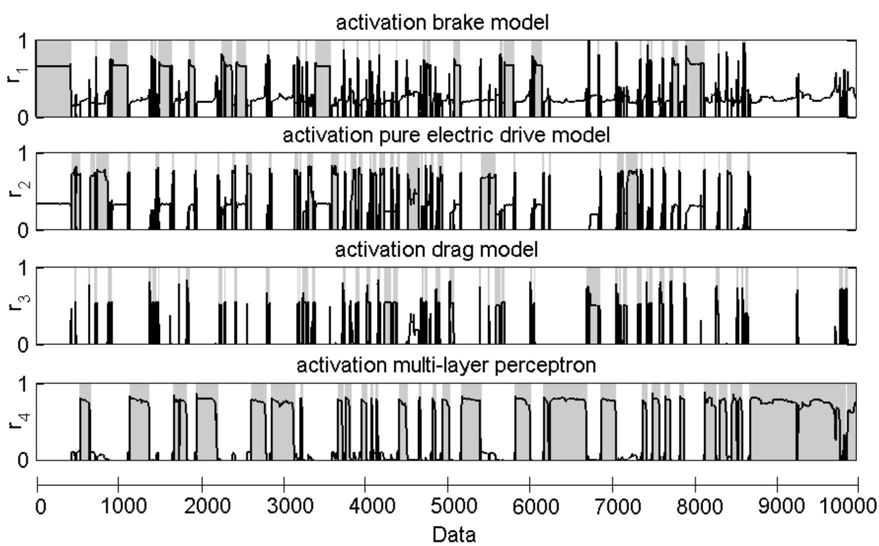


Fig. 6 The figure shows the activations of four submodels by the cluster gate of the HME model. The shaded background indicates data that correspond to the driving mode, which is represented by the mode model. The HME has correctly identified the different driving modes. The activations of the different submodels are distributed slightly more than the MLP gate

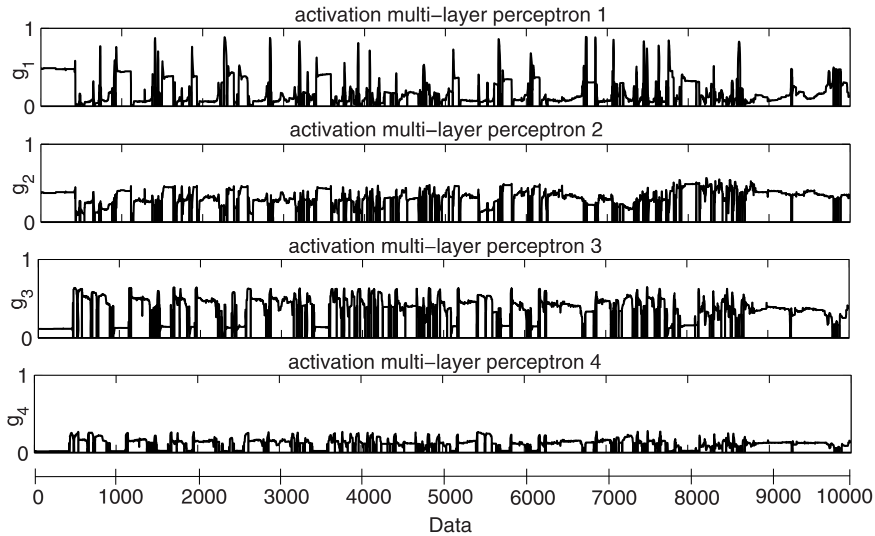


Fig. 7 The figure shows the activations of four data-driven submodels by the gate of the ME model. The ME model is not able to identify the driving modes and has divided the input space in a technically non-plausible way

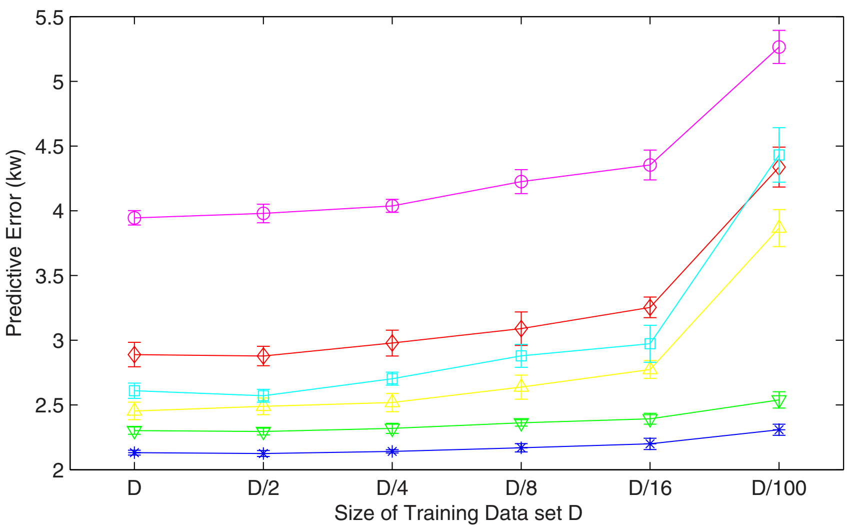


Fig. 8 The plot shows the predictive error of the models for different sizes of the training data set. The HME models with both MLP gate (*asterisk*) and clustering gate (*downward-pointing triangle*) have a slightly increasing error for small sizes of training data set. For small sizes of the training data set, the error increases of the ME model (*diamond*), the ensemble (*upward-pointing triangle*), the RBF (*circle*), and the MLP (*square*)

Table 2 Responsibilities of the mode models for data of the corresponding driving mode

HME model	driving mode (in %)			
	brake	pure electric drive	drag	hybrid
MLP gate	98	94	93	98
cluster gate	89	92	85	87

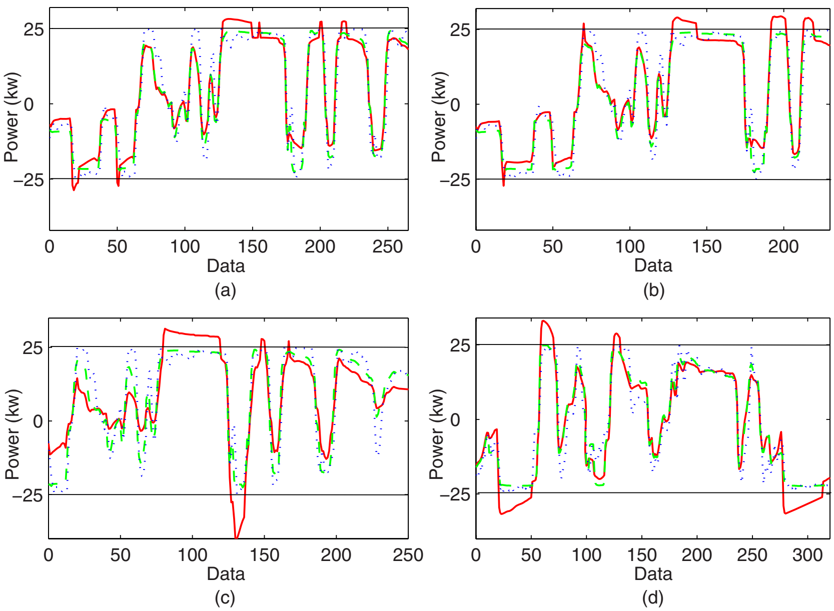


Fig. 9 The plots show examples of violations of the chemical battery limits (depicted as *horizontal lines*) of (**a**) the ME (*solid line*), (**b**) the ensemble (*solid line*), (**c**) the RBF (*solid line*), and (**d**) the MLP (*solid line*). The corresponding target values and outputs of the HME with MLP gate are depicted as *dotted* and *dashed lines*

performance of the models is approximately equal. However, for smaller sizes of the training data set, the error increases for purely data-driven models.

The chemical battery limits are violated by all models, except the HME models, because they predict energy flows that cannot be provided by the battery as shown in Fig. 9. The necessary information about these limits is not contained in the training data, but they are implicitly contained in the given knowledge-based models. Thus, purely data-driven models are not capable to maintain these limits.

Table 3 Predictive error of the models for different sizes of the training data set on the hybrid vehicle data

model	size of training data set D					
	D	$D/2$	$D/4$	$D/8$	$D/16$	$D/100$
HME with MLP	2.14	2.13	2.14	2.16	2.20	2.30
HME with cluster gate	2.31	2.31	2.33	2.35	2.39	2.51
ME	2.88	2.91	2.96	3.08	3.22	4.33
ensemble	2.47	2.49	2.54	2.64	2.82	3.81
RBF	3.96	4.00	4.08	4.21	4.37	5.26
MLP	2.61	2.62	2.69	2.82	2.99	4.35

6 Conclusions

By applying the proposed ensemble learning model, it is possible to fuse information from multiple sources, represented by knowledge-based models. In this way, information can be incorporated in the modeling process that is not contained in the training data. For example constraints can be implicitly contained in knowledge-based models but domain experts may not be able to describe them, since the domain experts do not explicitly perceive it or they cannot define such constraints. Data-driven submodels are used to complement knowledge-based ones with respect to the coverage of the input space. To be able to integrate given knowledge-based models into the process of simultaneously training the data-driven submodels and a gate model, it is crucial to incorporate the validity ranges of the knowledge-based models. A further advantage is the need of fewer training data, which is beneficial if a few training data are available or if the acquisition of data is expensive.

We have tested the HME models successfully for the simulation of electrical energy flow in the electrical system of a hybrid electric vehicle. They have achieved a superior performance compared to previous approaches.

References

1. Abonyi, J., Madar, J., Szeifert, F.: Combining first principles models and neural Networks for generic model control. Springer Engineering Series (2002)
2. Bengio, S., Marcel, C., Marcel, S., Mariéthoz, J.: Confidence measures for multimodal identity verification. Information Fusion 3, 267–276 (2002)
3. Beyer, J., Heesche, K., Hauptmann, W., Otte, C.: Combined knowledge-based and data-driven modeling by heterogeneous mixture-of-experts. In: Mikut, R., Reischl, M. (Hrsg.) Workshop Computational Intelligence, vol. 18, pp. 225–236. Universitätsverlag Karlsruhe (2008)
4. Bishop, C.M., Svensén, M.: Bayesian hierarchical mixtures of experts. In: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence, pp. 57–64 (2003)

5. Bloch, I.: Information Fusion in Signal and Image Processing: Major Probabilistic and Non-Probabilistic Numerical Approaches. John Wiley & Sons Inc., Chichester (2008)
6. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
7. Dasarathy, B.V.: Information fusion - what, where, why, when, and how? Information Fusion 2, 75–76 (2001)
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39, 1–38 (1977)
9. Durrant-Whyte, H.F.: Sensor models and multisensor integration. International Journal of Robotics Research 7, 97–113 (1988)
10. Ferrari-Trecate, G., Muselli, M.: A new learning method for piecewise linear regression. In: Proceedings of International Conference on Artificial Neural Networks, pp. 444–449 (2002)
11. Freund, Y., Schapire, R.E.: Decision–theoretic generalization of on–line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
12. Georgieva, P., de Azevedo, S.F.: Neural network-based control strategies applied to a fed-batch crystallization process. International Journal of Computational Intelligence 3, 224–233 (2006)
13. Hansen, L., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001 (1990)
14. Hashem, S.: Optimal linear combinations of neural networks. Neural Networks 10, 599–614 (1997)
15. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3, 79–87 (1991)
16. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Computation 6, 181–214 (1994)
17. Krogh, A., Vedelsby, J.: Neural networks ensembles, cross validation, and active learning. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 231–238. MIT Press, Cambridge (1995)
18. Quandt, R.E.: The estimation of the parameters of a linear regression system obeying two separate regimes. Journal of the American Statistical Association 53, 873–880 (1958)
19. Schapire, R.E.: The strength of weak learnability. IEEE Transactions on Pattern Analysis and Machine Intelligence 5, 197–227 (1990)
20. Schlang, M., Feldkeller, B., Lang, B., Poppe, T., Runkler, T.: Neural computation in steel industry. In: Proceedings European Control Conf. ECC 1999, VDI-Verlag (1999)
21. Titterington, D.M., Smith, A.F.M., Makov, U.E.: Statistical Analysis of Finite Mixture Distributions. John Wiley, New York (1985)
22. van Lith, P.F., Betlem, B.H.L., Roffel, B.: Combining prior knowledge with data driven modeling of a batch distillation column including start-up. Computers and Chemical Engineering 27, 1021–1030 (2003)
23. Wald, L.: Some terms of reference in data fusion. IEEE Transactions on Geoscience and Remote Sensing 37, 1190–1193 (1999)
24. Waterhouse, S., MacKay, D., Robinson, T.: Bayesian methods for mixture of experts. In: Advances of Neural Information Processing Systems, pp. 351–357 (1996)

25. White, F.E.: Data fusion lexicon. Joint Directors of Laboratories, Technical Panel of C3, Data Fusion Sub-Panel, Naval Ocean Systems Center, San Diego (1987)
26. Wolpert, D.H.: Stacked generalization. Neural Networks 5, 241–259 (1992)
27. Xu, L.: RBF nets, mixture experts, and bayesian ying–yang learning. Neurocomputing 19, 223–257 (1998)

7

Towards Developing Intelligent Autonomous Systems in Psychiatry: Its Present State and Future Possibilities

Subhagata Chattopadhyay and Dilip Kumar Pratihar

Abstract. Healthcare Informatics (HI) remains a pivotal factor to successful implementation of technology in medicine. Knowledge Engineering (KE) is the key towards developing Intelligent Decision Support Systems (IDSS) and it is an important area under the HI. KE techniques extract hidden knowledge from a set of raw data and the knowledge thus extracted, constitutes the knowledge base of an IDSS. Applications of KE techniques in medicine are gaining popularities for extracting the hidden patterns in biological and clinical data that are often subjective in nature. The key impetus is to diagnose diseases, decide on treatment plans, and predict prognosis. Mental illness being one highly complex in nature is often either under-diagnosed or over-diagnosed by the psychiatrists due to involved subjectivities with the signs, symptoms, course of morbidity, and treatment-responses. Given the fact, the key impetus for adopting KE techniques in mental illnesses is to analyze the clinical data for extracting information and process it into knowledge after scientific realization by the domain experts. Knowledge, thus extracted, could be useful to develop various IDSS to automate the diagnostic process and predicting prognosis for assisting the clinicians by virtue of its speed and precision. This chapter is a meta-analysis of current researches on the applications of various KE techniques and IDSS in psychiatry. The contribution of this chapter not only lies on the in-depth and critical review of the present state of research, but also the vision that is needed to successfully design, develop and practically implement autonomous intelligent systems in day-to-day psychiatry practice.

Subhagata Chattopadhyay
Reader
Department of Biomedical Engineering,
Manipal Institute of Technology, Manipal University
Manipal 576 104, India

Dilip Kumar Pratihar
Professor, Department of Mechanical Engineering,
Indian Institute of Technology Kharagpur 721302, West Bengal, India
e-mail: dkpra@mech.iitkgp.ernet.in

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 143–166.
springerlink.com

List of Abbreviations

AI	Artificial Intelligence
AUD	Alcohol Use Disorders
BPRS	Brief Psychiatric Rating Scale
CDSS	Computerized Decision Support System
CFG	Convergent Functional Genomics
DD	Differential Diagnosis
DOE	Design of Experiments
DSM	Diagnostic and Statistical Manual
EFC	Entropy-based Fuzzy Clustering
FCM	Fuzzy C-Means
FS	Forward Selection
GA	Genetic Algorithm
GWAS	Genome-Wide Association Studies
HAMA	Hamilton Anxiety Scale
HAMD	Hamilton Depression Scale
HI	Healthcare Informatics
HIS	Healthcare Information System
HOCTA	Hierarchical Optimal Classification Tree Analysis
IDSS	Intelligent Decision Support Systems
KB	Knowledge Base
KE	Knowledge Engineering
KMS	Knowledge Management System
MDD	Major Depressive Disorders
PA	Pattern Array
PANSS	Positive and Negative Syndrome Scale
PD	Provisional Diagnosis
PDG-ACE	Prioritizing Diseases Genes by Analysis of Common Disorders
PTSD	Post Traumatic Stress Disorder
RF	Random Forest
SOM	Self-Organizing Map

1 Introduction

Healthcare systems are flooded with various types of *data*, ranging from biological, clinical (including pathological, radiological, electrophysiological and so forth) to demographic and administrative ones. Some of these data are objective in nature and are directly measurable, e.g., blood pressure, temperature, blood biochemistry, anatomical data, drug dosage, and various others. On the other hand, most of the clinical data, especially symptoms and signs are often highly subjective, i.e., cannot be measured directly, e.g., anxiety, depression, feeling of pain or well-being. Moreover, interpretations of these data are equally skewed among the experts, e.g., two clinicians with similar level of expertise often interpret patient's symptoms differently by assigning different emphasis to it. One of the reasons could be the difference in their individual Knowledge Base (KB), intuitions, and reasoning abilities. These capabilities together with their clinical skills could be termed as the Medical logic [1]. Presentations of psychiatric diseases are also varying among the patients and often there are co-morbidities, i.e., associated with illnesses that could be either psychiatric or non-psychiatric. Hence, the net result is either under-diagnosis or over-diagnosis of an illness resulting into increased sufferings of the patients. Fig.1 displays the issues related to medical diagnosis.

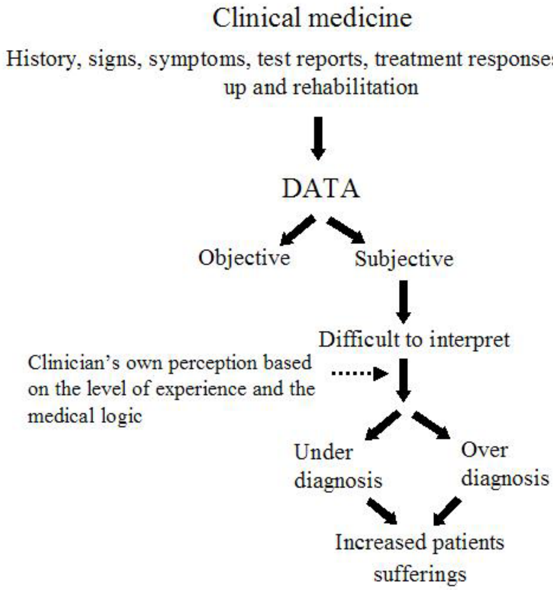


Fig. 1 Issues related to the medical diagnosis

1.1 Knowledge Engineering (KE) and Intelligent Decision Support Systems: The Parent and the Child

Knowledge Engineering (KE) deals with the discovery of interesting patterns from data sets, stored in database, data warehouse, or any other information repository [2]. KE is a budding multidisciplinary field stemming from different mathematical and engineering domains, such as database management system, data warehouse, statistics, machine learning, data visualization, information retrieval and processing, and high performance computing [2]. Currently, there are other contributing fields in KE, e.g., neural networks, optimization techniques, pattern recognizers, spatial data analysis, image databases, signal processing, and many application fields, such as engineering, business, economics, earth sciences, health sciences (medicine) and bioinformatics [2] This chapter focuses on KE in Medicine, especially in mental health. Followings are the major issues associated with KE [2] that are often encountered in medical data:

(i) Mining different kinds of knowledge that depends on user's interest (e.g., a set of sign-symptoms could be viewed and interpreted differently by doctors based on their individual knowledge-bases);

(ii) Interactive mining of knowledge at multi-level of abstraction – it needs appropriate sampling techniques to facilitate interactive data exploration;

(iii) Incorporation of background knowledge – basic to advanced level of expertise on the field under study remains key to correctly interpret the extracted knowledge and make it usable in real-world scenarios;

(iv) Query language and ad-hoc KE – relational query languages allow users to pose ad-hoc queries for data retrieval and its integration with data warehouses or databases, query languages offer flexible and efficient knowledge mining;

(v) Handling noisy and incomplete data: data cleaning, data analysis and outlier mining methods handle these issues;

(vi) Presentation and visualization of KE results – the knowledge thus discovered, should be expressed in high-level languages, visual representations, or any other expressive forms, so that knowledge can be easily understood and directly usable by the users, and

(vii) Knowledge incorporation into an autonomous system - the extracted knowledge could further be used for machine learning and in turn, developing *Intelligent Decision Support Systems (IDSS)* in medicine (refer to Fig. 2).

As mentioned above, one of the applications of the derived knowledge is the development of *IDSS* for automating screening, diagnosis and making prognostic evaluations of an illness. Knowledge Base (KB), which is the heart of any IDSS, is constituted by the knowledge derived by applying suitable KE technique(s) on a set of data. Hence, IDSS could be the child of KE.

It is interesting to note that the knowledge, extracted using KE techniques is considered to be *raw* in nature, unless the human experts effectively interpret or in computer term, process it. The processing of the *raw* knowledge into a usable one is the most useful step towards practical implementation and applicable in real-world situations (refer to Fig. 2). Hence, the performance of the KB of any autonomous system depends on the goodness of both extracted knowledge and its processing by domain experts.

The popularity that lies with developing and applying IDSS is that IDSS attempts to mimic the way human experts interpret a given set of information, e.g., doctors diagnosing diseases or treating patients in their practice. Other useful feature of an IDSS may be its ability to learn with time, especially when learning algorithms are used to develop its inference engine. This feature is similar to the way doctors learn by seeing patients and gain maturity in giving their decisions for diagnosis and treatment of the patients. Given these two considerable qualities of any IDSS, researchers are attracted to design, develop, and use IDSS for solving medical diagnosis problems.

A couple of other important features of IDSS are *precision* and *speed*. The former is embedded by the mathematical logic and algorithms used to develop its inference engine, while the later is achieved by virtue of computation. Hence, diseases that require a much focused diagnosis and specific treatments can be

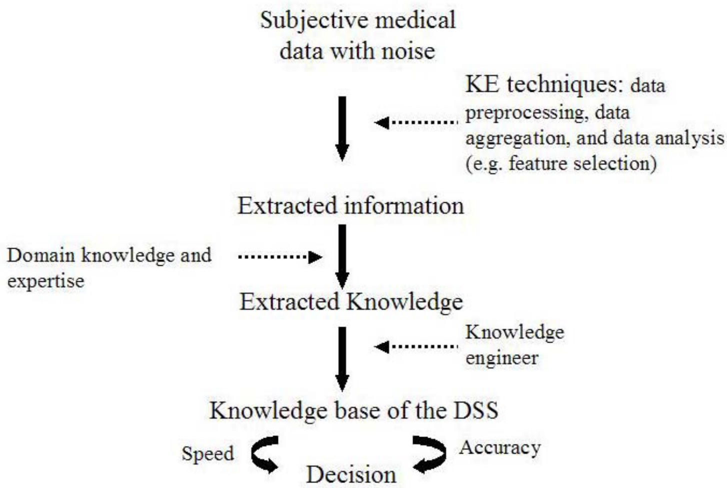


Fig. 2 Raw data to Intelligent Decision Support Systems: a few important steps

diagnosed with a reasonably good accuracy by the IDSS. On the other hand, diseases that consume much time to get diagnosed (e.g., psychiatric diseases) can be screened early by virtue of computational speed. That is why, IDSS are so demanding in today's medical informatics research.

1.2 Knowledge Engineering and IDSS in Psychiatry: Predicted Advantages

Psychiatry is a specialized domain under neuro-medicine that studies mental illnesses. Psychiatrists are the doctors, who are specialized to diagnose and manage (that is, investigate and treat medically) patients suffering from mental diseases. The challenges in screening an illness, understanding disease-course, planning the best management, and determining its prognostic features lie on the following facts [3]:

1. **Onset of the disease:** In psychiatry, the onset of an illness is often unrecognized because of vague signs and symptoms.
2. **Delayed diagnosis:** Due to persistence of vague symptoms for a long time, doctors often take a long time to diagnose an illness.
3. **Bizarre course:** Psychiatric illnesses often have very bizarre course for prolonged period even after the start of preliminary treatment and more importantly, no single investigation can pinpoint an illness.
4. **Multiple diseases:** Symptoms and signs are often a complex mixture of associated psychiatric and or non-psychiatric illnesses, e.g., a patient suffering from depression often suffers from anxiety disorder. On the other hand, cancer patients often suffer from depressions.

5. Varied presentation of a particular illness from one patient to other.
6. Varied interpretations of signs and symptoms by the doctors due to the differences in their individual medical logic.
7. Diagnostic difficulties, such as under-diagnosis or over-diagnosis cause increased patients' sufferings (morbidity, mortality, and socioeconomic loss).
8. Scarcity of clinical psychiatry data due to prevailing social taboos, lack of awareness among common people, and fright of economic and social loss.

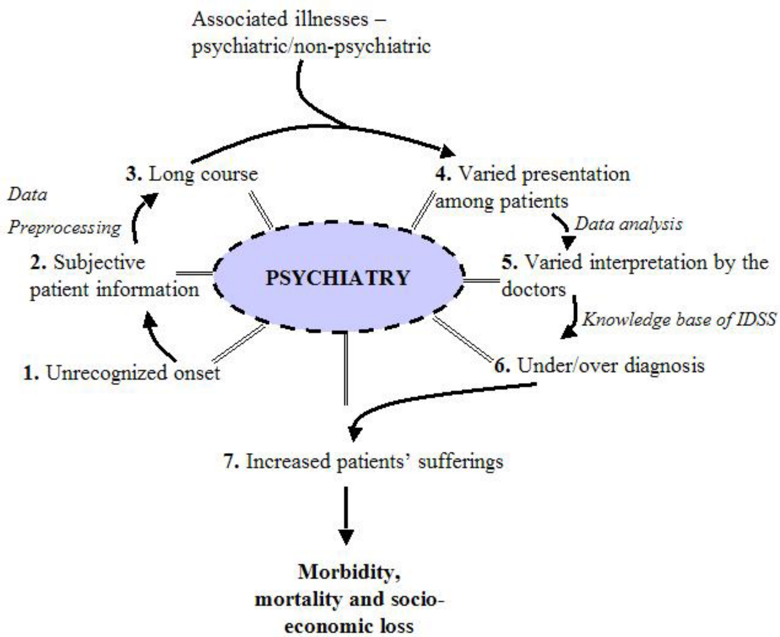


Fig. 3 Issues with screening and diagnosis of psychiatric illnesses

Given this picture, researchers attempted to use techniques to preprocess medical data (history, signs, symptoms, treatment response etc.) as the basic requirement of knowledge engineering, analyze the data to find out patterns of illnesses using domain knowledge. Psychiatric diseases are often associated with co-morbidities and therefore, often the signs and symptoms are misleading. Investigators also tried to differentiate the amalgamated illnesses using clustering or classification methods, discussed in the following sections. This even could be termed as screening of illnesses. The knowledge, thus extracted, is incorporated into the knowledge base of an autonomous system to make it *intelligent* in decision-making. The objective of the use of an autonomous IDSS is to bring accuracy and speed in diagnosis and prevent under/over diagnosis, often occurred in psychiatric practice. The overall objective is to cater a better patient care (refer to Fig. 3). Learning algorithms are used to mature the decision-making capacity of these systems with time.

Screening of an illness remains the first and most useful step to arrive into diagnosis (refer to Fig. 4). During screening, patient's information, such as history, signs and symptoms are assessed by the doctors using their medical logic (explained below).

Doctors try to find clinical patterns in the onset, chief-complaints, course, and treatment-response in their patients. They try to extract the hidden patterns using their clinical experience, theoretical knowledge, intuition, common sense logic, and historical evidences – these may together be termed as the *Medical logic* [1].

As the first step of screening, the patterns of the history, signs, symptoms, treatment response are initially analyzed based on the medical logic and then are matched with the salient patterns of plausible diseases. Based on the degree of closeness, the diseases are ranked as follows: the closest match is ranked top and then followed by other. These are popularly known as the *Differential Diagnosis (DD)*.

After the DD is made, doctors prescribe investigations that to be carried out to pinpoint their diagnosis further and recommend initial treatments. Based on the test reports and most importantly patient's response to the given treatment, doctors sort the diseases into some ranks. The disease that tops becomes the most possible diagnosis, called as *Provisional Diagnosis (PD)* in medical terminology. There are possibilities that more than one disease may rank top, as often encountered in psychiatry. In that case, doctors assign more sophisticated and definite test and treatment to further rank. The better the doctor can differentiate it; the better will be the treatment.

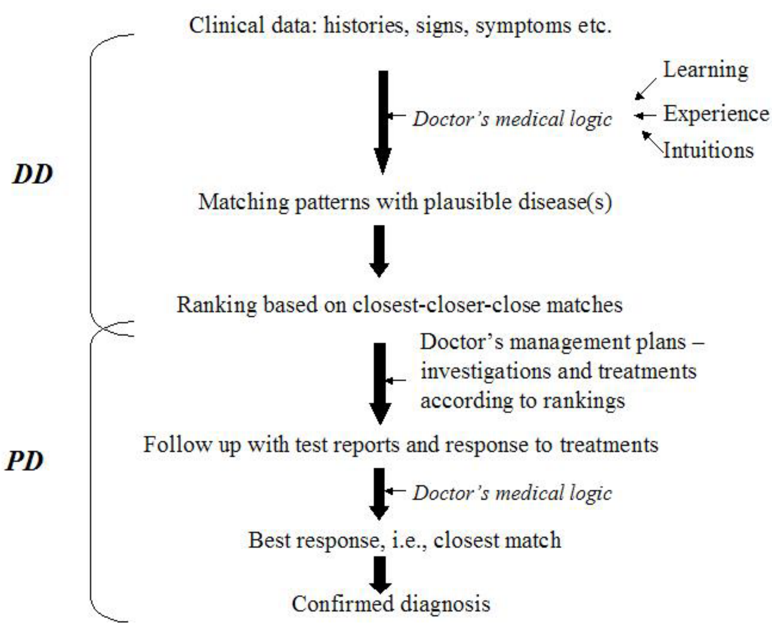


Fig. 4 Steps of screening-to-diagnosis in medicine

There is no doubt that screening and diagnosis is a complex and time-consuming process, especially when the signs and symptoms are highly subjective in nature, as seen in psychiatric illnesses. However, for an initial screening and then diagnosis, psychiatrists rely on various mental illness rating tools, e.g., Beck's Suicide Intent Scale [4], Brief Psychiatric Rating scale [5], Pierce's Suicide Intent Scale [6], Positive and Negative Syndrome Scale [7], Hamilton Anxiety Scale [8] and so forth. However, the use of these rating tools solely does not solve the diagnostic inaccuracies for all cases. One of the most important reasons behind this issue is manual interpretation of the symptoms that is individualized and largely guided by one's intuitions and experience. It leads to varied diagnoses. Hence, there are needs for bringing more accuracy along with the required speed for diagnosis of psychiatric illnesses to render a better patient care. In Fig. 3, the authors propose to apply various KE techniques, e.g., data preprocessing techniques to reduce the level of noise and subjectivities; classification and clustering algorithms, search algorithms, decision trees etc. to handle this screening and diagnosis issue, and IDSS for assisting psychiatrists to arrive into a diagnosis.

2 Knowledge Engineering in Psychiatry: Current State of Art

KE remains a useful and popular method of extracting hidden information by analyzing the class, patterns based on some feature selections [2]. It is very similar to what the psychiatrists do in their chamber, that is, extracting patterns from a set of given signs and symptoms, processing the information, thus extracted, using their medical logic, and finally matching these with a set of possible diseases. From the medical history, signs, symptoms, treatment response, and medical test reports, they confirm the diagnosis. Researchers have tried to model the method of screening and diagnosis of illnesses with the help of KE techniques.

This section has discussed recent applications of various KE techniques to understand the (1) patho-physiology of psychiatric disorders at the molecular and genetic levels and (2) clinical knowledge mining.

2.1 Knowledge Engineering at the Genetic and Molecular Levels

Most of the psychiatric disorders are genetically predisposed by nature, e.g., schizophrenia [9] and therefore, has familial predispositions. Identification of candidate genes is a complex task but very important to understand the pathways and mechanism of the diseases [10]. Le-Niculescu et al. [11] studied the utility of Convergent Functional Genomics (CFG) approach (commonly used by the bioinformaticians) as a way of mining first-generation Genome-Wide Association Studies (GWAS) datasets to identify *significant* genetic signals that remain under explored, when studied using genetics-only approach. The authors conclude that their study most comprehensively integrates genetics and functional genomics in Bipolar disorder yielding a series of novel candidate genes and blood biomarkers to understand the mechanism and pathways of the illness. The study of Le-Niculescu et al. [11] could also be very useful to develop new anti-bipolar drugs and focusing better towards the personalized medicine approaches.

McEachin et al. [12] applied Prioritizing Diseases Genes by Analysis of Common Elements (PDG-ACE) algorithm to show that two candidate genes at the backdrop of associated Major Depressive Disorders (MDD) and Alcohol Use Disorders (AUD) share significant similarities that is governed by environmental exposure to alcohol. This study also proves how integrated bioinformatics knowledge engineering approach could help us understanding the etiopathology and co-morbidities of psychiatric illnesses.

Kafkafi et al. [13] studied applications of Pattern Array (PA) in vivo (on animal model) to classify the psychopharmacological drugs based on their carcinogenicity. The main objective of the study was to discover how gene expression profiles could predict such adverse effects. The drugs under study were psychomotor stimulants, opoids, and psychotomimetic drugs. PA could be able to classify the drugs based on their dosages, dependencies, represented by rodent movements with about 100000 complex patterns. This research could be very useful to understand the drug de-sensitivities often associated with psychiatric disorders.

2.2 Knowledge Engineering on Clinical Psychiatry Data

Strong but focused researches are also the other way of understanding psychiatric diseases based on clinical data mining. Baca-Garcia et al. [14] studied 509 adult suicide attempters using traditional statistical technique and multivariate statistics as a new data mining technique to explore the factors that significantly guide psychiatrists to admit the patients in emergency. The authors found that data mining multivariate technique could give better prediction (99% sensitivity and 100% specificity). In another study, Baca-Garcia et al. [15] attempted to explore the variables that have associations with familial suicide attempts in a group of suicide attempters. They applied two data mining techniques – Random Forest (RF) and Forward Selection (FS). RF is a special technique in data mining that can develop multiple classification trees, where each tree votes for classes or even assigns new class and by counting the maximum number of votes the final selection is made. The tree can be grown as large as possible without pruning. By using RF, the authors were able to classify patient who attempted suicides in some part of their life times. They also could derive a new class having alcohol abuse as the co-morbidity. On the other hand, FS starts without any variable in the model and then, tries out the variables one by one and finally measures the level of significance. In that study, FS had been used to find out the significant classes. Alcohol abuse as the cause of suicidal attempts remains the new class that has immense clinical implications for future studies.

Marinic et al. [16] also used RF classifier on a set of 120 psychiatric patients, half of them were suffering from Post Traumatic Stress Disorder (PTSD) and the rest other than PTSD. The aim of the study was to explore the salient diagnostic features of PTSD in the given population. They developed two predictive models: one was based on structured psychiatric interview and the other one was using Clinician Administered PTSD Scale, i.e., CAPS, Positive and Negative Syndrome Scale, i.e., PANSS, Hamilton Anxiety Scale or HAMA, and Hamilton Depression Scale or HAMD. They observed that the first predictive model was able to distinguish PTSD from other neurotic, stress-related or somatoform disorders.

Multidimensional classifier was also used in a study to correlate alcoholism and homicidal tendencies (Reulbach et al. [17]). Multidimensional classification system of Lesch and Cloninger-based results showed that there is a high chance of committing murders by those who abuse alcohol, especially having co-morbidity with cyclothymias, and anxiety disorders.

Chattopadhyay et al. [18] and Chattopadhyay et al. [19] studied suicidal risks in a set of real-world psychiatric patients. The objective of those studies was to explore behavioral patterns among patients, who either attempted or committed suicide using Pierce's Suicide Intent Scale [5] that captures the behavioral evidence at the backdrop of suicide in a population at risk (refer to Fig. 5). In this figure, it is evident that the significant factors, such as *B* (timing of suicidal act), *J* (premeditation to commit such a lethal act), and *M* (culmination into death without medical attention) are vulnerable *moderate risk* predictors. Similarly, *L* (presumption of predictable outcome) could predict the *severe risks* for this given set of patients.

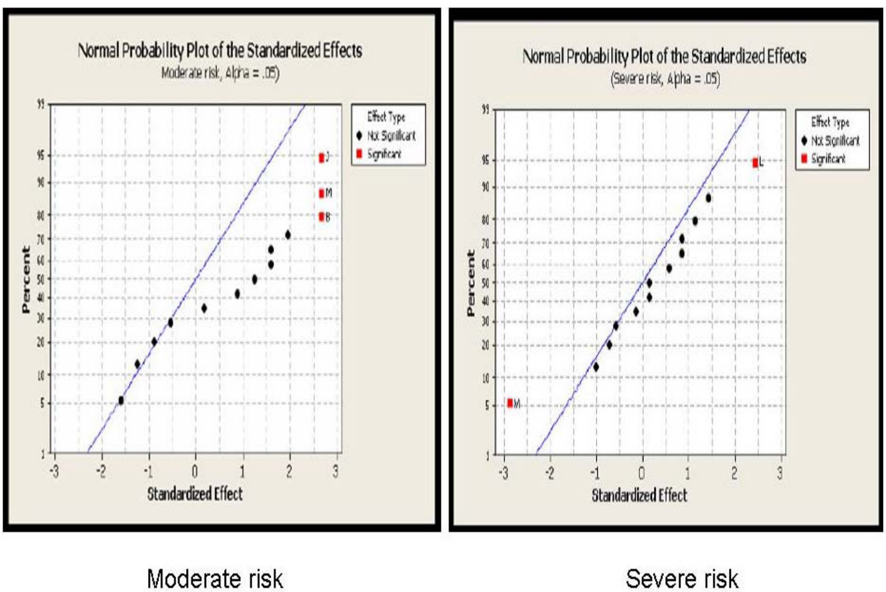
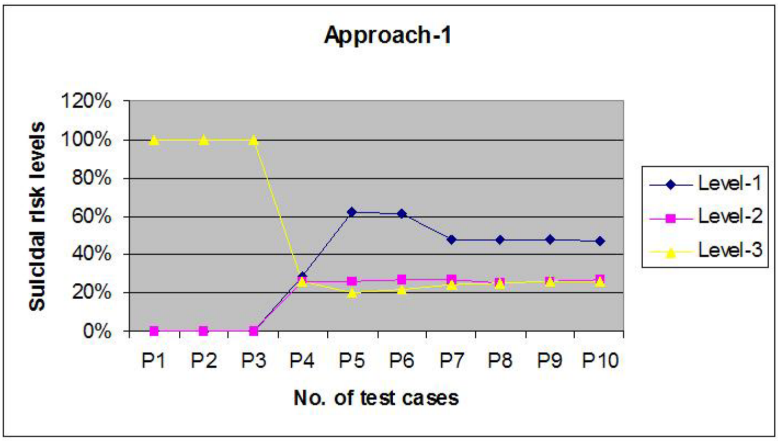
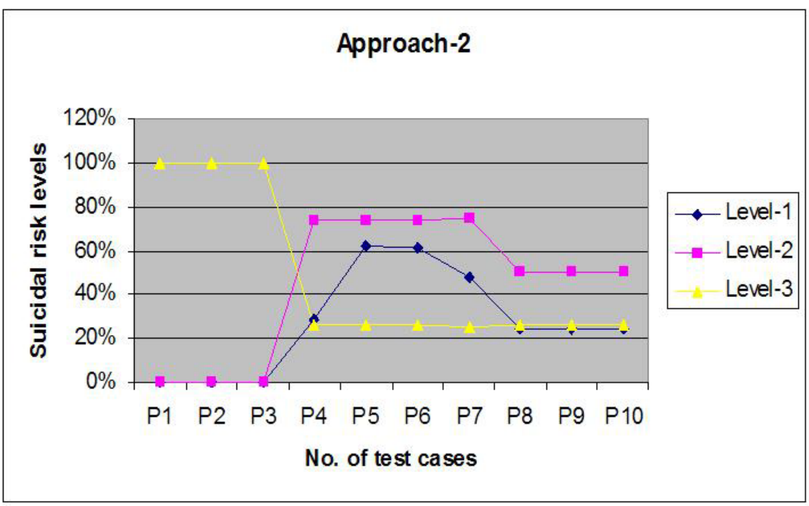


Fig. 5 Significant factors behind moderate and severe suicidal risks

Chattopadhyay et al. [19] used similarity-based class measure to identify a set of significant behavioral patterns behind mild risk (suicidal thoughts), moderate risk (suicidal attempts), and severe risk (committed suicide) in males and females with four age groups (<40, 40-60, 60-80, and >80 years). Two approaches were developed. The first approach was based on the PSIS-based data only, where *age* and *sex* of the patients were not considered. The second approach, on the other hand, added *age* and *sex* as two new factors to the PSIS-based data set. The objective of the study was to test whether by adding more factors horizontally into a matrix could improve the performance of the classifier. Euclidean distance-based similarity measure with pair-wise comparison and statistical significance tests were used to find out the neighboring classes (mild, moderate or severe risks) for a given test data (refer to Fig. 6). The study showed that the males aged 40-60 years with (J) *premeditation*; (L) *understanding of consequence*; and (A) *isolation* remain the most vulnerable behavioral patterns behind suicidal risk. Finally, suicidal risk classes could be predicted for three levels: mild, moderate and severe risks for a set of ten test cases and the results are shown in Figs.6a and 6b. The results indicate that the developed similarity-based classifier is able to predict the risk levels for a new case.



(a)



(b)

Fig. 6 Prediction of suicidal risk levels by the classifier using (a) Approach-I, (b) Approach-II

Nebeker et al. [20] used Hierarchical Optimal Classification Tree Analysis (HOCTA) as the KE method for rapidly developing clinically meaningful surveillance rules for a set of 3987 administrative patient data. The authors explored and validated surveillance rules for drug-induced bleeding, delirium, and psychosis as the adverse effects of drugs using HOCTA.

Chattopadhyay et al. [1][21] used statistical data mining techniques to predict seven types of psychosis in adults, such as schizophrenia, mania, depression with psychosis, delusional disorder, schizoaffective disorder, organic psychosis, and catatonia. They used Brief Psychiatric Rating Scale (BPRS-F2) to capture the

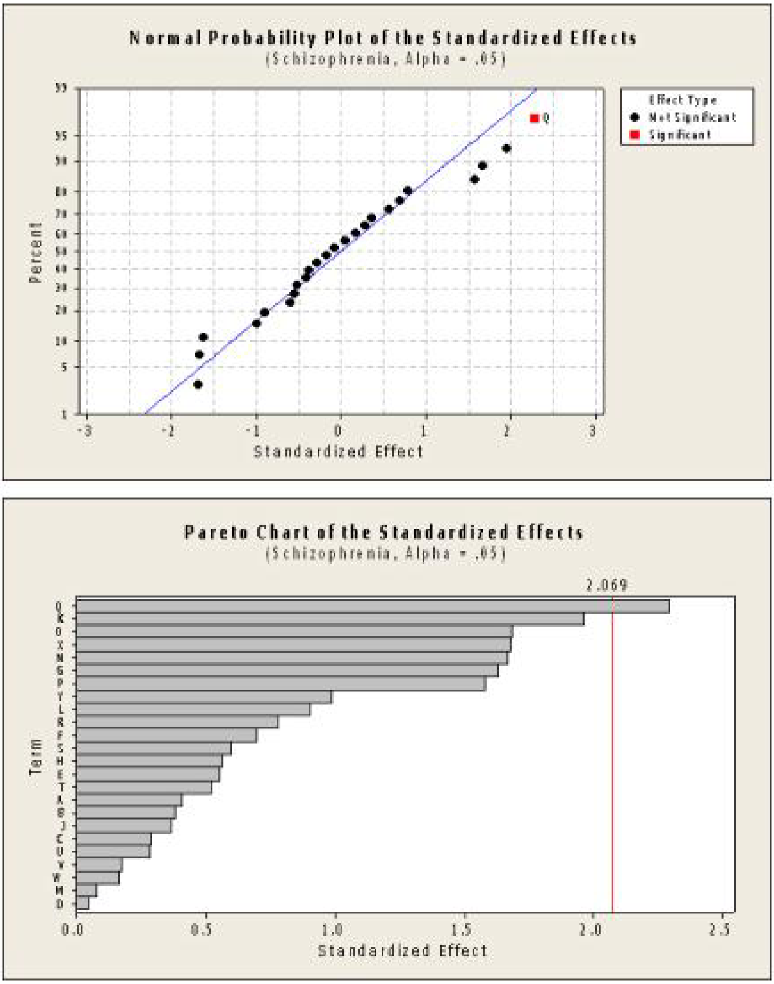


Fig. 7 Effect plot and Pareto chart showing Q for screening schizophrenia

symptoms of these disorders and performed statistical analysis using multiple linear regressions with significance testing. The aim was to extract the salient diagnostic features. They applied Plackett Burman Design of Experiments [22] as the fractional factorial design to measure the effects of various factors (derived based on BPRS-F2 [5]) on these seven adult psychoses. Fig. 7 displays the effect plot and Pareto chart, which show a significant factor Q denoting *blunted affect* as the salient feature to diagnose schizophrenia. Similarly, other significant factors were extracted from the given set of patient data for screening of remaining illnesses (e.g., Mania, Depression with psychosis, Delusional disorder, Schizoaffective disorder, Organic psychosis, and Catatonia). This study claims that regression analysis carried out based on statistical Design of Experiments (DOE) is a useful knowledge engineering technique to extract hidden knowledge from a set of psychiatry data.

In psychiatry, patients often present with symptoms that are similar in many psychiatric as well as non-psychiatric illnesses and there are possibilities of the mixture of disorders. As diseases have got no clear-cut boundaries, it is difficult to differentiate one illness from other. Given this scenario, Chattopadhyay et al. [23] applied various fuzzy clustering techniques to understand and explore the contribution of illnesses at the backdrop of a given set of symptoms. They used Fuzzy C-Means (FCM) [24] and Entropy-based Fuzzy Clustering (EFC) [25] algorithms. Both the algorithms work based on the concept of similarity among the data points. They utilized FCM and EFC algorithms on a set of real-world clinical psychiatry data and were able to cluster the diseases (schizophrenia, mania, depression mixed with psychosis, delusional disorder, schizoaffective disorder, organic psychosis, and catatonia). Multidimensional data (representing the number of patients) were then mapped into two-dimension for visualization using a Self Organizing Map (SOM) (refer to Fig. 8).

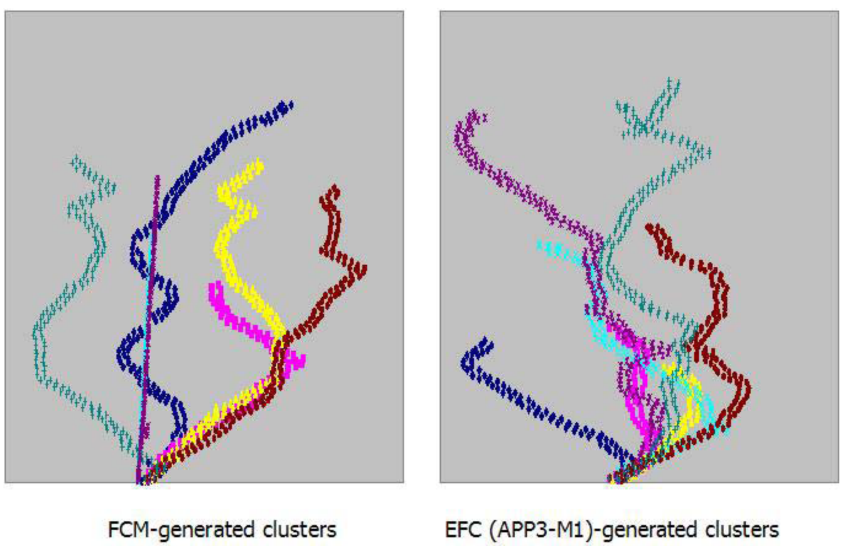


Fig. 8 SOM-based data visualization of seven clusters denoting seven diseases

3 Towards Developing Autonomous Intelligent Decision Support Systems in Psychiatry: Current State of Art

The ultimate application of KE techniques, as mentioned in Section 1, is to develop an IDSS by incorporating extracted knowledge into the KB of the control system. In medicine, IDSS are developed by aiming towards automating the process of screening, forecasting, diagnosis, and determination of prognosis (i.e., future outcome) of the illness.

In psychiatry, researchers welcome the concept of IDSS presuming that it may be more accurate and faster than that of the conventional screening tools [26, 27,28]. Moreover, human interpretations using screening tools often vary to a large extent producing confusions among the doctors and psychologists [29].

Attempts were made to develop autonomous tools for diagnosis of psychiatric diseases since couple of decades. Traditional Artificial Intelligence (AI) techniques were used to develop these tools utilizing the concept of probability theory [30]. In a nutshell, these are called *hard computing* techniques.

3.1 Hard Computing Techniques and Autonomous Systems in Psychiatry: A Review

Hard computing techniques are the first initiative towards making the computer to think faster and give the diagnostic inference. Thirty-odd years from now, two researchers Heiser and Brookes developed HEADMED [31] extending the concept of MYCIN [32]. HEADMED was developed to automate prescription processes for neurotic and psychotic illnesses. Apart from just inferring the best drug for the disorders, HEADMED also could provide with valuable information about dosing, route of administration and side effects. Clinging to the concept of EMYCIN many other expert tools were developed over the period of time, e.g., BLUE BOX [33], PATHFINDER KET [34], PSYXPERT [35], ADINFER [36], SET COVERING MODEL [37], Support Vector Machine (SVM)-based approach [38] for suicidal risk evaluations and so forth. A detailed discussion of these tools could not be made here due to space constraint and to be consistent with the original focus of the chapter.

However, due to inherent rigidity within these algorithms and lack of learning abilities, over the period of time, AI researchers inclined towards adopting *soft computing* techniques [39].

In an interesting study, Berman and Fors [40] compared *Computerized Decision Support System (CDSS)* with that of traditional *paper and pencil method SCID1* for psychiatric diagnosis. The hypothesis derived by them is that computerized decision support, being a novel method in medical diagnosis, may perform better that the traditional paper-pencil method SCDI1. Sixty-three clinicians volunteered the study and they are instructed to solve two paper-based cases using either CDSS or manually. The study showed that CDSS and traditional paper and pencil-based diagnostic method were equally effective to arrive at the diagnosis. However, paper-and-pencil method performed better in terms of both accuracy and speed than that of CDSS to diagnose psychiatric illnesses. The major limitation of the study was that most of the clinicians were ignorant of the CDSS. They probably were not accustomed with its use due to the lack of training. Technical ignorance and a strong tendency clinging towards the conventional practice are possibly the two most important reasons behind the apparent failure of many of these expert systems in psychiatry.

3.2 Soft Computing Techniques and Intelligent Autonomous Systems in Psychiatry: A Critical Review

Soft Computing is a family consisting of the techniques like fuzzy logic, neural networks, genetic algorithms and others, and their possible combinations, namely genetic-fuzzy system, genetic-neural system, neuro-fuzzy system and so on [39]. One of the key reasons for anchoring to these algorithms is to incorporate *learning* and hence, *intelligence* towards developing a complete IDSS.

Design and development of an IDSS in Psychiatry is a bit tricky issue because of the nature of psychiatry data, lack of multidisciplinary (technology and mental health) domain knowledge, involved complexities, lack of awareness and probably low level of acceptance among the psychiatrists. These are some of the important reasons at the backdrop of scanty amount of reported work.

Literature review shows that neural network poses to be the mostly applied soft computing technique in psychiatry. Neural networks mimic the way we reason by using synapses among neurons in the brain. Lowell and Davis [41] used feed-forward neural network for classifying the groups of psychiatric diseases based on hospital-stay. Zou et al. [42] classified psychiatric diseases using artificial neural network. Buscema et al. [43] used neural net for classifying various eating disorders. Jefferson et al. [44] utilized artificial neural net to predict occurrence of depression after manic episodes in a group of patients. Yana et al. [45] used a three-layered perceptron and interview-based neural network classifier for differentiating psychotic disorders (mood disorders and schizophrenia) from neurotic disorders (anxiety). They compared the screening performance of their tool with that of two experienced psychiatrists and it was found to be satisfactory. Clinging to neural net is probably because of its easy implementation and iterative reduction of error through back-propagation algorithm. However, neural net's important drawback lies on its black-box view.

Fuzzy logic [46] is a newer version of traditional mathematical logic that trusts on *possibility* rather than *probability*. It mimics the reasoning methods of human brains and is claimed to be a useful tool for medical diagnosis. There are several applications of fuzzy sets, e.g., fuzzy clustering, fuzzy logic controllers, and others. Fuzzy sets are unique techniques to handle real-world datasets that have no well-defined boundary. Fuzzy clustering techniques are the most suitable techniques for differentiating look-alike clusters by virtue of unsupervised learning, especially in those cases where case-to-case differentiations are much more difficult, as seen in psychiatric diseases that are often a mixture of many look-alike diseases.

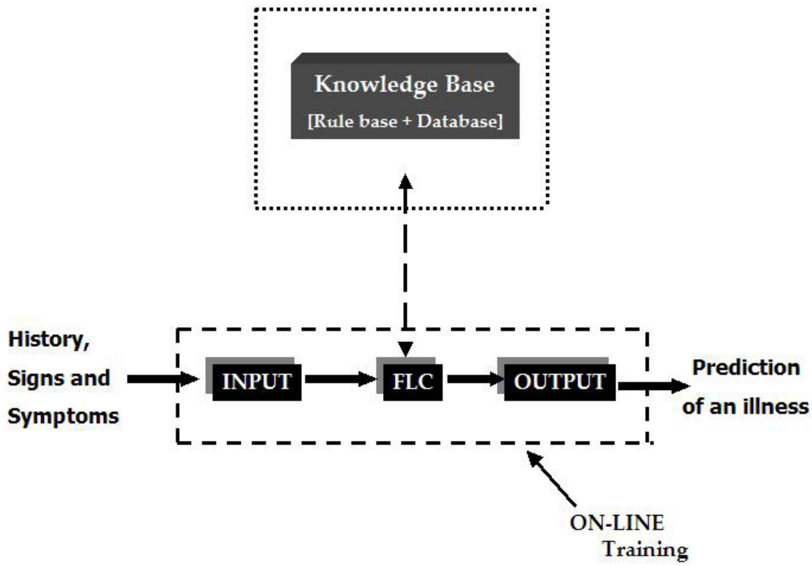
Kovacs and Juranovics [47] developed an expert system known as Auctoritus. The aim was to assist physicians for diagnosing diseases, storing and retrieval of administrative data, following-up psychiatric patients diagnosed by Diagnostic and Statistical Manual (DSM)-IV nosology or ICD-10 from distant locations, and taking therapeutic decisions. Auctoritus had four main parts: administration, diagnostic decision support system (most important part), activities concerning treatment, and statistics. The inference engine of the system was made up of fuzzy logic and backward chaining for validation: verification of the inferences and injecting precision to the inferences, respectively. Ohayon [48] proposed an IDSS called Sleep-Eval to study epidemiology of sleep disorders. Membership values were assigned against various sleep disorders based on the given set of inputs.

Decisions provided by any control process often need to be optimized for obtaining the best possible decisions and there are several optimizers available to develop a hybrid control process. Genetic algorithm is a powerful optimizer [49]. It is essentially a population-based search method that resembles Darwin's principle of natural selection. Chapman [50] used genetic algorithm as a search algorithm for classifying nine types of psychiatric illnesses including depression based on DSM-IV criteria (history, signs and symptoms). The author found that genetic algorithm-based search performed better that statistical discriminant analysis and logistic regressions.

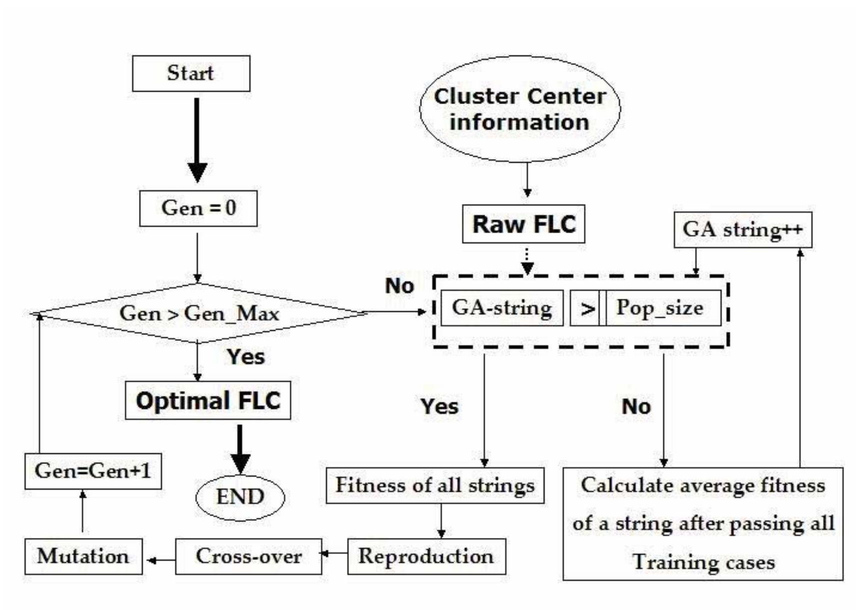
Though there were attempts to use individual soft computing technique for psychiatric disease diagnosis, combination of more than one technique has gained more popularity in medical diagnosis because of their inherent ability to complement each other's weaknesses.

Aruna et al. [51] proposed a neuro-fuzzy approach for diagnosing psychosomatic disorders. They defined fuzzy membership values based on the linguistic inputs (interview-based), which were fed, in turn, into a multi-layer feed- forward neural network. It was then tuned by back-propagation algorithm for refining the diagnosis. They found that the combined tool was a better classifier than the traditional models, e.g., Bayesian belief network, linear-discriminant analysis.

Chattopadhyay et al. [52] developed a combined genetic algorithm (GA)-fuzzy logic-based system for screening of seven adult psychotic disorders. The authors used fuzzy clustering-to-controller technique [24], where cluster center information was incorporated into the fuzzy logic controller as its knowledge base (refer to Fig. 9a). Two fuzzy clustering algorithms were used in this study, namely Entropy-based Fuzzy Clustering (EFC) [24] and Fuzzy-C-Means (FCM) [23] to develop two respective fuzzy logic controllers – FLC-I and II. The controllers were designed based on the principle of Takagi and Sugeno's approach [53]. Finally, the designed fuzzy logic controllers were optimized using a binary-coded GA (refer to Fig. 9b).



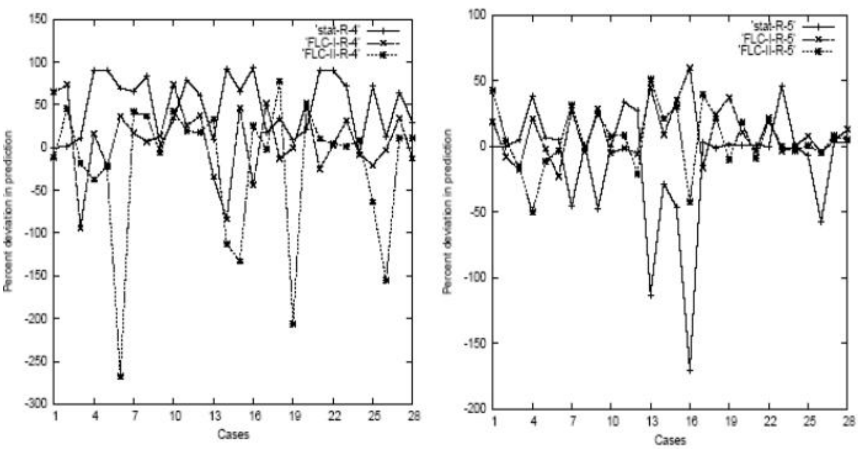
(a)



(b)

Fig. 9 Working principles of (a) fuzzy logic controller and (b) genetic-fuzzy system

Finally, the performances of the fuzzy logic controllers, developed based on FCM and EFC algorithms, were tested on two sets of data related to randomly chosen patients and real-world patients, whose diagnoses were known a-priori. FCM algorithm-based fuzzy logic controller was found to outperform the EFC-based controller. Percentage deviations in predictions were calculated as a measure of the performances of these controllers on two sets of patients: twenty-eight randomly generated patients (in consultation with the psychiatric colleagues) and forty-one real-world cases (collected from the psychiatric colleagues). Both the fuzzy logic controllers could show the better performance compared to the traditional statistical measures, such as multiple linear regression-based approach (refer to Fig. 10).



(a) (b)

Fig. 10 Comparisons of the performances of FLC-I and II with that of traditional statistical technique for (a) R-4 (Response 4: Delusional disorder) and (b) R-5 (Response 5: Schizoaffective disorder)

However, it is worth mentioning that developing a control system does not merely complete the task to bring it down to practice in the real world. Its successful implementation depends on various factors, such as social, political, economic, legal and many others. This chapter is, therefore, interested to talk about some of the issues relevant for successful implementations of a psychiatry information system.

4 Issues Behind Automating Psychiatric Decision-Making

In the previous sections, we have discussed the followings:

- Issues associated with medical data and hence decisions on screening, diagnosis, prognosis etc., especially in mental health (refer to Fig.1)

- Doctor's medical logic and modeling it using knowledge engineering techniques (refer to Fig. 4)
- Design and development of intelligent autonomous decision support tools by incorporating both extracted knowledge (by applying knowledge engineering techniques) and domain knowledge of human experts.
- Studies related to applications of knowledge engineering and intelligent autonomous decision support tools in psychiatry.

Undoubtedly, the strength, quality and versatility of mathematical techniques and tools are proven in today's world of advanced technology. The development in medical science partly bestows on technology applications in laboratory investigations, radiology, treatment etc. Doctors are now catering more efficient and smart healthcare using machines. However, as far as the subjectivity of the clinical data is concerned, there is almost no technology available. Probably, this is one of the reasons, why medical decisions cannot completely come out of human errors. Screening and diagnosis are two important steps towards a successful patient management and we know, if the diagnosis is wrong, every step after that goes in the wrong direction.

Given this scenario, this section explains two important issues that probably hinder applications of IDSS in real-world psychiatry practice, especially at a large scale. As a direction towards future research, this chapter also has stated some tentative solutions to these problems. In a nutshell, the current section proposes the possible ways to facilitate applications of IDSS as a complete information system in mental health.

4.1 Lack of Multidisciplinary Approach

Applications of KE in psychiatry seem to be scattered in nature. Seldom there are series of works leading to applications of extracted knowledge in the real-world scenario. Moreover, there are only a few reported studies that attempt to correlate basic researches, for example, correlation findings of biological researches (at the genetic and molecular levels) with the clinical research in mental health [13][16].

Similar picture can be obtained from the researches on clinical data mining, where effective propagation of the extracted knowledge to develop IDSS in psychiatry is less reported. One important reason could be the lack of multidisciplinary approach, e.g., doctors conduct clinical research, as they understand the signs, symptoms and history of patients better. On the other hand, technologists understand mathematical algorithms and a control system better than the doctors do, but do not understand clinical data as the doctors can. Hence, there is rarely or no interfacing of these expertise while developing a complete health information system. This is probably the greatest barrier towards developing a complete mental health information system in practice.

Medical informatics, therefore, has emerged as the demanding research field, by which we can bring multiple disciplines under one roof for effective interfacing.

This work proposes a collaborative knowledge management [54] involving both psychiatrists and technologists for developing such systems and bringing down the system for regular psychiatric practice. The key impetus to adopt collaborative knowledge management is to ensure a flexible Knowledge Management System (KMS) [54] throughout that facilitates high productivity and throughput of the developed IDSS in practical life.

In this regard, it is important to mention that due to the dynamic nature, KMS is able to adapt newer technologies with time to build a more integrated system. Moreover, a KMS is a living system and easy to adopt by the end-users [54]. Therefore, in future, attempts are to be made to incorporate KMS with IDSS to derive its business value through practical implementation.

4.2 Issues Related to Adoption of IDSS

Another significant problem behind the success of a health information system lies in the fact that the end-users sometimes are not ready to adopt it [55]. It leads to a loss of resources, such as man, technology, and time [56]. There are many reasons behind the failure of a system's adoption. Some of the relevant issues are as follows:

1. It is sometimes seen that the stakeholders (e.g., health care professionals) are not prepared to adopt a new system, especially in medical practice. A reason could be the tendency of the health care staff clinging to the conventional practice, in which they feel more comfortable. Given this issue, this chapter proposes that a preparedness study should be conducted by either an extensive survey or demonstrating the possible efficacies of the system to the stakeholders [57].
2. Technologists, naturally, may not understand the requirement of a health information system, as they are not the doctors. But, understanding the basic requirements of the end-users is the most important task towards a successful implementation [57]. It can only be achieved by a rigorous multidisciplinary approach.
3. It is often seen, especially in the developing countries, that the doctors are uncomfortable using computers in their practice [58, 59]. To implement an information system in psychiatry practice, appropriate training should be given to them for a successful implementation of the system and ensuring its long-term future.
4. The fourth issue is probably the cost factor that includes cost of human and technical resources for effectively running such a system. However, cost can be initially focused on basic patient care and a private-government partnership may be adopted while implementing such systems [60].
5. Among some of the remaining issues are poor documentation, accuracies within the patient data, and insufficient communications among the systems [61]. However, due to space constraint these issues are not elaborated in this chapter.

5 Concluding Remarks and Future Work

This chapter does not merely review the state of research on KE techniques and design and development of IDSS in psychiatry. The key question that has been raised is that, despite of proven efficacies why these technologies are still encaged in the research publications but not in the mainstream applications of day-to-day psychiatry practice. To answer this question, the present chapter has identified two important reasons, such as (i) lack of multidimensional approach while developing IDSS in psychiatry and (ii) poor adoption of the systems by the users. In conclusion, the present chapter proposes to design a complete Health Information System (HIS) incorporating both the technology as well as principles of collaborative knowledge management in psychiatry. The said system should be (i) adaptive to embrace newer technologies and (ii) able to incorporate technology, knowledge management concepts in psychiatry. This effort may reduce adoption problems among the healthcare stakeholders.

Use of a complete HIS of this type may screen the patients faster by virtue of robust data mining, automatic decision-making by an intelligent inference engine, and inserting precision and speed in diagnosis. In other words, such a system could be very useful to encourage people to be aware of psychiatric diseases, its mode of onset, symptoms, course, treatment strategies and so forth. This, in turn, may take care of various social issues, such as taboos, frustrations among caregivers, and economic loss, which may be the final ambition of a researcher working on mental health.

References

[1] Chattopadhyay, S.: Fuzzy logic-based Expert Systems for Screening and Prediction of Adult Psychoses. Ph.D. dissertation, School of Information Technology, Indian Institute of Technology, Kharagpur, India (2007)

[2] Han, J., Kamber, M.: Data Mining Concepts and Techniques. Morgan Kauffmann Publishers, San Francisco (2006)

[3] Sadock, B.J., Sadock, V.A.: Kaplan and Sadock's Synopsis of Psychiatry Behavioral Sciences/Clinical Psychiatry, 10th edn. Wolters Kluwer Lippincott Williams & Wilkins (2007)

[4] Beck, A.T., Schuyler, D., Herman, J.: Development of Suicidal Intent Scales. In: Beck, A.T., Resnick, H., Lettieri, D. (eds.) The Prediction of Suicide. The Guilford Press, NY (1974)

[5] Overall, J.E., Gorham, D.R.: The brief psychiatric rating scale. Psy. Rep. 10, 779–812 (1962)

[6] Pierce, D.W.: Suicidal Intent in Self-injuries. Br. J. Psychiatry 130, 377–385 (1977)

[7] Kay, S.R., Fiszbein, A., Opler, L.A.: The positive and negative syndrome scale (PANSS) for schizoprenia. Schizophr. Bull. 13(2), 261–276 (1987)

[8] Hamilton, M.: A rating scale for depression. J. Neuro., Neurosurg. Psychiatry 23, 56–62 (1960)

[9] Arion, D., Unger, T., Lewis, D.A., et al.: Molecular evidence for increased expression of genes related to immune and chaperone function in the prefrontal cortex in schizophrenia. Biol. Psychiatry 62, 711–721 (2007)
[10] Moreno, D., Moreno-Iñiguez, M., Vigil, D., et al.: Obstetric complications as a risk factor for first psychotic episodes in childhood and adolescence. Eur. Child Adolesc. Psychiatry PMID, 19184163 (2009)
[11] Le-Niculescu, H., Patel, S.D., Bhat, M., et al.: Convergent functional genomics of genome-wide association data for bipolar disorder: Comprehensive identification of candidate genes, pathways, and mechanisms. Am. J. Med. Genet. B Neuropsychiatr. Genet. PMID, 19025758 (2008)
[12] McEachin, R.C., Keller, B.J., Saunders, E.F., et al.: Modeling gene-by-environment interaction in comorbid depression with alcohol use disorders via an integrated bioinformatics approach. BioData Min. 1(1), 2 (2008)
[13] Kafkafi, N., Yekutieli, D., Yarowsky, P., et al.: Data mining in behavioral test detects early symptoms in a model of amyotrophic lateral sclerosis. Behav. Neurosci. 122(4), 777–787 (2008)
[14] Baca-Garcia, E., Perez-Rodriguez, M.M., Basurte-Villamor, I., et al.: Using data mining to explore complex clinical decision: A study of hospitalization after a suicide attempt. J. Clin. Psychiatry 67(7), 1124–1132 (2006)
[15] Baca-Garcia, E., Perez-Rodriguez, M.M., Saiz-Gonzalez, D., et al.: Variable associated with familial suicide attempts in a sample of suicide attempters. Prog. Neuropsychopharmacol. Biol. Psychiatry 31(6), 312–316 (2007)
[16] Marinic, I., Supek, F., Kovacic, Z., et al.: Posttraumatic stress disorder: diagnostic data analysis by data mining methodology. Croat. Med. J. 48(2), 185–197 (2007)
[17] Reulbach, U., Biermann, T., Bleich, S., et al.: Alcoholism and homicide with respect to the classification systems of Lesch and Clininger. Alcohol Alcohol 42(2), 102–107 (2007)
[18] Chattopadhyay, S.: A Study on Suicidal Risk Analysis. In: Proceedings of 9th IEEE Intl. Conf. on e-Health Networking, Applications and Service (HEALTHCOM 2007) Taipei, Taiwan, pp. 74–79 (2007)
[19] Chattopadhyay, S., Ray, P., Chen, H.S., Lee, M.B., Chiang, H.C.: Suicidal Risk Evaluation using a Similarity-based Classifier. In: Tang, C., Ling, C.X., Zhou, X., Cercone, N.J., Li, X. (eds.) ADMA 2008. LNCS (LNAI), vol. 5139, pp. 51–61. Springer, Heidelberg (2008)
[20] Nebekar, J.R., Yarnold, P.R., Soltysik, R.C., et al.: Developing indicators of inpatient adverse drug events through nonlinear analysis using administrative data. Med. Care 45(10), S81–S88 (2007)
[21] Chattopadhyay, S., Pratihar, D.K., De Sarkar, S.C.: Developing fuzzy classifiers for predicting the chance of occurrence of adult psychoses. Knowl.-Based Syst. 20, 479–497 (2008)
[22] Plackett, R.L., Burman, J.P.: The design of optimum multi-factorial experiments. Biometrika 33, 305–325 (1946)
[23] Chattopadhyay, S., Pratihar, D.K., De Sarkar, S.C.: Fuzzy clustering of psychosis data. Int. J. Business Intel. Data Min. 2(2), 143–159 (2007)
[24] Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell (1981)
[25] Yao, J., Dash, M., Tan, S.T., et al.: Entropy-based fuzzy clustering and fuzzy modeling. Fuzzy Sets Syst. 113, 381–388 (2000)

[26] Wing, J.K., Giddens, R.G.J.: Industrial rehabilitation of male chronic schizophrenic. Lancet 2, 505–507 (1959)
[27] Wing, J.K., Cooper, J.E.: Sarorius the description and classification of psychiatric symptoms: An instruction manual of PSE and CATEGO system. Cambridge University Press, London (1974)
[28] Keinmuntz, B.: The computers as clinicians. Am. Psych. 75, 1269–1274 (1967)
[29] Spitzer, R.L., Endicott, J.: DIAGNO-II: Further developments in a computer programme for psychiatric diagnosis. Am. J. Psychiatry 125, 12–21 (1969)
[30] Takahashi, T.: A mathematical framework for probabilistic choice based on information theory and psychophysics. Med. Hypotheses 67(1), 183–186 (2008)
[31] Heiser, J.F., Brookes, R.E.: Design consideration for a clinical psychopharmalogy advisor. In: Proceedings of second annual symposium of computer applications in medical care, pp. 78–85. IEEE, Los Alamitos (1978)
[32] Shortliffe, E.H.: Computer-based medical consultation MYCIN. Elsevier North Holland, NY (1976)
[33] Mulsant, B., Servan-Schreiber, D.: Knowledge engineering: a daily activity on a hospital ward. Comp. Biomed. Res. 17, 71–91 (1984)
[34] Horwitz, E.J., Heckermann, D.E., Nathwani, B.N.: Diagnostic strategies in the hypothesis directed PATHFINDER system. In: Proceedings of 1st conference on Artificial Intelligence Applications, pp. 30–36 (1984)
[35] Overby, M.A.: Psyxpert: an expert system prototype for aiding psychiatrists in the diagnosis of psychotic disorders. Comput. Biol. Med. 17(6), 383–393 (1987)
[36] Ohayon, M., Poinso, Y.: Compatbilite du DSM-III avec la nosographic psychiatric francaise. Psy. Med. 19, 367–370 (1987)
[37] Johri, S.K., Guha, S.K.: Set-covering diagnostic expert system for psychiatric diagnosis: the third world context. Comp. Meth. Prog. Biomed. 31, 1–7 (1994)
[38] Chattopadhyay, S., Ray, P., Lee, M.B., Chen, H.S.: Towards the Design of an e-Health System for Suicide Prevention. In: Proceedings of IASTED Conference on Artificial Intelligence and Soft Computing (ASC), pp. 191–196 (2007)
[39] Pratihar, D.K.: Soft Computing. Alpha Science International Ltd., Oxford (2008)
[40] Berhman, L.G., Fors, U.G.: Decision support in psychiatry – a comparison between the diagnostic outcomes using a computerized decision support system versus manual diagnosis. BMC Med. Inform. Decis. Mak. 8, 8–9 (2008)
[41] Lowell, W.E., Davis, G.E.: Predicting length of stay for psychiatric diagnosis groups using neural networks. J. Am. Info. Asso. 1(6), 459–466 (1994)
[42] Zou, Y., Shen, Y., Leu, L., et al.: Artificial neural network to assist psychiatric diagnosis. Br. J. Psychiatry 169, 64–67 (1996)
[43] Buscema, M., Mazzetti de Pietralata, M., Salvemini, V., et al.: Application of artificial neural network in eating disorders. Subs. Use Mis. 33, 765–791 (1998)
[44] Jefferson, M.F., Pendleton, N., Lucas, C.P., et al.: Evolution of artificial neural network architecture: prediction of depression after mania. Meth. Inform. Med. 37, 220–225 (1998)
[45] Yana, K., Kawachi, K., Iida, K., et al.: A neural set screening of psychiatric patients. In: Proceedings of 16th Annual international conference of the IEEE Engineering in Medicine and Biology Society, vol. 16, pp. 1366–1367 (1994)
[46] Zadeh, L.A.: Fuzzy Sets. Inform. Control 8, 338–353 (1965)
[47] Kovacs, M., Juranovich, J.: “Auctoritus” psychiatric expert system shell. Med. Info. 8(Pt 2), 997 (1995)

[48] Ohayon, M.M.: Improving decisionmaking processes with the fuzzy logic approach in the epidemiology of sleep disorders. J. Psychosom. Res. 47(4), 297–311 (1999)
[49] Holland, J.H.: Adaptation in natural and applied systems. Univ. Michigan Press, USA (1975)
[50] Chapman, C.N.: A genetic algorithm system to find symbolic rules for diagnosis of depression. In: Proceedings of the Congress of evolutionary computation, Washington, DC, USA, pp. 130–136 (1999)
[51] Aruna, P., Puviarasan, N., Palaniappan, B.: An investigation of neuraofuzzy systems in psychosomatic disorders. Exp. Syst. Appli. 28, 673–679 (2005)
[52] Chattopadhyay, S., Pratihar, D.K., De Sarkar, S.C.: Fuzzy Logic-based Screening and Prediction of Adult Psychoses: A Novel Approach. IEEE - Tran. Sys., Man and Cyber., Part A 39(2), 381–387 (2009)
[53] Takagi, T., Sugeno, M.: Derivation of fuzzy control rules from human operator's control action. In: Proceedings of IFAC symposium on fuzzy information, knowledge representation and decision analysis, pp. 55–60 (1983)
[54] Debowski, S.: Knowledge Management. John Wiley & Sons Australia Ltd., Qld Australia (2006)
[55] Wickramasinghe, N.S., Goldberg, S.: Facilitating superior chronic disease management through a knowledge-based systems development model. Int. J. Electron Healthcare 4(3), 299–310 (2008)
[56] Wickramasinghe, N.S., Fadlalla, A.M.A., Geisler, E., et al.: A framework for assessing e-health preparedness. Int. J. Electro Healthcare 1(3), 316–334 (2005)
[57] Chattopadhyay, S., Ray, P.L., Li, J.: A framework for assessing ICT preparedness for e-health implementation. In: Proceedings of 10th IEEE Intl. Conf. on e-Health Networking, Applications and Service (HEALTHCOM 2008), pp. 147–152 (2008)
[58] Briggs, B.: Doctors sound off on IT concerns. Health Data Manag. 9(1), 38–40 (2001)
[59] Carroll, J.: Doctors say getting wired too costly, time-consuming. Manag. Care. 17(10), 5–6 (2008)
[60] Blank, M., Bauer, A.: Issues to consider before investing in bedside terminal systems. Hosp. Technol. Ser. 10(30), 1–17 (1991)
[61] AHIMA: e-HIM Workgroup on assessing and improving healthcare data quality in the EHR. Assessing and improving EHR data quality. J. AHIMA 78(3), 69–72 (2006)

8

Condition Monitoring of Internal Combustion Engine Using EMD and HMM

Sandeep Kumar Yadav and Prem Kumar Kalra

Abstract. The acoustic signature of an internal combustion (IC) engine contains valuable information regarding the functioning of its components. It could be used to detect the incipient faults in the engine. Acoustics-based condition monitoring of systems precisely tries to handle the questions and in the process extracts the relevant information from the acoustic signal to identify the health of the system. In automobile industry, fault diagnosis of engines is generally done by a set of skilled workers who by merely listening to the sound produced by the engine, certify whether the engine is good or bad, primary owing to their excellent sensory skills and cognitive capabilities. It would indeed be a challenging task to mimic the capabilities of those individuals in a machine. In the fault diagnosis setup developed hereby, the acoustic signal emanated from the engine is first captured and recorded; subsequently the acoustic signal is transformed on to a domain where distinct patterns corresponding to the faults being investigated are visible. Traditionally, acoustic signals are mainly analyzed with spectral analysis, i.e., the Fourier transform, which is not a proper tool for the analysis of IC engine acoustic signals, as they are non-stationary and consist of many transient components. In the present work, Empirical Mode Decomposition (EMD) and Hidden Markov Model (HMM)- based approach for IC engine is proposed. EMD is a new time-frequency analyzing method for nonlinear and non-stationary signals. By using the EMD, a complicated signal can be decomposed into a number of intrinsic mode functions (IMFs) based on the local characteristics time scale of the signal. Treating these IMFs as feature vectors HMM is applied to classify the IC engine acoustic signal. Experimental results show that the proposed method can be used as a tool in *intelligent autonomous system* for condition monitoring and fault diagnosis of IC engine.

1 Introduction

Condition monitoring, fault diagnosis and quality assurance have become of paramount importance in various industrial sectors, such as automobile industries, aeronautics industries, power generating units. Interestingly, all these sectors can

Sandeep Kumar Yadav · Prem Kumar Kalra
Department of Electrical Engineering, IIT Kanpur, India, 208016
e-mail: sandeepy@iitk.ac.in, kalra@iitk.ac.in

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 167–185.
springerlink.com

use generic solution approach to achieve greater autonomy with quality assurance utilizing non-parametric classification and prediction methodologies.

Reliable operation of any *intelligent autonomous* system depends on continuous evaluation of its performance trend, detection of any deterioration in trends, establishment of the cause of deterioration, initiation of the actions to arrest its propagation and prediction of the future trend. The process of detecting deterioration is generally known as **condition monitoring**. Identifying the cause and suggesting the remedial measures is termed as **diagnosis**, and predicting what the trend would be in future is called **prognosis**. Modern electronic gadgets connected to a sensor web enable one to measure parameters of any system, thereby providing inputs which can be used to take appropriate decisions. Central to the whole setup is the signal which is generated by the system and means-cum-methods to capture and process the signal. While this is true, a fundamental question which we should address is whether the decision making process adopted should be a deterministic approach or a non-deterministic one. Decision making in deterministic settings would require accurate measurement of important parameters of system, while the case of non-deterministic settings would generally require monitoring the current trend and correlating with the past values.

Three key elements of effective fault diagnosis system of internal combustion engine are **data acquisition, data processing, and decision making**. Increased automaton and mechanization have made computerized diagnosis system a valuable tool for early detection of fault. Today's concept of internal combustion engine fault diagnosis comprises of the automated detection and classification of faults.

Vibration and acoustic emission signals are often used for fault diagnosis in mechanical systems, since they often carry dynamic information for mechanical elements. A number of papers had been published on this topic [1-6]. Continuous wavelet transform represents the time-frequency distribution properties of the acoustical signal and extracts the time-localized fault features [7]. However, the results of wavelet transform highly depend on the choice of the wavelet basis function to have a chance to lead to coefficients of high value, while all other features will be masked or even completely ignored [8].

Recently, a new time frequency analysis method, designated as Empirical Mode Decomposition (EMD), has been proposed by Huang and his colleagues [9]. Using the EMD, any complicated signal can be decomposed into a collection of Intrinsic Mode Functions (IMFs) based on the local characteristics time scale of the signal. The IMF represents the natural oscillatory mode embedded in the signal. EMD is self-adaptive because the IMFs, working as the basis function, are determined by the signal itself rather than what is pre-determined. Therefore, EMD is highly efficient in non-stationary signal analysis. Some applications of EMD in mechanical fault diagnosis have been studied, for example, in structural health monitoring [10-11].

Classification of internal combustion engines acoustic signal is essentially a pattern recognition problem, in which there are two basic issues: feature selection, and classification based on the selected features. For the first issue, EMD is applied to obtain IMFs, and for the second, the useful IMFs are selected based on correlation coefficients and Hidden Morkov Model (HMM) is used as a tool for classification.

Early application of HMM in fault classification and diagnostics treated the real machine's faulty state and its normal state as the hidden states of the HMM [12-13]. Two recent applications of HMM in fault classification have assumed an HMM with hidden states having no physical meaning for each machine condition [14-15]. The trained HMM is then used to decode an observation with unknown machine condition for fault classification. Xu et al. [16] presented an intelligent fault diagnosis system based on an HMM.

The scope of this chapter is to identify whether an engine is good or defective and if it is found to be defective, we are required to identify the faults. We have focused our study on Tappet fault, Cam Chain fault, Primary Gear damage fault and Cylinder Head fault. In section 2, working principle of Internal Combustion (IC) engine and its noise sources are briefly explained. EMD technique for feature extraction is discussed in section 3. Section 4 deals with the fundamentals of HMM. In section 5, proposed EMD-HMM technique is discussed. The proposed technique is investigated briefly for IC engine fault diagnosis in section 6. The last section concludes the chapter.

2 Working Principle of IC Engine and Noise Sources

In IC engines, combustion takes place in a confined space, producing expanding gases that are used directly to provide mechanical power. Such engines are generally classified as reciprocating or rotary, spark ignition or compressor ignition, and two-stroke or four-stroke [17]. Reciprocating, spark ignited, four stroke gasoline engines are the ones, which are the most commonly used. The present chapter deals with analysis of single-cylinder four stroke IC engine.

An IC engine's noise signal is composed of many components generated from various sources. These sources include combustion, mechanical and the combination of both [18]. To identify the requirement of noise signal analysis, we will discuss the noise signal components. The combustion noise is produced by a rapid rise of pressure, which besides being a source of engine structural vibration also excites resonance in gas inside the combustion chamber cavity. The later is also a source of vibration and noise. The contribution of combustion to the whole noise signal is some transient components. In a normal condition, the combustion noise is mostly above 100 Hertz, as the combustion energy below this range is mostly transformed into useful work by pushing the piston forward. In the case of abnormal conditions, degradation in the combustion quality may produce low frequency signal in the combustion noise.

A rise in the cylinder pressure pushes the piston from the top dead center (TDC) to the bottom dead center (BDC). In this movement, the clearance between the piston and the cylinder or damage to piston rings can cause the piston slap, which is a major source of engine noise. As the piston slap is caused by both the combustion and clearance, the impacts will add the transient components to the engine noise signal.

One important feature of IC engine is that it has both reciprocating and rotating parts. Rotating parts, such as the flywheel and front pulley, can add harmonic components to the noise. An increase in the amplitude of the harmonic component indicates condition variations of these rotating parts. Contributions of various rotating parts to the noise can be identified with reference to their speeds. The timing gear chain is another source of mechanical noise. Any damage to the gears will produce impacts and thus, adds extra transient components.

An engine has many inlet and exhaust valves. A valve is opened by a Cam shaft and pushed back to its seat by a spring. Any problem with valve seats, tappets and mechanisms can cause a change to the transient vibrations produced during opening and closing, and thus, the corresponding transient components of the noise signal. These valves open and close at different times, and so, the contribution of different valves to the noise can be identified from the times of events.

Fluid-induced noise, such as exhaust and inlet noise, is also an important part of the noise. Along with the sudden release of gas into the exhaust system or the rush of a sharp pulse of fresh air into the cylinder, oscillation of the air volume in the cylinder and the exhaust system is excited and hence, noise is produced. When inlet and exhaust valve close, noise will also be generated for a change in the fluid field. The fluid induced noise is normally very low. Damage or problem with the exhaust and inlet system will increase the magnitude of the fluid-induced noise.

3 Empirical Mode Decomposition

The method of EMD is necessary to deal with signal from non-stationary and nonlinear system [19]. The method is intuitive, direct and adaptive, with posterior defined basis, from the decomposition method, derived from the signal. The decomposition is based on the simple assumption that any signal is composed of different simple intrinsic modes of oscillations. Each intrinsic mode, linear or nonlinear, represents a simple oscillation, which will have the same number of extrema and zero crossing. Furthermore, the oscillation will also be symmetric with respect to *local mean*. At any given time, the signal may have many different coexisting methods of oscillation, one superimposing on the others. Each of these oscillatory modes is represented by an IMF with the following definition:

1. In the whole signal, the number of extrema and number of zero-crossing must either be equal to each other or may differ at most by one;
2. At any point, the mean value of the envelope defined by the local maxima and that defined by the local minima is zero.

An IMF represents a simple oscillatory mode as a counterpart to the simple harmonic function, but it is much more general: instead of constant amplitude and frequency, as in a simple harmonic component, the IMF can have a variable amplitude and frequency as the function of time.

With the above definition of the IMF, we can decompose any signal $x(t)$ based on these three assumptions:

1. There are at least two extrema in the signal, one maximum and one minimum;
2. The characteristic time scale is defined by the time lapse between the extrema;
3. If the signal is totally devoid of extrema but contain only inflection points, then it can be differentiated once or more times to reveal the extrema.

Once the extrema are identified, the maxima are connected using the cubic spline and used as the upper envelope. The minima are interpolated as well to form the lower envelope. The upper and lower envelopes should cover all the data in the time series. The mean of the upper and lower envelope, that is $m_1(t)$ is subtracted from the original signal to obtain the first component $h_1(t)$ of the shifting process as given below.

$$h_1(t) = x(t) - m_1(t) \tag{1}$$

Ideally, if $h_1(t)$ is an IMF, the shifting process will stop. In reality, however, overshoots and undershoots are common, which also generate the new extrema. It means that the process will continue until the result approximates the IMF, $h_{1k}(t)$. In other words, it will shift the signal again in the same way to get another component $h_{11}(t)$.

$$h_{11}(t) = h_{1k}(t) - m_{11}(t), \tag{2}$$

where $m_{11}(t)$ is the mean of the upper and lower envelopes of $h_{11}(t)$. The $h_{1k}(t)$ is then designated as the first component $c_{1k}(t) = h_{1k}(t)$.

In order to cause the shifting process, a criterion is introduced to stop it, which retains enough physical sense of both amplitude and frequency modulations. This can be accomplished by limiting the size of the standard deviation (S.D.) calculated from the two consecutive shifting as in [5].

$$S.D. = \sum_{t=0}^{t=T} \frac{\left|h_{1(k-1)}(t) - h_{1k}(t)\right|^2}{h_{1(k-1)}^2(t)}, \tag{3}$$

where T is the length of the signal in time domain, and the threshold value is usually set between 0.3 and 0.4.

Overall, $c_1(t)$ should contain either the finest scale or the highest frequency information of the signal at each time point. The residual after the first shifting process can be determined as follows:

$$r_1(t) = x(t) - c_1(t). \tag{4}$$

Since the residue $r_1(t)$ still contains the lower frequency components, it is considered as a new signal and subjected to the same shifting process as described above. The procedure can be repeated on all the subsequent residuals to generate all the IMFs. The shifting process should stop according to the requirement of the physical process. However, there are some general requirements. For example, the sum-square of the residuals is less than a predefined threshold value or the residual becomes a monotonic function or a function with only one maximum and one minimum, from which no more IMFs can be created. The residual shifted out for *n* components is calculated as follows:

$$r_n(t) = x(t) - \sum_{i=1}^{i=n} c_i(t)\,, \tag{5}$$

where $r_n(t)$ is the residue of the signal and $c_i(t)$ stands for the $i-th$ IMF.

4 Basic Fundamentals of HMM

HMM is an extension of Markov chains. Considering the fact that the actual problem of IC engine fault diagnosis is more complex than that can be described by Markov chains, in an HMM, each state does not correspond to an observable event, but is connected to a group of probability distributions of the state. In some applications, the states may have certain physical meaning attached to the states or the sets of states [20-21]. An HMM is characterized by the following:

1. *N* represents the number of states in the model. The individual states are denoted by $S = (S_1, S_2, \ldots, S_N)$, and the state at time *t* is indicated by q_t and obviously $q_t \in S$.
2. *M* denotes the number of distinct observation symbols per state (i.e., discrete observations). We represent the observation symbols as $V = (V_1, V_2, \ldots, V_M)$, and the observable symbols at time $t \in V$.
3. $\pi = \pi_i$, the initial state probability distribution, where $\pi_i = P(q_1 = S_i), 1 \le i \le N$.
4. $A = a_{ij}$, the state transition probability distribution matrix, where $a_{ij} = P(q_{t+1} = S_j / q_t = S_i), 1 \le i, j \le N$.
5. The observation symbol probability distribution in state j, $B = b_j(k)$, where $b_j(k) = P(v_k(t) / q_t = S_j),\ 1 \le j \le N, 1 \le k \le M$.

So, HMM may be denoted as $\lambda(N, M, \pi, A, B)$. Given appropriate values of N, M, A, B and π, the HMM can be used as a generator to give an observation sequence as follows:

$$O = O_1, O_2, \ldots, O_T; O_T \in V \tag{6}$$

1. Choose the initial state $q_1 = S_i$, based on the initial state distribution π.
2. For $t = 1$ to T
   - Choose $O_t = v_k$ according to the symbol probability distribution in state S_i, $b_i(k)$.
   - Transition to a new state $q_{t+1} = S_j$ according to the state transition probability S_i, a_{ij}.
3. Increment t by 1, return to step 2, if $t \le T$; else terminate the procedure.

We can use this procedure for generating the observation as well as model for how a given observation sequence is generated by an appropriate HMM. From the above discussion, the complete specification of an HMM requires the information of two model parameters M and N, specification of observation symbols, and that of three probability measures A, B, π. For the ease of representation, we use the compact notation $\lambda(N, M, \pi, A, B)$.

There are three basic algorithms in HMM, namely Forward-backward algorithm, Viterbi algorithm and Baum-Welch algorithm [20-21]. These three algorithms can solve the three basic problems as mentioned below.

1. **Scoring:** Given an observation sequence $O = O_1, O_2, \ldots, O_T$, and a model $\lambda = A, B, \pi$; we can compute $P(O / \lambda)$, the probability of observation sequence. This is also called the Forward-backward algorithm.
2. **Matching:** Given an observation sequence $O = O_1, O_2, \ldots, O_T$, we can compute a state sequence $Q = q_1, q_2, \ldots, q_T$; which is optimal in meaningful sense. This is known as the Viterbi algorithm.
3. **Training:** We can adjust the model parameters $\lambda = A, B, \pi$ to maximize the probability of observation sequence $P(O / \lambda)$ for a given model.

5 Proposed Method

In the proposed method, we decompose the signal at first using EMD to obtain IMFs. The real intrinsic mode functions will have good correlation with the original signal [22], but the pseudo-component will have poor correlation with the signal. The correlation coefficients between the analyzed signal and the IMFs are used to eliminate the undesired IMFs. Therefore, the correlation coefficient between the IMFs and the signal are considered as a criterion to check the validity of the IMFs. The IMFs with significant information will be kept, and the remaining one will be added to the residue. The correlation coefficient μ_i ($i = 1, 2, 3, \ldots, n$; n is the number of IMFs) will be calculated, which have the maximum value that is not more than 1. After computing the correlation coefficients, a threshold λ will be chosen to examine the IMFs, which can be defined as the ratio of the absolute value of maximal correlation coefficient and a factor δ, as follows:

$$\lambda = \frac{\max(\mu_i)}{\delta}, \tag{7}$$

where the factor δ is kept more than 1. For our analysis, we use $\delta = 2.5$. Depending upon various industrial conditional requirements, different thresholds will be selected.

After the selection of useful IMFs according to above mentioned correlation-based procedure, we use these IMFs as a feature vector for training of the HMM, and the same procedure is used for the testing of signals.

6 IC Engine Fault Diagnosis Using Proposed Method

The following steps have been adopted for the IC engine fault diagnosis as discussed below.

6.1 Experimental Setup

To evaluate the proposed EMD-HMM technique for fault diagnosis, a single-cylinder IC engine testing rig has been used to measure the acoustic emission signals from it. Conventional engine fault diagnosis is carried out by a skilled worker by listening to the sound produced by the engine in order to make sure of the first analysis. Unfortunately, the conventional technique is not found to be a precise approach for fault diagnosis. In the present study, the acoustic signals will be extracted for the fault diagnosis.

The experimental set-up consists of a single-cylinder IC engine, an optical encoder for engine speed measurement, four ICP microphones (PCB 130D20), and data acquisition module (NI cDAQ-9172). Figure 1 shows the signal flow diagram of the engine fault diagnostic system. To test and verify the proposed method, four different faults of the engine have been considered as discussed below.

1. **Cam Chain Fault:** Cam chain is an element within the engine, which transfers the drive from Crank shaft to Cam shaft. The Cam chain rides along two riders. The tension on the Cam chain is adjusted by pushing the slider inwards or outwards. This can be done using the tension adjuster, which is accessible. Whenever the Cam chain is under tension, it produces the Cam chain noise. All Cam chain noise faults are seeded by varying the tension on the Cam chain using the tension adjuster.
2. **Tappet Fault:** Tappet noise appears, whenever the tappet clearance is too high. Under ideal settings, the inlet tappet and outlet tappet clearances are kept close to 0.07-0.08 mm. Any deviation from these set values results in generation of tappet noise. To seed tappet fault, the tappet clearance of both inlet and outlet

ports are disturbed from their ideal settings. The tappet clearances of both inlet and outlet ports are set and audio signals are recorded.

3. **Primary Gear Damage:** The gear assembly is located with the crank case. It comprises of a set of drive gear and driven gear assemblies. Any abnormality in these gear assemblies in the form of tooth damage, tooth to tooth error, eccentric/inclined bore, results in typical impact kind of noises. The primary gear damage fault is seeded by individually introducing defects in the driving and driven gears.
4. **Cylinder Head Fault:** Any noise emanating from the cylinder head, which is not produced by the tappet clearance is termed as cylinder head noise. The component other than the valve and rocker arm assembly within the Cylinder head to be considered is Cam shaft. Any defect on the Cam shaft tends to produce the Cylinder head noise. The Cylinder head fault is seeded by deforming/machining the surface of the Cam shaft. In view of recording the signals for these seeded faults, four microphones are placed at different locations of the engine as shown in Figure 2.

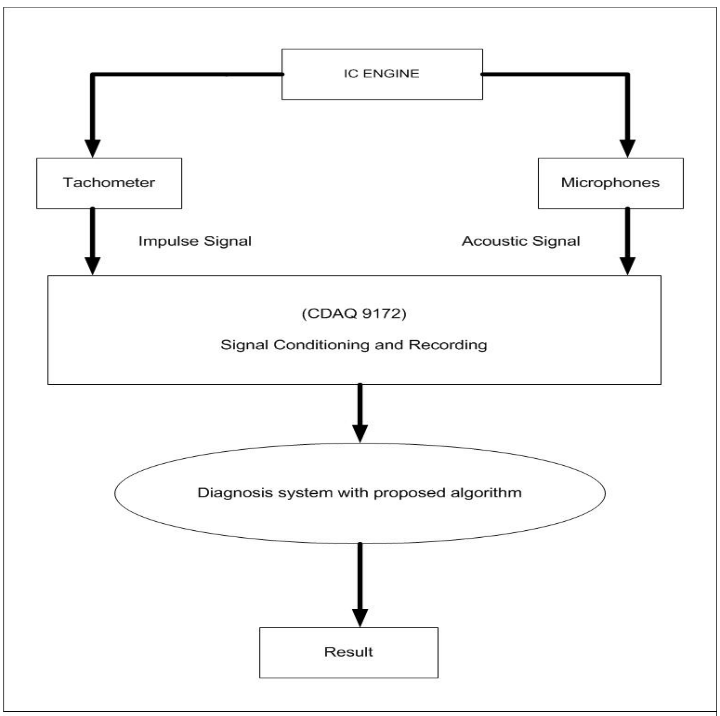


Fig. 1 Signal flow graph representation of engine fault diagnostic system

LabVIEW 8.0 has been used as a platform to communicate with the cDAQ-9172 hardware. Using VI's in LabVIEW, an easy to use menu driven user interface is utilized to record the engine acoustic samples. The engine is operated in the

2500 rpm speed. The sampling rate of the data acquisition system is 50 kHz. Along with this, the time instants of the main events, such as engine start time, engine gears check time, engine sound check time of the test cycle are also recorded.

The motivation behind recording the time instants of the main events of every test cycle is to ensure that measurements are taken to extract signals pertaining to each regime of engine test cycle. This is not alone, to ensure proper analysis of signal collected for all engines are analyzed under identical condition of operation. By following this approach, we are able to collect a total of 600 samples of engines along with normal engines.

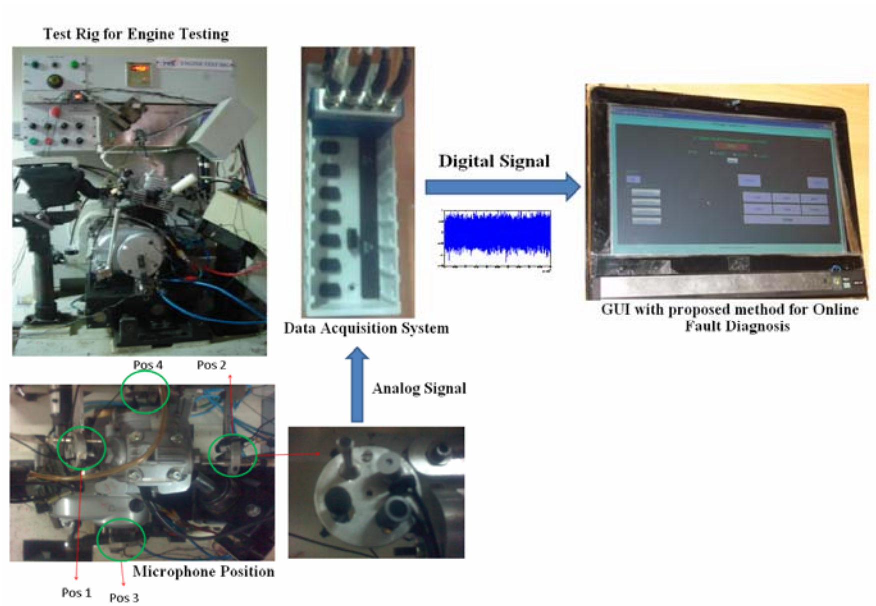


Fig. 2 IC engine fault diagnostic system

6.2 Feature Extraction

After signal recording of above mentioned four faults and the normal engine (without fault), we extracted the features (IMFs) of each individual fault (that is, audio-signal) at each microphone location using the EMD. The audio-signals and their IMFs for normal engine and four faults are shown in Figures 3 through 7. Obviously, it is not easy to detect faults from these diagrams. For selection of the most effective features (IMFs), we adopted correlation-based method. IMFs, which have the higher degree of correlation (correlation coefficient > 0.4) with the signal, are selected as the input for classification using the HMM. The correlation coefficients of IMFs with their corresponding acoustic signal are shown in Tables 1 through 5.

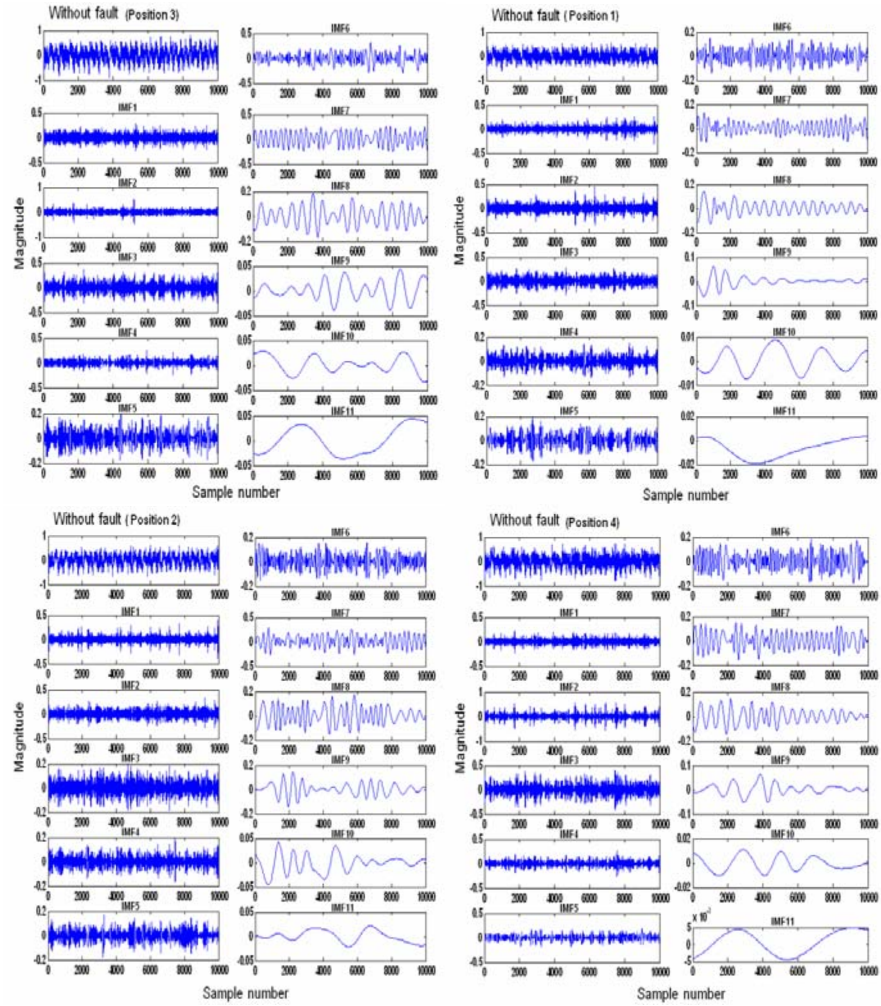


Fig. 3 Signal and its IMFs for normal engine

Table 1 Correlation coefficients of the IMFs and normal engine sample

IMFs	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7	IMF8	IMF9	IMF10	IMF11
Pos.1	0.4200	0.5483	0.5873	0.3496	0.4136	0.4709	0.4155	0.3621	0.1210	0.1054	0.0184
Pos.2	0.4242	0.4487	0.3851	0.3161	0.3711	0.4425	0.6587	0.4243	0.2320	0.0186	0.0142
Pos.3	0.3692	0.4182	0.4892	0.3208	0.3846	0.5062	0.6258	0.2992	0.1271	0.0121	0.0101
Pos.4	0.3385	0.6278	0.6011	0.3398	0.4125	0.4306	0.3966	0.3345	0.1368	0.0168	0.0159

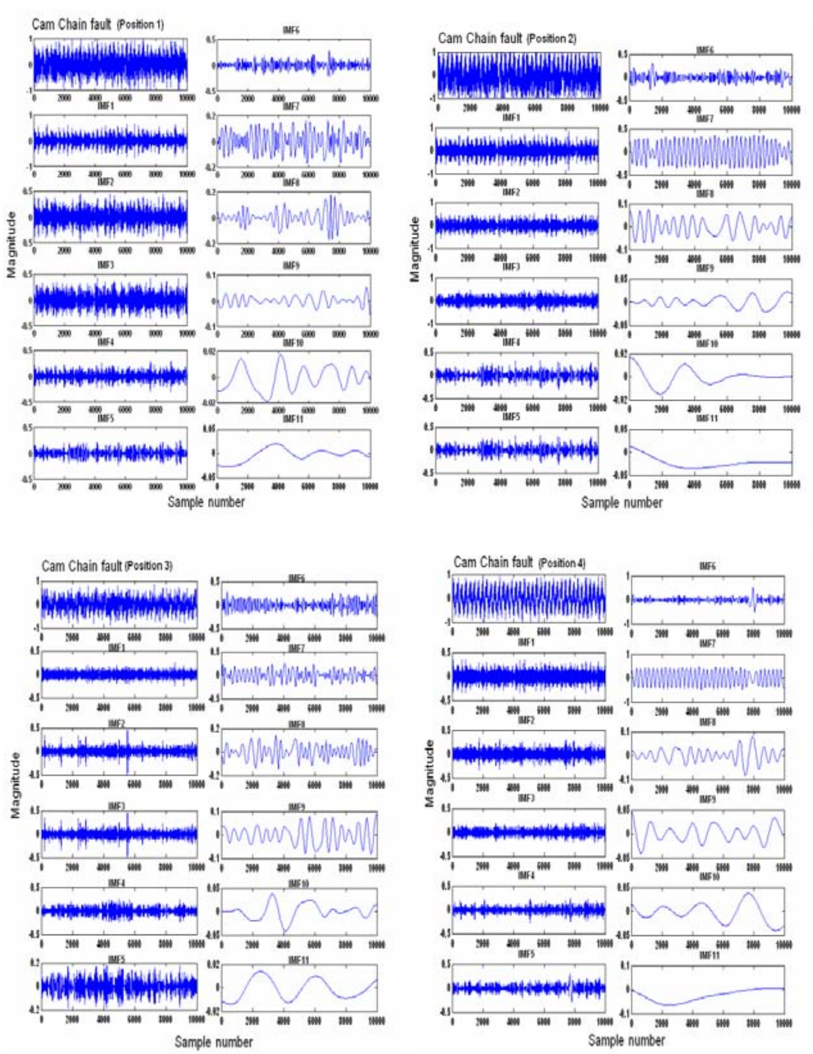


Fig. 4 Signal and its IMFs for Cam chain fault

Table 2 Correlation coefficients of the IMFs and Cam chain fault engine sample

IMFs	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7	IMF8	IMF9	IMF10	IMF11
Pos.1	0.6664	0.5454	0.5293	0.3542	0.4542	0.3170	0.3044	0.2325	0.1746	0.1247	0.0477
Pos.2	0.6134	0.5813	0.5229	0.3862	0.5099	0.3545	0.5571	0.0571	0.03331	0.0231	0.0399
Pos.3	0.5529	0.5292	0.5330	0.4271	0.3590	0.3611	0.5042	0.3880	0.1222	0.0127	0.0504
Pos.4	0.5850	0.5284	0.4831	1.4882	0.5687	0.5533	0.7023	0.1365	0.1187	0.1119	0.0032

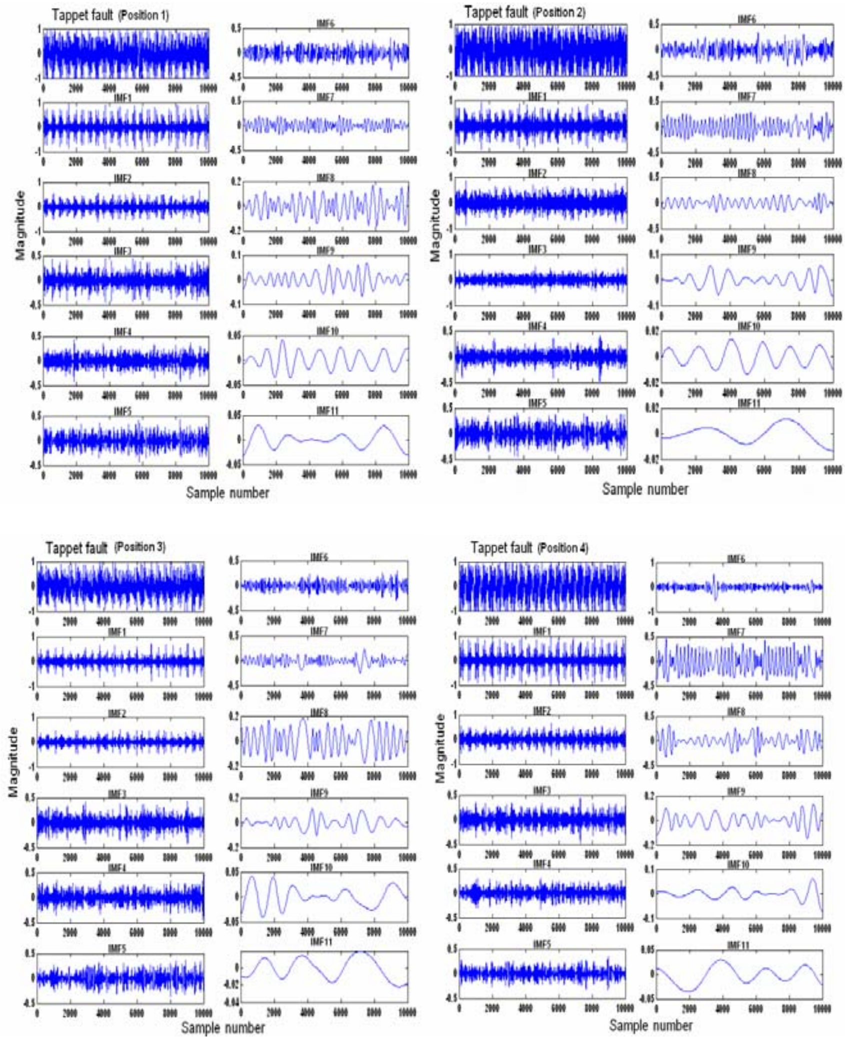


Fig. 5 Signal and its IMFs for Tappet fault engine

Table 3 Correlation coefficients of the IMFs and Tappet fault engine sample

IMFs	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7	IMF8	IMF9	IMF10	IMF11
Pos.1	0.6470	0.4896	0.4027	0.4577	0.5367	0.4103	0.2959	0.2144	0.1485	0.0148	0.0019
Pos.2	0.6102	0.5231	0.4178	0.3899	0.4490	0.3959	0.4738	0.2204	0.1153	0.050	0.0073
Pos.3	0.5019	0.4686	0.5224	0.5147	0.5430	0.3668	0.2711	0.1960	0.1284	0.0014	0.0011
Pos.4	0.5870	0.3840	0.3854	0.3612	0.4569	0.4641	0.6309	0.2223	0.0148	0.0134	0.0024

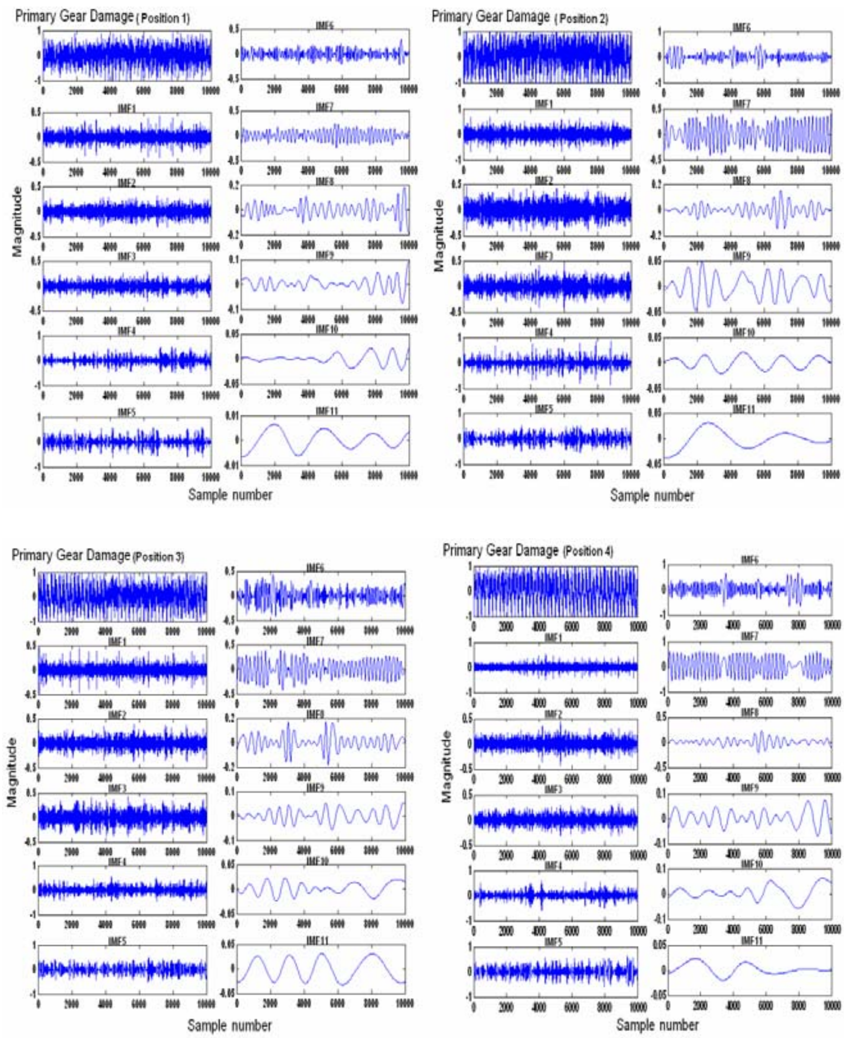


Fig. 6 Signal and its IMFs for PGD fault engine

Table 4 Correlation coefficients of the IMFs and PGD fault engine sample

IMFs	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7	IMF8	IMF9	IMF10	IMF11
Pos.1	0.4204	0.3236	0.4816	0.4664	0.4842	0.3549	0.2459	0.2362	0.0763	0.0234	0.0132
Pos.2	0.5090	0.4886	0.4683	0.4575	0.3129	0.4237	0.7014	0.2371	0.0336	0.0189	0.0694
Pos.3	0.3234	0.3064	0.3367	0.4101	0.4756	0.4824	0.4998	0.5015	0.0738	0.1930	0.1091
Pos.4	0.3303	0.2317	0.3545	0.4523	0.4546	0.4623	0.5538	0.6792	0.4310	0.0265	0.0142

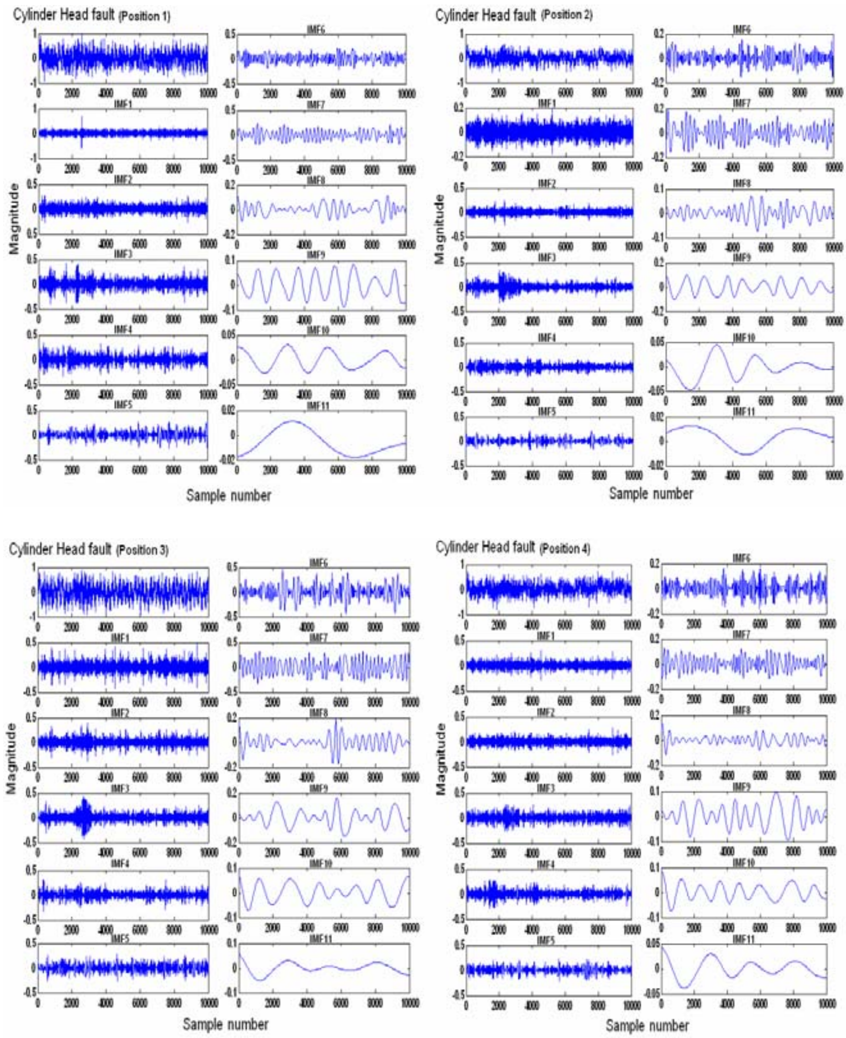


Fig. 7 Signal and its IMFs for CHN fault engine

Table 5 Correlation coefficients of the IMFs and CHN fault engine sample

IMFs	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7	IMF8	IMF9	IMF10	IMF11
Pos.1	0.4084	0.3986	0.5206	0.5211	0.4725	0.4834	0.4944	0.2986	0.2742	0.1746	0.0086
Pos.2	0.3799	0.3940	0.5289	0.4858	0.4363	0.4053	0.4977	0.2727	0.3824	0.1806	0.1074
Pos.3	0.3018	0.3499	0.4002	0.3522	0.4266	0.5748	0.6189	0.1954	0.2514	0.2583	0.0968
Pos.4	0.3992	0.4222	0.5062	0.5313	0.4790	0.4130	0.3670	0.2634	0.2859	0.2388	0.1920

6.3 Experimental Result and Classification

To carry out fault classification using the HMM, two parameters must be set: such as number of states and probability distribution of the observation. As mentioned earlier, the states may or may not have physical meaning, and they may affect accuracy of fault detection. Experiments have been carried out with a number of cases starting from $N = 2$ up to $N = 8$. As per the classification ratio shown in Figure 8 for different states of HMM, we found that $N = 3$ has yielded the better performance. The HMM training process flow has been shown in Figure 9. In the next phase, the test/new signal is evaluated against the trained models. The process flow for evaluating the test signal is also shown in Figure 9.

The objective of faults classification is to demonstrate the effectiveness of the proposed feature selection of IMFs. For this purpose, feature vectors in each fault condition are summarized as inputs to Hidden Markov Model as shown in Table 6. There are 120 data sets for each fault condition along with normal engine. IMFs of 20 engine signal set for each fault condition are used for the training of HMM and the remaining 100 engine signals are utilized to test the proposed fault diagnosis technique. The performance of the fault diagnostic system is evaluated using the classification rate (CR), which is defined as

$$\text{Classification Rate} = \frac{\text{Samples correctly classified}}{\text{Total testing samples}} \times 100\% . \tag{8}$$

As summarized in Table 6, all of the evaluation results have a performance over 98% classification rates in four different fault conditions. All the faults are seen to have the better recognition performance for acoustic signal near the source of origin of the fault. For overall classification, we have chosen a majority voting scheme, which means that eventual fault reporting is decided after a total fault reporting from all the positions, and the results obtained are more than 98%. The

Table 6 Performance of the proposed fault diagnostic technique

Fault category	Position 1		Position 2		Position 3		Position 4		Overall CR %
	IMFs	CR%	IMFs	CR%	IMFs	CR%	IMFs	CR%	
Normal Engine	1,2,3,5,6,7	98	1,2,6,7,8	97.5	2,3,6,7	96.57	2,3,5,6	96.25	99.33
Cam chain fault	1,2,3,5	97	1,2,3,5,7	96.33	1,2,3,4,7	95.25	1,2,3,4,5,6,7	98.33	98.67
Tappet Fault	1,2,3,4,5,6	98.7	1,2,3,5,7	97.89	1,2,3,4,5	96.89	1,6,7	94.56	98.89
PGD fault	1,3,4,5	98.39	1,2,3,4,6,7	94.23	4,5,6,7,8	98.33	4,5,6,7,8,9	94.86	98.56
CH fault	1,3,4,5,6,7	97.8	3,4,5,6,7	95.45	3,5,6,7	94.25	2,3,4,5,6	96.89	98.32

experimental results show that EMD-HMM fault diagnosis technique can be effectively used in diagnosis of various faults of IC engine through the measurement of acoustic emission signal.

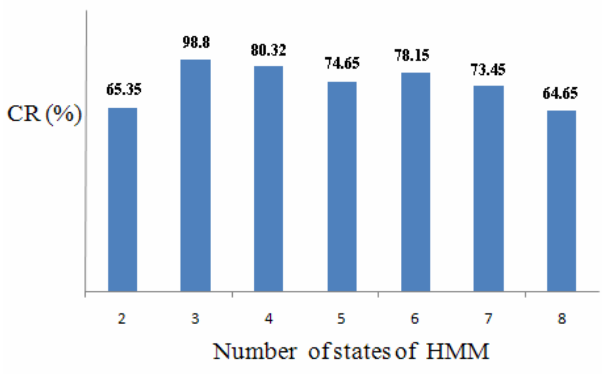


Fig. 8 Classification ratio vs. states of HMM

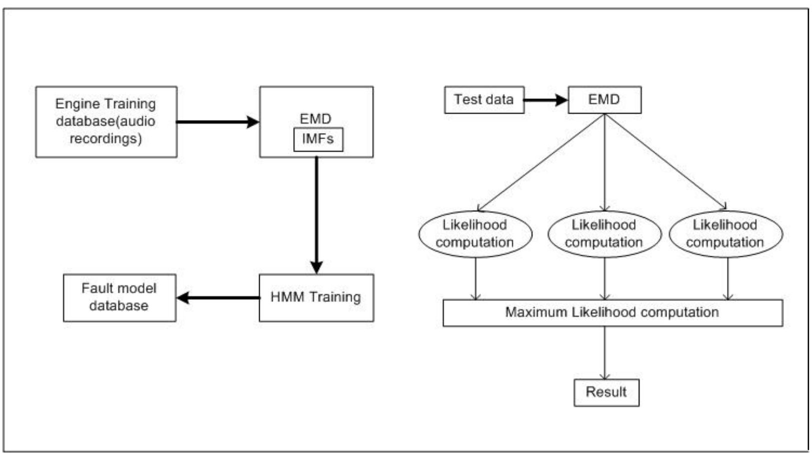


Fig. 9 HMM training and testing

7 Conclusions

Condition monitoring and fault diagnosis technique are used to detect and prevent early fault in a mechanical system. In this chapter, EMD-HMM method has been proposed for detecting and classifying the IC engine in various fault conditions. The work has begun with developing a data acquisition system for capturing and recording the acoustic signatures of the engine being tested. Once the signals are collected, they are decomposed into different IMFs using EMD. By calculating the correlation coefficient between the signal and its IMFs, we have extracted the desired IMFs, which are used as a feature vector to create model/templates using the HMM. The engine test signal is evaluated against the model, which has returned the maximum *likelihood* considered as the representative class of the engine being tested. Experimental investigations are carried out to evaluate the proposed system in the fault diagnosis of a single-cylinder IC engine. The experimental results have indicated that the proposed method can extract the features and classify the different faults of IC engine.

Acknowledgments

The research presented in this chapter is supported by Technology Information, Forecasting and Assessment Council (TIFAC), Department of Science and Technology (DST), Government of India, under the project number TIFAC/EE/20070174.

References

[1] Gao, Q., Duan, C., Fan, H., Meng, Q.: Rotating machine fault diagnosis using empirical mode decomposition. Mechanical System and Signal Processing 22, 1072–1081 (2008)

[2] Li, H., Zhang, Y.: Bearing fault diagnosis based on EMD and Wigner-Ville distribution. In: Proceedings of 6th World Congress on Intelligent Control and Automation, Dalian, China, pp. 5447–5451 (2006)

[3] Wu, J.-D., Liu, C.-H.: Investigation of engine fault diagnosis using discrete wavelet transform and neural network. Expert Systems with Applications 35, 1200–1213 (2008)

[4] Sharkey, A.J.C., Gopinath, O., Chandroth: Accoustic emission, cylinder pressure and vibration: a multisensor approach to robust fault diagnosis. In: Proceedings of IJCNN, Comodem, Italy, pp. 223–228 (2000)

[5] Chandroth, G., Sharkey, A.J.C., Sharkey, N.E.: Cylinder pressure and vibration in internal combustion engine condition monitoring. Proceedings of Comadem 99, 294–297 (1999)

[6] Mba, D., Rao, R.B.K.N.: Development of acoustic emission technology for condition monitoring and diagnosis of rotating machines: bearings, pumps, gearboxes, engines, and rotating structures. The Shock and Vibration Digest 38, 3–16 (2006)

[7] Wu, J.-D., Chen, J.-C.: Continuous wavelet transform technique for fault signal diagnosis of internal combustion engines. NDT and E International 39, 304–311 (2006)

[8] Loutridis, S.J.: Damage detection in gear system using empirical mode decomposition. Engineering Structures 26, 1833–1841 (2004)

[9] Huang, N.E., et al.: The empirical mode decomposition and the Hilbert spectrum for nonlinear non-stationary time series analysis. Proceedings of the Royal society, London 454, 903–995 (1998)

[10] Pinesa, D., Salvinob, L.: Structural health monitoring using empirical mode decomposition and Hilbert phase. Journal of Sound and Vibration 294, 97–124 (2006)

[11] Li, H., Deng, X., Dai, H.: Structural damage detection using the combination method of EMD and wavelet analysis. Mechanical Systems and Signal Processing 21, 298–306 (2007)
[12] Ying, J., Kirubarajan, T., Pattipati, K.R., Patterson-Hine, A.: A hidden Markov model-based algorithm for fault diagnosis with partial and imperfect tests. IEEE Transaction Systems, Man and Cybernetics, Part C (Appendix and Reviews) 30, 463–473 (2007)
[13] Bunks, C., McCarthy, D., Al-Ani, T.: Condition-based maintenance of machines using hidden Markov models. Mechanical Systems and Signal Processing 14, 597–612 (2000)
[14] Ge, M., Du, R., Xu, Y.: Hidden Markov model based fault diagnosis for stamping process. Mechanical Systems and Signal Processing 18, 391–408 (2004)
[15] Li, Z., Wu, Z., He, Y., Fulei, C.: Hidden Markov model-based fault diagnostics method in speed-up, speed-down process for rotating machinery. Mechanical Systems and Signal Processing 19, 329–339 (2005)
[16] Xu, Y., Ge, M.: Hidden Markov model-based process monitoring system. Journal of Intelligent Manufacturing 15, 337–350 (2004)
[17] Internal Combustion Engine, The Columbia Encyclopedia, 6th edn. Columbia University Press, New York (2003)
[18] Zheng, G.T., Leung, A.Y.T.: Internal Combustion Engine Noise Analysis with Time-Frequency Distribution. Journal of Engineering for Gas Turbines and Power 124, 645–649 (2002)
[19] Huang, N.E., Shen, S.S.P.: Hilbert-Huang Transform and Its Applications. Interdisciplinary Mathematical Sciences 5, 1–24 (2005)
[20] Jinhui, X.: Hidden Markov Model (HMM) applied in the speech recognition, pp. 3–25. Hauzhong University of Science and technology Press, Wuhan (1995)
[21] Lawrence, R., Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of IEEE 77, 257–285 (1989)
[22] Xun, J., Yan, S.: A revised Hilbert-Huang transformation based on the neural networks and its application in vibration signal analysis of a deployable structure. Mechanical Systems and Signal Processing 22, 1705–1723 (2008)

An Intelligent Approach for Security Management of an Enterprise Network Using Planner

Nirnay Ghosh and S.K. Ghosh

Abstract. Over the past decade, there has been a revelation as far as the development of the computer network is concerned. With the increased reliance and dependency on the computer networks, security threats to these networks have increased substantially. Therefore, securing the network from attempted intrusions and actual attacks have become more frequent and widespread. Present day networks are vulnerable against multi-stage, multi-host attacks, which combine the vulnerabilities existing on different machines and cause more damage. One of the tools for analyzing security vulnerabilities in enterprise networks is *attack graph*. An *attack graph* consists of a number of *attack paths*, each of which is a chain of *exploits* which an attacker utilizes during different stages of any attack. Each *exploit* in the series satisfies the pre-conditions for subsequent *exploits* and makes a cause-effect relationship among them. An *attack graph* is a complete graph, which is used to correlate multi-stage, multi-host attacks to represent various attack scenarios. One of the intrinsic problems with the generation of such a full *attack graph* is its scalability. In this chapter, a novel approach based on an *artificial intelligence* techniques, called *planner*, has been reported for scalable representation of the *attack graphs*. A *planner* is a special purpose search algorithm for finding out solutions within a large state space and does not suffer from *state space explosion* problem. A case study has also been presented to demonstrate the efficacy of the approach.

1 Introduction

In today's enterprise, with increasing dependency on IT infrastructure, one the major objectives of a network administrator is to maintain a stable and secure network. With increasing security threats, the network security vulnerability must consider exploits in the context of multi-stage, multi-host attack scenarios. Therefore, the

Nirnay Ghosh · S.K. Ghosh
School of Information Technology, Indian Institute of Technology, Kharagpur - 721302, India
e-mail: nirnay.ghosh@gmail.com, skg@iitkgp.ac.in

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 187–214.
springerlink.com

greatest challenge for the network administrators is to appropriately analyze the network, configure it so that it becomes sufficiently both secure as well as operational. However, even well-administered networks are susceptible to some levels of attacks, since eliminating all these vulnerabilities is virtually isolating the network, which is not an option in most of the organizations. Present day's security technologies include some efficient network scanners, such as [3] [6] [4] [5], and so on. These scanning tools are useful as far as detecting vulnerabilities local to a system. They also provide with information regarding the kind of patch to be used to plug these bugs. However, these scanners do not identify all conditions for a complete attack to take place, or how different vulnerabilities existing in different systems are correlated to produce attacks, potentially more harmful than individual attacks. In general, complex, multi-stage attack scenarios are not characterized, if computer attacks are described in terms of a single-point exploited vulnerability or as a signature composed of a specific sequence of events. One such tool that gives description about the correlated attacks in a network is the *attack graph*. In some literatures, *attack graph* is also termed as the *exploit dependency graph* [24]. Typically, these exploits take advantage of the known vulnerabilities in various systems and services. Most of the known vulnerabilities are well documented in public sites, such as [1] [2] and in some commercial tools viz. [3] [6].

An *attack graph* shows the network administrator all possible sequences of attackers' actions that eventually lead to the desired level of privilege on the target. Therefore, a complete *attack graph* quickly becomes unmanageably large, as the network complexity grows past a few machines. Analysis shows that such an *attack graph* has exponential complexity. For scalable representation of *attack graphs*, some researchers have used *model-checker*-based formal methods. But, these *model-checkers* have *state space explosion* problem and do not handle realistic set of *exploits* even for moderate-sized real-world networks. Some of the approaches introduced the concept of *monotonicity*, which implies that once an attacker has gained certain level of privileges, he does not have to relinquish them. This removes the concept of back-tracking from the *attack graphs* and the complexity is improved from exponential to polynomial one. However, the *attack graph*, which is generated based on *monotonic* assumptions is still not scalable and also contains a number of redundant paths. This makes its visual presentation extremely clumsy, for which its apprehension becomes tough. But, it is desirable to present the network administrator with *attack graphs* that are understandable so that appropriate network hardening measures may be adopted. Therefore, a notion of *minimal attack graph* has been proposed. *Minimal attack graphs* consist of *attack paths* that always terminate to a desired *goal* node. But, different literature surveys have depicted that there exists no formal method to generate a scalable as well as time-efficient *minimal attack graph*.

In the present study, an attempt to generate time-efficient and scalable *minimal attack graph* has been made. *Planner*, a special purpose search algorithm from *artificial intelligence* domain, has been used as a *model-checker*, as it does not suffer from *state space explosion* problem.

The present chapter is organized into six sections. Section 2 provides a detailed study related to *planner*. The related works on *attack graph* for network security mechanism has been discussed in section 3. Section 4 describes the method of generation of *minimal attack graph* using a case study. Section 4.5 analyzes the efficacy of the proposed approach and a conclusion has been drawn in section 5.

2 Planner

A *planner* is a special purpose search algorithm from *artificial intelligence* domain, used for finding out solutions within a large state space and does not suffer from *state space explosion* problem. In this section, a detailed study on *GraphPlan* and *SGPlan*, the variants/implementation tools of *Planner*, has been presented. These are used for generation and scalable representation of *minimal attack graph*.

2.1 Introduction

The *Graphplan* algorithm is based on *Planning Graph* analysis, which plans in *STRIPS-like* domains [11]. In this approach, rather than immediately embarking upon a search as in standard planning methods, the algorithm begins by explicitly constructing a compact structure, the *Planning Graph*. A *Planning Graph* encodes the planning problem in such a way that many useful constraints inherent in the problem become explicitly available to reduce the amount of search needed [11]. Further, *Planning Graph*s can be constructed quickly: they have polynomial size and can be built in polynomial time. It is worth pointing out that a *Planning Graph* is *not* the state-space graph, which of course could be huge. In fact, unlike the state-space graph in which a plan is a *path* through the graph, in a *Planning Graph* a plan is essentially a *flow* in the network flow sense. *Planning Graph*s are closer in spirit to the Problem Space Graphs (PSGs), though unlike PSGs, *Planning Graph*s are based not only on domain information, but also the goals and initial conditions of a problem and an explicit notion of time. Planning Graphs offer a means of organizing and maintaining search information that is reminiscent of the efficient solutions to Dynamic Programming problems. *Planning Graph* Analysis appears to have significant practical value in solving planning problems even though the inherent complexity of STRIPS planning, which is at least *PSPACE-hard*, is much greater than the complexity of standard Dynamic Programming problems. We provide empirical evidence on a variety of natural and artificial domains showing that Planning Graph Analysis is able to provide with a quite substantial improvement in running time. The *Graphplan* uses the *Planning Graph* that it creates to guide its search for a plan. The search that it performs combines aspects of both total-order and partial-order planners. Like traditional total-order *planners*, *Graphplan* makes strong commitments in its search. The semantics of such a plan is that the actions in a given time step may be performed in any desired order. Conceptually, this is a kind of parallel plan, since one could imagine executing the actions in three time steps, if one had as many workers as needed to load and unload and fly the rockets. One valuable feature of our algorithm is that it guarantees to find the shortest plan among those in which independent actions may take place at the same time. Empirically and subjectively, these sorts of plans seem particularly sensible. Another significant feature of the algorithm is that it is not particularly sensitive to the order of the goals in a planning task, unlike traditional approaches.

2.2 Definitions and Notations

Planning Graph Analysis applies to STRIPS-like planning domains. In these domains, operators have preconditions, add-effects, and delete-effects, all of which are conjuncts of propositions, and have parameters that can be instantiated to objects in the world. Operators do not create or destroy objects and time may be represented discretely. Specifically, a planning problem will have the following parameters:

- STRIPS-like domain (a set of operators),
- Set of objects,
- Set of propositions (literals) called the *Initial Conditions*,
- Set of Problem Goals which are propositions that are required to be true at the end of a plan.

The *action* is defined as a fully-instantiated operator. For instance, the operator *put ?x into ?y* may instantiate to the specific action *put Object1 into Container2*. An action taken at time t adds to the world all the propositions, which are among its $Add-Effects$ and deletes all the propositions which are among its $Delete-Effects$. It will be convenient to think of doing nothing to a proposition in a time step as a special kind of action, termed as a $no-op$ or frame action.

2.3 Description of the Algorithm

The high-level description of our basic algorithm is the following. Starting with a *Planning Graph* that only has a single proposition level containing the Initial Conditions, Graphplan runs in stages. In stage i, Graphplan takes the *Planning Graph* from stage $i-1$, extends it one time step (the next action level and the following proposition level), and then searches the extended *Planning Graph* for a valid plan of length i. Graphplans search either finds a valid plan (in which case it halts) or else determines that the goals are not all achievable by time i (in which case, it goes on to the next stage). Thus, in each iteration through this Extend/Search loop, the algorithm either discovers a plan or else proves that no plan having that many time steps or fewer is possible. Graphplans algorithm is sound and complete: any plan the algorithm finds is a legal plan, and if there exists a legal plan then Graphplan will find one.

2.3.1 Extending *Planning Graph*s

All the initial conditions are placed in the first proposition level of the graph. To create a generic action level, we do the following. For each operator and each way of instantiating preconditions of that operator to propositions in the previous level, insert an action node *if no two of its preconditions are labeled as mutually exclusive*. Also, insert all the no-op actions and insert the precondition edges. To create a generic proposition level, simply look at all the Add-Effects of the actions in the previous level (including no-ops) and place them in the next level as propositions, connecting them via the appropriate add and delete-edges. Mark two propositions as exclusive, if all ways of generating the first are exclusive of all ways of generating the second. As we demonstrate in the following theorem, the time taken by our algorithm to create this graph structure is polynomial in the length of the problems description and the number of time steps.

Theorem 1. *Consider a planning problem with n objects, p propositions in the Initial Conditions, and m STRIPS operators each having a constant number of formal parameters. Let l be the length of the longest add-list of any of the operators. Then, the size of a t-level planning graph created by Graphplan, and the time needed to create the graph, are polynomial in n, m, p, l, and t.*

Proof. Let k be the largest number of formal parameters in any operator. Since operators cannot create new objects, the number of different propositions that can be created by instantiating an operator is O(lnk). So, the maximum number of nodes in any proposition level of the planning graph is O($p + mln^k$). Since any operator can be instantiated in at most O(nk) distinct ways, the maximum number of nodes in any action-level of the planning graph is O(mn^k). Thus, the total size of the planning graph is polynomial in n, m, p, l, and t, since k is constant.

The time needed to create a new action and proposition level of the graph can be broken down into (A) the time to instantiate the operators in all possible ways to preconditions in the previous proposition-level, (B) the time to determine mutual exclusion relations between actions, and (C) the time to determine the mutual exclusion relations in the next level of propositions. It is clear that this time is polynomial in the number of nodes in the current level of the graph.

2.3.2 Searching for a Plan

Given a *Planning Graph*, Graphplan searches for a valid plan using a backward-chaining strategy. Unlike most other planners, however, it uses a level-by-level approach, in order to best make use of the mutual exclusion constraints. In particular, given a set of goals at time t, it attempts to find a set of actions (no-ops included) at time $t-1$ having these goals as add effects. The preconditions to these actions form a set of subgoals at time $t-1$ having the property that if these goals can be achieved in $t-1$ steps, then the original goals can be achieved in t steps. If the goal set at time $t-1$ turns out to be not solvable, Graphplan tries to find a different set of actions, continuing until it either succeeds or has proven that the original set of goals is not solvable at time t.

In order to implement this strategy, Graphplan uses the following recursive search method. For each goal at time t in some arbitrary order, select some action at time $t-1$ achieving the goal that is not exclusive of any actions that have already been selected. Continue recursively with the next goal at time t. (Of course, if by good fortune a goal has already been achieved by some previously-selected action, we do not need to select a new action for it). If our recursive call returns failure, then try a different action achieving our current goal, and so forth, returning failure once all such actions have been tried. Once finished with all the goals at time t, the preconditions to the selected actions make up the new goal set at time $t-1$. We call this a goal-set creation step. Graphplan then continues this procedure at time step $t-1$.

A forward-checking improvement to this approach (which is implemented in Graphplan and helps modestly in our experiments) is that after each action is considered, a check is made to make sure that no goal ahead in the list has been cut-off. In other words, Graphplan checks to see if for some goal still ahead in the list, all the actions creating it are exclusive of actions we have currently selected. If there is some such goal, then Graphplan knows that it needs to back up right away.

2.4 Additional Features

We have discussed so far the basic algorithm used by Graphplan. We now describe a few additional features that can be added in a natural way (and have been added as options in our implementation), and discuss their significance.

The first feature is a type of reasoning that is quite natural in our framework. The reasoning is that if the current goal set contains n goals, such that no two of them can be made true at the same time by a non-loop action (and none of them are present in the Initial Conditions), then any plan will require at least n steps. For instance, one could use this reasoning in a path-finding domain to show that it must take at least n steps to visit n distinct places. Unfortunately, finding the largest such subset of any given goal set is equivalent to the maximum Clique problem. However, we can find a *maximal* such set using greedy methods.

This form of reasoning turns out to be very useful on traveling-salesman-like problems, where the goal is to visit all the nodes in a graph in as few steps as possible. On very dense graphs (such as the complete graph) for which the problem should be easy, Graphplan without this reasoning can be quite slow because the pairwise exclusion relations do not propagate well. For instance, on a complete graph, after two time steps any two goals of the form visited X will be non-exclusive. However, with this reasoning, Graphplans performance is more respectable.

A second feature concerns graph creation. Although, as demonstrated in Theorem 1, the graph size is polynomial, it may be unnecessarily large if there are many irrelevant facts in the initial conditions. One way around this problem is to begin with a regression analysis going backward from the goals to determine if any initial conditions may be thrown out.

One final feature (not currently in our implementation) that could be added easily is the ability to use the information learned on one planning problem for another problem on the same domain having the same Initial Conditions. Specifically, the same graph and the same memorized unsolvable goal sets could be re-used in this case.

Therefore, the motivations behind selecting *Planner* as a technique for generating attack paths are as follows:

1. *Planner* makes direct connection between states and actions, thereby prunes unnecessary actions from the system. This enables it to find the *shortest* path among those, in which independent actions may take place at the same time.
2. It is free to add actions to the plan wherever they are required, rather than incremental way starting from the initial state. This helps in our domain by adding new exploits whenever they are published.
3. It uses richer input language, PDDL, to express complex state space domains relatively easier than custom built analysis engines.

3 Related Works

Attack graph has been a topic of active research in recent time. A good number of research works have been reported in various literatures related to the *attack graphs*. The literature survey done in this section primarily focuses on the following aspects of the *attack graphs*:

- Construction of attack graphs to analyze network security;
- Scalable and polynomial representation of attack graphs in terms of space and time, respectively;
- Presenting formal languages that can be used to describe actions and states in attack graph;
- Determining appropriate network hardening measures in terms of services and costs from attack graphs;
- Improving visual representation of complete attack graphs.

One of the earliest work in the field of *attack graph* was done by Moskowitz et al. in [22]. The authors used a graph to represent *insecurity flow* to identify the possible loop-holes in a network. Edges represent penetration of a security barrier viz. firewall and are weighted with the probability of successfully breaching the defense to discover a security hole from a source machine to a target machine. Assigning weights to the edges in the graphs related to the security assessment of any network was first formalized in [14]. The authors in this paper have introduced the concept of a *privilege graph* where the nodes represents the a set of privileges for a user or a group of users, and the edges represent the vulnerabilities that govern the state transitions. The privilege graph is transformed into a Markov Model for quantification of security. The Markov model represents all possible probing sequences of a non-omniscient hacker.

Swiler et al. gave a formal definition of *attack graph* in their paper [28]. It is a tool, which can identify the set of attack paths that have high probability of success or low effort cost from the attacker's view point. Each edge has a weight representing a success probability, which is a function of configuration and attacker's profile. The different fields that an *attack graph* node should contain as: *User level, Machine(s), Vulnerabilities, Capabilities, and States*. A backward search algorithm is used to generate the *attack graph*, which is NP-complete. In [25], the authors have used privilege graphs to describe the security of a single UNIX system, in which 13 known vulnerabilities were present. The authors have used similar mathematical model as described in [14] to calculate the *mean effort to security failure (METF)*. They have extended this METF computation for different graph building approaches, such as *Total memory (TM)* (based on BFS), *Memoryless (ML)* (based on DFS), and *Shortest path (SP)* assumptions. In [31], a manual approach to generate *attack graphs* following a backward reachable strategy is proposed, where the nodes below the goal node are the actions that can reach this goal. Actions are combined using either OR (disjunctive) or AND (conjunctive) logics.

Ritchey and Ammann in [30] have shown how a chain of exploits may be used by an attacker to reach from the source node to the goal node, i.e., attacks due to the security ramifications of offering a variety of combined services. The authors have used *SMV model-checkers* [7] to determine if a final goal state is reachable from an attacker starting with limited privileges. The model checker can handle large set spaces and generate counterexamples efficiently if any predefined security invariant turns out to be false. Swiler et al. extended their work [28] on development of *attack graph* generation tool in [35] by eliminating redundant nodes and finding a set of near-optimal shortest path from a given *attack graph* that indicates the most exploitable components of the system configuration.

Templeton and Levitt in [36] discussed about complex attack scenarios that are generated automatically by linking multiple attack actions and subgoals, which are called concepts. The attacks are viewed as a set of capabilities that provide support for abstract attack concepts that, in turn, provides new capabilities to support other concepts. This model does not require a priori knowledge of a particular scenario, as the attack concepts are defined locally. Tidwell et al. [37] described an enhanced attack tree model of Internet attacks and system as well as attack specification languages based on BNF grammars. In this chapter, Internet attack is modeled, so that composite attacks may be captured and a distributed attack notification and visualization system may be described. The model extends the work of [31] related to attack trees.

An automated technique for generating and analyzing *attack graphs* using symbolic model checking algorithm is presented in [33]. For modeling the network, the authors have deployed a special purpose compiler which takes XML descriptions as input and translates into input language of the model checker. The *attack graph* generated by the model checker is exhaustive and succinct. The network model includes intrusion detection components and distinguishes between stealthy (not detectable by the IDS) and detectable attack variants. Ritchey et al. [29] used a state-based model of network security, called Topological Vulnerability Analysis (TVA), to discover the attack paths. The proposed TVA consists of four major elements, such as a network of hosts, including the network services, components, and configuration details that give rise to vulnerabilities, connectivity of the hosts, exploits, or attacks that changes the state of the model, and a list of security requirements that the model should attempt to validate. In [10], Artz describes NetSPA (Network Security Planning Architecture) system that generates a worst-case *attack graph* using a *depth-limited forward-chaining depth-first* algorithm. This is a C++ tool that takes input from custom database on host and software types and versions, intrusion detection system placement, gateways between subnets, firewall rules, and exploits. Informations, such as critical network resources and attacker's initial location are provided during runtime. Informations regarding the vulnerabilities, firewall rules, and the network topology are entered manually into the database. Ammann et al. [9] proposed an algorithm for more compact and scalable representation of *attack graphs*. This approach relies on an explicit assumption of monotonicity. This assumption reduces the complexity of analysis problem from exponential to polynomial. Jajodia et al. [21] described a Topological Vulnerability Analysis (TVA) tool that models network security conditions and attack techniques (exploits), automatic population of models via Nessus [3], and analysis of exploit sequence leading to specific attack goals. The tool generates a graph of dependencies among exploits that represent all possible attack paths without having to enumerate them explicitly. The polynomial time algorithm described in [9] is used to construct and analyze the graph. The generated *attack graph* is subjected to symbolic analysis technique to identify sets of actions (e.g., patch vulnerabilities, network services, remove program on hosts) that, if followed, prevent the attacker from reaching the goal states. In [24], Noel et al. used a polynomially efficient *exploit-dependency* representation based on monotonicity to compute *actual* set of network hardening measures that guarantee the safety of given critical resources and also present the assignments that minimize the overall network hardening cost with minimal impact on the network.

In [23], the authors described various approaches to collapse the *attack graphs* generated by TVA [21] to make visual understanding easier. The display used exploit-dependency graphs. In those graphs, exploits between different hosts were treated as separate and unique and appears only once as nodes. Various approaches to group exploit-dependency graphs were presented in that work. In [16], a formalism and methodology for multi-stage network attack analysis was proposed. A depth-limited attack chain algorithm was applied on a full *attack graph* to generate a *minimum cut set* or *Goal-inducing attack chains (GACs)*. It was the smallest collection of paths, such that if any one vulnerability was removed, correct prediction whether the final goal is reachable could not be done. Combining minimum *GACs*, attack trees could be constructed in which the common goal identified by the *GACs* was the root node. Sheynar et al. [34] built an *attack graph* generation tool based on the concept proposed in [33] [32]. The tool consists of three components viz. *network model builder, scenario graph generator, and graphical user interface (GUI)*. The tool used three external data sources to provide automatic information which were *Nessus* vulnerability scanner [3], *MITRE Corp.'s Outpost*, and *Lockheed Martin's ANGI* that the model builder used to create a finite model and encode it in the input language of the scenario graph generator. The scenario graph generator then built the actual *attack graph* through execution of a model checker. The graphical user interface displayed the generated *attack graph*. Zhang et al. [42] presented a forward-search, breadth-first and depth-limited algorithm to produce attack paths and implemented the tools to generate the attack path. The authors followed the approaches proposed in [28] [9] [33] to build a model for the network security analysis referencing the attack character that brings privilege escalation.

In [8], the authors presented an intuitive, polynomially efficient, and scalable vulnerability analysis approach, from a penetration tester's perspective, that generates suboptimal attack paths rather than the complete graph. This chapter presented a novel approach to automate penetration testing, which is essentially an expensive, labor intensive, and often incomplete. It considered the mental model of system administrator and penetration tester, who analyzed network security in terms of maximal compromise possible on each host and also provided them with suboptimal choices of making the network secure. The proposed model constructs an *access graph* with a node for each host in the network. A *directed edge* between a pair of hosts represents the access that is available. Instead of adding multiple edges onto the *access graph*, only edges with maximal access are retained.

Wang et al. [41] proposed a *Multi-Stage Finite State Machine Model (M-FSM)* to reason how each step-activity within the whole operation contributes to the attack goal. Initially, the model represented each attack as an *atom FSM (aFSM)*. Each *aFSM* consists of a single transition and two states viz. the preconditions, and the postconditions. The state transitions is given by the application of a single exploit. These *aFSM*s can be easily combined to depict M-FSM, modeling vulnerable operations and possible exploits. A safety configuration expression is derived whose analysis yields all possible hardening measures that have minimal impact on services. In [26] [27], the authors proposed methods for generation of polynomial-sized logical *attack graphs* using a network security analyzer based on logic programming known as *MulVAL*. *MulVAL* produced counter-examples for a given security policy over an enterprise network. It used *Datalog* as its input specification language. The logical *attack graph* directly illustrates logical dependencies among attack goals configuration information. A node in such a graph is a logical statement. The edges in the graph specify the causality relations between network configuration and an attacker's potential privilege. In [20], the authors proposed a *multi-prerequisite graph* that scaled nearly linearly with the size of a network. A prototype system, called NetSPA, used readily available source data to automatically calculate network reachability, classify vulnerabilities, generate the graph, and provide recommendations to improve security. The network topology was obtained from user and the Nessus Vulnerability scanner [3]. For reachability computation, NetSPA merged sections of the matrix into *reachablity groups* by collapsing the filtering rulesets into *Binary Decision Diagrams (BDDs)*. The reachability groups were either intra-subnet where the rulesets of the filtering device had no impact, or might be inter-subnet which were treated identically by the filtering device on the network. The maximum number of state nodes in the *multi-prerequisite graph* was linearly related to the number of *vulnerability instance, state, reachability group, and credential*. The graph building algorithm was based on breadth-first technique.

Quantification of risks was done in [12], where the authors analyzed the risk related to *Commercial Off The Shelf (COTS)* systems using a quantitative threat modeling method based on attack path (T-MAP) analysis. Dantu et al. [15] estimated the type of attack behavior as they assumed that sequence of network actions by attackers depends on their social and attack profile. In [40], the authors proposed a metric, called *attack resistance*, for assessing and comparing the security of different network configurations. The metric was based on intuitive properties derived from *common sense*. In [39], probabilistic metric for network security was proposed to quantify the likelihood of potential multi-step attacks that combined multiple vulnerabilities. This metric combined measurements of individual vulnerabilities obtained from the existing metrics and the casual relationships between them as encoded in an attack graph. Feng et al. [17] described a flexible approach of measuring security of crucial resources in a vulnerable network with *incomplete* input data and also proposed a *backward iterative algorithm* to solve the problem of cyclic attack paths in measuring security using *attack graph*s. In [38], the authors developed a novel quantitative security metric, *VEA-bility*, which presented an accurate estimation of the comparative desirability of various network configurations, among which an administrator can choose the best one. Frigault et al. [19] proposed a probability metric based on *attack graph*s as a special *Bayesian Network (BN)*. *Probability values* were assigned to each node in the *attack graph* and then *Conditional Probability Tables* were developed for each of them. The nodes in the *attack graph* represented vulnerabilities as well as the pre and post conditions resulting from the exploitation of such vulnerabilities.

In the present chapter, an intelligent technique from *artificial intelligence* domain has been used to generate *minimal attack paths* and then the generated paths are collapsed to form a *minimal attack graph*.

4 Generation of Minimal Attack Graph Using Planner

In this section, an approach to generate *minimal attack graph* using *Planner* has been proposed. The *Planner* function is depicted in Figure 1. It starts with the assumption that the initial network configuration and the vulnerability analysis has been done apriori and are inputs to our *Planner* engine, i.e., the *domain.pddl* and the *fact.pddl* files written in PDDL [18]. With the initial network configurations, connectivity relationships, vulnerability analysis, a *minimal* attack path is generated. To generate other *minimal* attack paths, the *fact.pddl* file is modified. If all the attack paths are generated, they are collapsed to form the *minimal attack graph*. In nutshell, the overall mechanism is given below:

1. Initial network configuration and description of the *exploits* (in form of *domain* and *fact* files) are inputs to *planner* to generate a *shortest attack path*.
2. Customized *attack path enumeration* algorithm does automatic modification of *fact.pddl* to generate the next *shortest attack path* or no path.

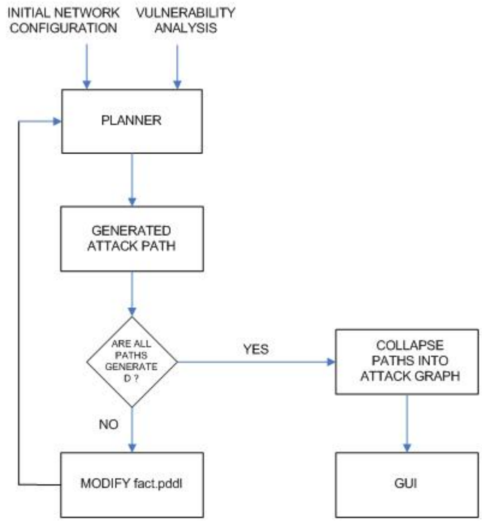


Fig. 1 Flow Chart showing *Planner* actions

3. Customized *attack graph building* algorithm collapses the generated *shortest attack paths* to generate a *minimal attack graph*.

A case study has been presented in the next section to demonstrate the efficacy of the proposed methodology.

4.1 Case Study

The test network used in this work is shown in Figure 2. It is similar to the one considered by Sheynar [32]. The network consists of four hosts viz. $Host0(H0)$, $Host1(H1)$, $Host2(H2)$, and $Host3(H3)$. The internal network is separated from the Internet by a firewall. The *Attacker* is assumed to be someone from the external network and all the hosts in the network are in the active state. $H3$ is taken as our target machine or *goal* and the MySQL[1] database running on that machine is the critical resource. The system characteristics of the hosts in the network including their operating system type, services running in them, number of open ports, and existing *generic* vulnerabilities, are composed in the Table 1. These data are obtained from *Nessus* [3] vulnerability scanner and public websites viz. NVD [2], Bugtraq [1]. Each *generic* vulnerability present in Table 1 has an effect. These effects obtained from [2] [1] [32] [34] may be summarized in Table 2. The firewall in the test network (refer to Figure 2) allows external hosts to connect to IIS web service running on port 80 on $H0$. But, connections to all other ports are blocked. The internal hosts are allowed to connect on any port within the network. The connectivity limiting firewall policy is presented in Table 3. An entry in Table 3 specifies

[1] http://www.mysql.com

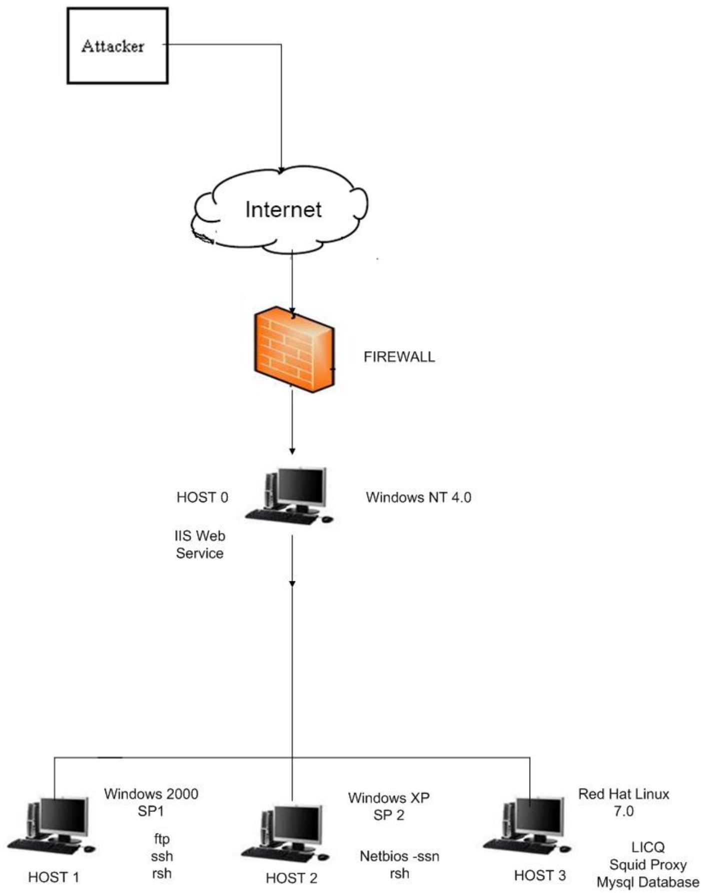


Fig. 2 Test Network

connectivity between a pair of hosts. The different values in the table has the following significances:

- *All- source* host may connect to any port on the *destination* host.
- *Yes- source* host may connect to port number 80 on the *destination* host.
- *None- source* host cannot connect to any port on the *destination* host.

Depending upon connectivity limiting firewall policies, each *generic* exploit results in some *instantiated* exploits [9]. Some of the *instantiated* exploits are as follows:

- *IIS_bof(0,0)*- IIS buffer overflow exploit from $Host0$ to $Host0$.
- *ftp_rhosts (0,1)*- rsh trust from $Host0$ to $Host1$.

Table 1 System Characteristics

Host	Services	Ports	Vulnerabilities	CVE – IDs	OperatingSystem
H0	IIS Web Service	80	IIS buffer overflow	CVE-2002-0364	Windows NT 4.0
H1	ftp ssh rsh	21 22 514	ftp rhost overwrite ssh buffer overflow rsh login	CVE-2008-1396 CVE-1999-1455 CVE-1999-0180	Windows 2000 SP1
H2	Netbios-ssn rsh	139 514	Netbios-ssn nullsession rsh login	CVE-2003-0661 CVE-1999-0180	Windows XP SP2
H3	LICQ Squid Proxy Mysql DB	5190 80 3306	LICQ-remote-to-user squid-port-scan local-setuid-bof	CVE-2001-0439 CVE-2001-1030 CVE-2006-3368	Red Hat Linux 7.0

Table 2 Vulnerabilities and their effects

Vulnerability(actions)	CVE – ID	Effects
IIS buffer overflow	CVE-2002-0364	Remotely gains root privileges
ftp rhost overwrite	CVE-2008-1396	Creates a .rhost file in *ftp home* directory establishing remote login trust relationship.
ssh buffer overflow	CVE-1999-1455	Gives a root shell on the victim machine
rsh login	CVE-1999-0180	Existing remote login trust relationship is used to log into target machine without providing password
netbios-ssn-nullsession	CVE-2003-0661	Establishing null session with blank username, password, or domain to list resources (shares)
LICQ-remote-to-user	CVE-2001-0439	Gives user privilege remotely by executing arbitrary command on victim machine
squid-port-scan	CVE-2001-1030	Scans network ports on machines which is otherwise inaccessible
local-setiud-bof	CVE-2006-3378	Exploiting buffer overflow on a *setuid root* file giving *superuser* privilege on alocal machine

Table 3 Connectivity-Limiting Firewall Policies

Host	Attacker	H0	H1	H2	H3
Attacker	All	Yes	None	None	None
H0	None	All	All	All	All
H1	None	Yes	All	All	All
H2	None	Yes	All	All	All
H3	None	Yes	All	All	All

- *squid_port_scan (1,3)*- squid port scan done from $Host1$ on $Host3$.
- *LICQ_remote_to_user (1,3)*- LICQ-remote-to-user exploit is used by $Host1$ to obtain user level privilege on $Host3$.
- *local_setuid_bof (3,3)*- Root privilege is obtained on $Host3$ by exploiting local buffer overflow exploit.
- *netbios_ssn_nullsession (1,2)*- User level privilege is obtained on $Host2$ from $Host1$ by utilizing netbios ssn-nullsession exploit.

4.2 Identification of Attack Path Using GraphPlan

GraphPlan, a variant of *Planner*, is a search algorithm which finds out solution within a large state space. Initial network configuration, attacker's objective, and exploits are considered as inputs to the *GraphPlan*. The *domain.pddl* contains predicates, functions, and exploit descriptions characterized by their *preconditions* and *effects*. The *fact.pddl* represents the network objects viz. hosts, firewalls, initial state, and the *goal* state as instantiated predicates (like *(superuser-priv Attacker H2)* implies attacker need *superuser* privilege on $Host2$). Removal of *monotonicity* assumption in *plangraph* generation helps to model some real world attacks like *denial-of-service(DoS)*, where an attacker has to relinquish his control over the victim. Once a *goal* is identified, the *Planner* does a *backward-chaining search* to identify the actual attack path.

In this work, *SGPlan 5.2.2*[2] [13], a variant of *GraphPlan*, is used as an attack path identification component. *SGPlan* is a classical *GraphPlanner* that partitions a large problem into sub-problems, each with its own sub-goal. It detects a reasonable order among sub-goals, hierarchically decomposes each sub-problem, and uses a search-space-reduction algorithm to eliminate irrelevant actions in the sub-problem. We have preferred *SGPlan* to other variants of *GraphPlan* viz. *LPG-td*[3], *Metric-FF*[4], as it supports numeric predicates or fluents, derivative predicates, and durative predicates. Attack path is represented as an exploit sequence, where each node represents the *exploit*, and each edge represents either a *condition* or an *available exploit* or the *privilege* gained after the application of that *exploit*.

4.2.1 *Domain* and *Fact* Files

As mentioned in section 2, *Planner* requires two files viz. *domain.pddl* and *fact.pddl*. The *domain.pddl* encodes the network configurations, vulnerability instances, user privileges, and possible set of exploits in predicate form. An instance of *domain.pddl* is shown in Figure 3. From Figure 3, it can be noted that the network configurations viz. service running on hosts, vulnerabilities present on the hosts, transport and application layer connectivity between the hosts is depicted in the *predicate* part of the file. We have defined two *functions* viz. *has_priv*, to assign privilege levels, and

[2] http://manip.crhc.uiuc.edu/programs/SGPlan/sgplan5.html
[3] http://www.zeus.ing.unibs.it/lpg/
[4] http://www.members.deri.at/ joergh/metric-ff.html

```
(define(domain attackgraph)
   (:requirements :strips :fluents :equality)
   (:predicates

                          (IIS_web_service ?H)
                          (ftp ?H)
                          (ssh ?H)
                          (rsh ?H)
                          (netbios_ssn ?H)
                          (LICQ_chat_service ?H)
                          (squid_proxy ?H)

                          (IIS_bof ?H)
                          (ftp_rhost_overwrite ?H)
                          (rsh_login ?H)
                          (ssh_bof ?H)
                          (netbios_ssn_nullsession ?H)
                          (LICQ_remote_to_user ?H)
                          (local_setuid_bof ?H)

                          (IIS_port_connectivity ?S ?T)
                          (ftp_port_connectivity ?S ?T)
                          (ssh_port_connectivity ?S ?T)
                          (squid_port_connectivity ?S ?T)
                          (LICQ_port_connectivity ?S ?T)
                          (rsh_port_connectivity ?S ?T)
                          (netbios_port_connectivity ?S ?T)

                          (IIS_apps_connectivity ?S ?T)
                          (ftp_apps_connectivity ?S ?T)
                          (ssh_apps_connectivity ?S ?T)
                          (squid_apps_connectivity ?S ?T)
                          (LICQ_apps_connectivity ?S ?T)
                          (rsh_apps_connectivity ?S ?T)
                          (netbios_apps_connectivity ?S ?T))


   (:functions (has_priv ?A ?H)
                    (root_priv)
                    (user_priv)
                    (none_priv))

   (:functions     (port_scan ?A ?H)
                   (port_scan_done)
                   (port_scan_not_done))


   (:action IIS-buffovflw
       :parameters
        (?A
         ?S
         ?T)

       :precondition
            (and (>=(has_priv ?A ?S) (user_priv)) (IIS_web_service ?T)
            (IIS_port_connectivity ?S ?T) (IIS_bof ?T) (< (has_priv ?A ?T)
            (root_priv)))

       :effect
            (and (not (IIS_bof ?T)) (assign (has_priv ?A ?T)(root_priv))
            (not (IIS_web_service ?T)))
   )
```

Fig. 3 domain.pddl for the test network (refer to Figure 2) *port_scan*, to determine whether port scanning has been done or not. The possible privilege level are *root, user, and none*. It may be noted (refer to Table 2) that there are *eight* vulnerabilities in the test network, each of which may be encoded in *formal logic* as an *action rule*. The *action rule* specification has four components: *intruder precondition, intruder effect, network preconditions, and network effects* [32]. Following the rule set given in [32] [34], the *actions* in *domain.pddl* have been designed. The *IIS-buffovflw action* has been shown in Figure 3 which has some *preconditions* and *effects*. The generic *parameters* A, S, and T, representing *an attacker, a source,* and *a destination* respectively, are used for handling of the *predicates* and the *functions*.

The *fact.pddl* encodes various network *objects* that includes the hosts, the attacker, the firewall etc. In this work, there are five objects viz. *Host0, Host1, Host2, Host3, Attacker*. There are a set of *propositions* or *literals* which identify the initial network conditions and the *goal*. *SGPlan's* ability to represent numeric literals helps to assign numerical values to predicates for privilege levels and made them comparable. The *fact.pddl* (refer to Figure 4) contains numerical predicates like *(=(root_priv) 3), (=(user_priv) 2), (=(none_priv) 1)* with respect to function *has_priv* and *(=(port_scan_not_done) 0), (=(port_scan_done) 1)* for function *port_scan*. From Figure 4, it can be noted that a *goal* condition is given as *(:goal (and (= (has_priv Attacker Host3) 3)))*, which suggests that the *attacker's* objective is to acquire *root* privilege on *Host*3. The initial network conditions include the service running in the hosts (*IIS_web_service Host0, ssh Host1, netbios_ssn Host2*), vulnerability instances in the hosts (*IIS_bof Host0, ssh_bof Host1, ftp_rhost_overwrite Host1, netbios_ssn_nullsession Host2*), transport layer connectivities (*IIS_port_connectivity Host0 Host1, ssh_port_connectivity Host2 Host1, ftp_port_connectivity Host3 Host1*), and application layer connectivities (*netbios_apps_connectivity Host1 Host2, ftp_apps_connectivity Host0 Host1*) among different pair of hosts.

SGPlan uses *domain.pddl* and *fact.pddl* to generate single *shortest* attack path. Systematic invalidation of the identified path enables *SGPlan* to identify alternate *shortest* attack path. It depends on the administrator's discretion about which network configurations should be changed to invalidate the paths. Invalidation is done in *fact.pddl* by disabling a service running in one of the hosts, or a connectivity between a pair of hosts by placing a *double-semicolon (;;)* before that predicate. From the given *domain.pddl* and *fact.pddl*, the *shortest* attack path generated by *SGPlan* is as follows:

```
;Time 0.01
; ParsingTime 0.00
; NrActions 6
; MakeSpan
; MetricValue
; PlanningTechnique Modified-FF(enforced hill-climbing search) as the subplanner
0.001: (IIS-BUFFOVFLW ATTACKER ATTACKER HOST0) [1]
1.002: (FTP-RHOST ATTACKER HOST0 HOST1) [1]
```

```
(define (problem Attack)
    (:domain attackgraph)
    (:objects
        Host0
        Host1
        Host2
        Host3
        Attacker
    )

    (:init
       (= (has_priv  Attacker Attacker) 3)
       (= (has_priv  Attacker Host0) 1)
       (= (has_priv  Attacker Host1) 1)
       (= (has_priv  Attacker Host2) 1)
       (= (has_priv  Attacker Host3) 1)

       (= (root_priv) 3)
       (= (user_priv) 2)
       (= (none_priv) 1)

       (= (port_scan Attacker Host3) 0)
       (= (port_scan_not_done) 0)
       (= (port_scan_done) 1)

       (IIS_web_service Host0)
       (ssh Host1)
       (ftp Host1)
       (rsh Host1)
       (netbios_ssn Host2)
       (squid_proxy Host3)
       (LICQ_chat_service Host3)

       (IIS_bof Host0)
       (ssh_bof Host1)
       (ftp_rhost_overwrite Host1)
       (rsh_login Host1)
       (netbios_ssn_nullsession Host2)
       (LICQ_remote_to_user Host3)
       (local_setuid_bof Host3)

       (IIS_port_connectivity Attacker Host0)

       (ssh_port_connectivity Host0 Host1)
       (ssh_apps_connectivity Host0 Host1)
       (ssh_port_connectivity Host2 Host1)
       (ssh_apps_connectivity Host2 Host1)
       (ssh_port_connectivity Host3 Host1)
       (ssh_apps_connectivity Host3 Host1)

       (ftp_port_connectivity Host0 Host1)
       (ftp_apps_connectivity Host0 Host1)
       (ftp_port_connectivity Host2 Host1)
       (ftp_apps_connectivity Host2 Host1)
```

```
(ftp_port_connectivity Host3 Host1)
       (ftp_apps_connectivity Host3 Host1)

       (netbios_port_connectivity Host0 Host2)
       (netbios_apps_connectivity Host0 Host2)
       (netbios_port_connectivity Host1 Host2)
       (netbios_apps_connectivity Host1 Host2)

       (squid_port_connectivity Host0 Host3)
       (squid_port_connectivity Host1 Host3)
       (squid_port_connectivity Host2 Host3)

       (LICQ_port_connectivity Host0 Host3)
       (LICQ_port_connectivity Host1 Host3)
       (LICQ_port_connectivity Host2 Host3)

)

    (:goal (and(= (has_priv  Attacker Host3) 3)))
)
```

Fig. 4 fact.pddl for the test network (refer to Figure 2)

2.003: (RSH-LOGIN ATTACKER HOST0 HOST1) [1]
3.004: (SQUID-PORT-SCAN ATTACKER HOST1 HOST3) [1]
4.005: (LICQ-REMOTE-TO-USER ATTACKER HOST1 HOST3) [1]
5.006: (LOCAL-SETUID-BUFFOVRFLW ATTACKER HOST3) [1]

SGPlan generated attack path may be re-written in the following way:

$Attacker \rightarrow IIS\_bof(Att,H0) \rightarrow squid\_port\_scan(H0,H3) \rightarrow LICQ\_remote\_to\_user(H0,H3) \rightarrow local\_setuid\_bof(H3,H3).$

If another alternate attack path is to be generated, the *fact.pddl* needs to be modified. Figure 5 shows a modified *fact.pddl* file, where the administrator has disabled transport layer connectivity between $Host0$ and $Host3$ on *Squid_proxy* and *LICQ* services by using *double-semicolons (;;)*.

The *shortest* attack path generated by *SGPlan* using this modified *fact.pddl* and the existent *domain.pddl* is given below:

; Time 0.01
; ParsingTime 0.00
; NrActions 5
; MakeSpan
; MetricValue
; PlanningTechnique Modified-FF(enforced hill-climbing search) as the subplanner
0.001: (IIS-BUFFOVFLW ATTACKER ATTACKER HOST0) [1]
1.002: (NETBIOS-NULLSESSION ATTACKER HOST0 HOST2) [1]
2.003: (SQUID-PORT-SCAN ATTACKER HOST2 HOST3) [1]
3.004: (LICQ-REMOTE-TO-USER ATTACKER HOST2 HOST3) [1]
4.005: (LOCAL-SETUID-BUFFOVRFLW ATTACKER HOST3) [1]

SGPlan generated attack path may be represented in the following way:

$Attacker \rightarrow IIS\_bof(Att,H0) \rightarrow ssh\_bof(H0,H1) \rightarrow squid\_port\_scan(H1,H3) \rightarrow LICQ\_remote\_to\_user(H1,H3) \rightarrow local\_setuid\_bof(H3,H3).$

4.3 Attack Path Enumeration Algorithm

Planner generates a single *shortest* path on each run, if *fact.pddl* is properly invalided. However, to generate all possible solutions (*shortest* paths), there is a need for a customized algorithm that will use *planner* as a lower level module. Since each solution is an attack path, a set of all acyclic paths may be obtained. In the present work, a customized *attack path enumeration* algorithm (refer to algorithm 1) is used, which executes *planner* at the low-level by performing automatic invalidation of *fact.pddl*.

```
(define (problem Attack)
    (:domain attackgraph)
    (:objects
        Host0
        Host1
        Host2
        Host3
        Attacker
    )

    (:init
       (= (has_priv  Attacker Attacker) 3)
       (= (has_priv  Attacker Host0) 1)
       (= (has_priv  Attacker Host1) 1)
       (= (has_priv  Attacker Host2) 1)
       (= (has_priv  Attacker Host3) 1)

       (= (root_priv) 3)
       (= (user_priv) 2)
       (= (none_priv) 1)

       (= (port_scan Attacker Host3) 0)
       (= (port_scan_not_done) 0)
       (= (port_scan_done) 1)

       (IIS_web_service Host0)
       (ssh Host1)
       (ftp Host1)
       (rsh Host1)
       (netbios_ssn Host2)
       (squid_proxy Host3)
       (LICQ_chat_service Host3)

       (IIS_bof Host0)
       (ssh_bof Host1)
       (ftp_rhost_overwrite Host1)
       (rsh_login Host1)
       (netbios_ssn_nullsession Host2)
       (LICQ_remote_to_user Host3)
       (local_setuid_bof Host3)

       (IIS_port_connectivity Attacker Host0)

       (ssh_port_connectivity Host0 Host1)
       (ssh_apps_connectivity Host0 Host1)
       (ssh_port_connectivity Host2 Host1)
       (ssh_apps_connectivity Host2 Host1)
       (ssh_port_connectivity Host3 Host1)
       (ssh_apps_connectivity Host3 Host1)

       (ftp_port_connectivity Host0 Host1)
       (ftp_apps_connectivity Host0 Host1)
       (ftp_port_connectivity Host2 Host1)
       (ftp_apps_connectivity Host2 Host1)
       (ftp_port_connectivity Host3 Host1)
       (ftp_apps_connectivity Host3 Host1)
```

```
       (netbios_port_connectivity Host0 Host2)
       (netbios_apps_connectivity Host0 Host2)
       (netbios_port_connectivity Host1 Host2)
       (netbios_apps_connectivity Host1 Host2)

;;     (squid_port_connectivity Host0 Host3)
       (squid_port_connectivity Host1 Host3)
       (squid_port_connectivity Host2 Host3)

;;     (LICQ_port_connectivity Host0 Host3)
       (LICQ_port_connectivity Host1 Host3)
       (LICQ_port_connectivity Host2 Host3)


    )

    (:goal (and(= (has_priv  Attacker Host3) 3)))
)
```

Fig. 5 A modified fact.pddl for the test network (refer to Figure 2)

```
Input: Exploit precedence list, domain.pddl, fact.pddl
Output: An exhaustive set of attack paths Path
while true do
    run planner to obtain path i;
    if no path found then
        while Stopped is not empty do
            service ← Pop(Stopped);
            restart service corresponding to service;
            idx1 ← path to which service belongs;
            idx2 ← index in Critical_idx1 of service;
            if idx2 is not the last index in idx1 then
                idx2 ← idx2 + 1;
                service ← Critical_idx1[idx2];
                stop service corresponding to service;
                Push(service, Stopped);
            end
        end
        if Stopped is empty then
            false_run ← false_run + 1;
            if false_run = 2 then
                print paths from Path;
                Exit;
            end
        end
    end
    if path i already exists then
        service ← Critical_i[0];
        stop service corresponding to service;
        Push(service, Stopped);
    end
    if path i is new then
        add path i to Path;
        restart all stopped services;
        Stopped ← Φ;
        foreach node in path i do
            if service of node has same dependency as any other service then
                Enqueue(node, Marker);
                add node to Critical_i;
            end
        end
        if Marker is not empty then
            node ← Dequeue(Marker);
            stop service of node;
        end
    end
end
```

Algorithm 1. Attack Path Enumeration Algorithm

4.3.1 Exploit Precedence List

Different *IDS* logs and independent sensory data (from an organization's security audits, past attack graph analysis, etc.) have discovered various patterns and modes of cyber attacks which are used to infiltrate an enterprise's critical resources. Typical means by which an attacker penetrates into a network is by executing a chain of *exploits*. Each *exploit* has a set of *preconditions* that needs to be satisfied *conjunctively* for its successful execution [32]. It has been observed that some *exploits* are executed only if some other *exploit* has already been executed. One such example may be *rsh_login* exploit, which can be utilized only if *ftp_rhost_overwrite* exploit is successfully executed on a target machine [34]. Therefore, not every *exploit* can act as the precondition of every other exploit. The concept of logical succession of *exploits* in an *exploit dependency graph* has been used in the present work to generate an *exploit precedence list*. This list contains a set of *exploits* that needs to be executed before a particular *exploit* is accomplished.

4.3.2 Logic Used

The *exploit precedence* information is used to generate all attack paths. This idea rests on the fact that there might be some exploit which has the same *immediate predecessor exploit*, as another exploit in another path. Such a scenario is illustrated in Figure 6. Upon finding any such *exploit (Exploit A)*, which depends on some other *exploit (Immediate predecessor)* which could have other dependent *exploits Exploit B)*, it is logical to shut off the basic network conditions for *Exploit A*. Once such a step is taken, *planner* cannot output any *attack path* involving *Exploit A*. This would provide opportunities for newer paths to be found, since the preceding exploit (here, *Immediate predecessor*) on which some other exploit (corresponds to *Exploit B*) could depend is still active. Each such exploit should be terminated in an iterative fashion to open possibilities of finding newer paths. This would lead to a state, where all the attack paths for the network have been enumerated. However, the assumption, that shutting off an *exploit* will always result in a new *attack path*, is not right. This is due to the fact that *planner* will always output the *shortest path* based on its algorithm. This will most likely be a repeat path instead of being a new path. If a repeat path is found, other *exploits* constituting the repeat path need to be blocked, so that the algorithm can proceed. This resorts to a form of *controlled brute force* method. There could also be scenarios, where no path is found. This is possible, if *exploits* have been blocked in some wrong order. This requires restating of blocked services (corresponding to the *exploit*) in the reverse order of their stopping. This, in essence, is a backtracking algorithm, which helps in escaping from infeasible solutions which might occur as the algorithm proceeds.

4.3.3 Data Structures

Different data structures used for customized *attack path enumeration* algorithm have been summarized in Table 4.

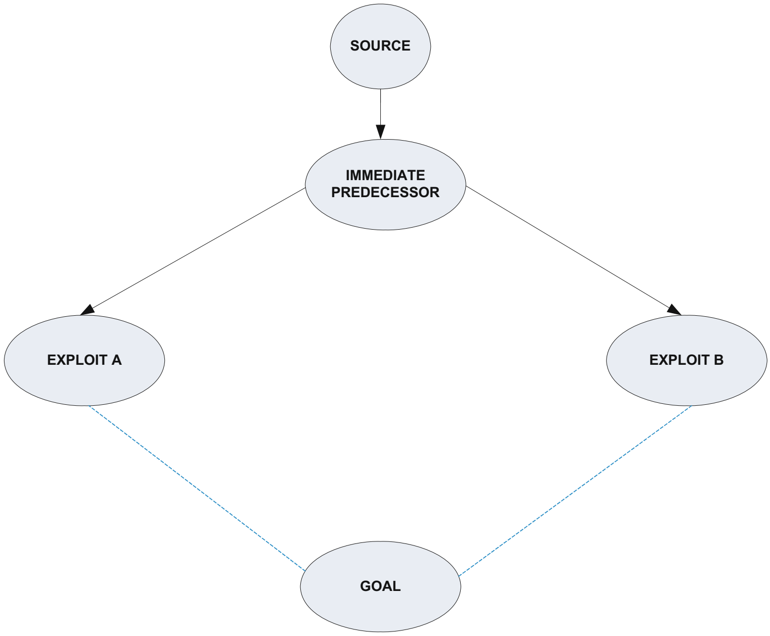


Fig. 6 Exploit precedence logic

Table 4 Data Structures

Marker (*type*: **Queue**)
Marks and enqueues those *exploits/services* which are to be blocked to generate next attack path (based on *exploit precedence list*)
Critical (*type* : **2-D array**)
Stores *exploits* corresponding to each path. Used in situations where *planner* generates a redundant path or no path.
Stopped (*type* : **Stack**)
Tracks which *exploits/services* are currently shut off. Services to be blocked are pushed onto the stack and the ones to be restarted is popped out.
false_run (*type* : **integer**)
Used as a flag to identify condition when no path is generated even after a complete iteration of all possible combinations of network modification has been done.
Path (*type* : **2-D array**)
Stores *shortest attack paths* generated by the algorithm in form of *adjacency matrix*.

The different *minimal attack paths* generated by the customized *attack path enumeration* algorithm (excluding the paths mentioned in section 4.2) are as follows:

1. *Attacker* → IIS_bof(Att,H0) → netbios_ssn_nullsession(H0,H2) → squid_port_scan(H2,H3) → LICQ_remote_to_user(H2,H3) → local_setuid_bof(H3,H3)
2. *Attacker* → IIS_bof(Att,H0) → ftp_rhost_overwrite(H0,H1) → rsh_login(H0,H1) → squid_port_scan(H1,H3) → LICQ_remote_to_user(H1,H3) → local_setuid_bof(H3,H3)
3. *Attacker* → IIS_bof(Att,H0) → ssh_bof(H0,H1) → netbios_ssn_nullsession(H1, H2) → LICQ_remote_to_user(H2,H3) → local_setuid_bof(H3,H3)
4. *Attacker* → IIS_bof(Att,H0) → ftp_rhost_overwrite(H0,H1) → rsh_login(H0,H1) → netbios_ssn_nullsession(H1,H2) → squid_port_scan(H2,H3) → LICQ_remote_to_user(H2,H3) → local_setuid_bof(H3,H3)
5. *Attacker* → IIS_bof(Att,H0) → netbios_ssn_nullsession(H0,H2) → ssh_bof(H2, H1) → squid_port_scan(H1,H3) → LICQ_remote_to_user(H1,H3) → local_setuid_bof(H3,H3)
6. *Attacker* → IIS_bof(Att,H0) → netbios_ssn_nullsession(H0,H2) → ftp_rhost_overwrite(H2,H1) → rsh_login(H2,H1) → squid_port_scan(H1,H3) → LICQ_remote_to_user(H1,H3) → local_setuid_bof(H3,H3)

4.4 Attack Graph Building Algorithm

The *minimal attack paths* obtained by using *attack path enumeration* algorithm (refer to algorithm 1) are collapsed to form the *minimal attack graph*. The attack graph building algorithm takes as input a set of attack paths given by *Path*, a set of *nodes* that constitute the paths, and a *two-dimensional matrix*. The attack graph generation algorithm is given in algorithm 2.

```
Input: A set of attack paths Path, a set of nodes N, a 2-D matrix arr[p][p]
Output: An attack graph
Initialize arr[p][p] = {0} ;
Enumerate each node in N ;
foreach path in Path do
  | foreach valid directed path from node i to node j do
  |   | Set arr[i][j] = 1;
  | end
end
foreach i = 1 to p do
  | foreach j = 1 to p do
  |   | if arr[i][j] = 1 then
  |   |   | Draw a directed edge from i to j;
  |   | end
  | end
end
```

Algorithm 2. Attack Graph Building Algorithm

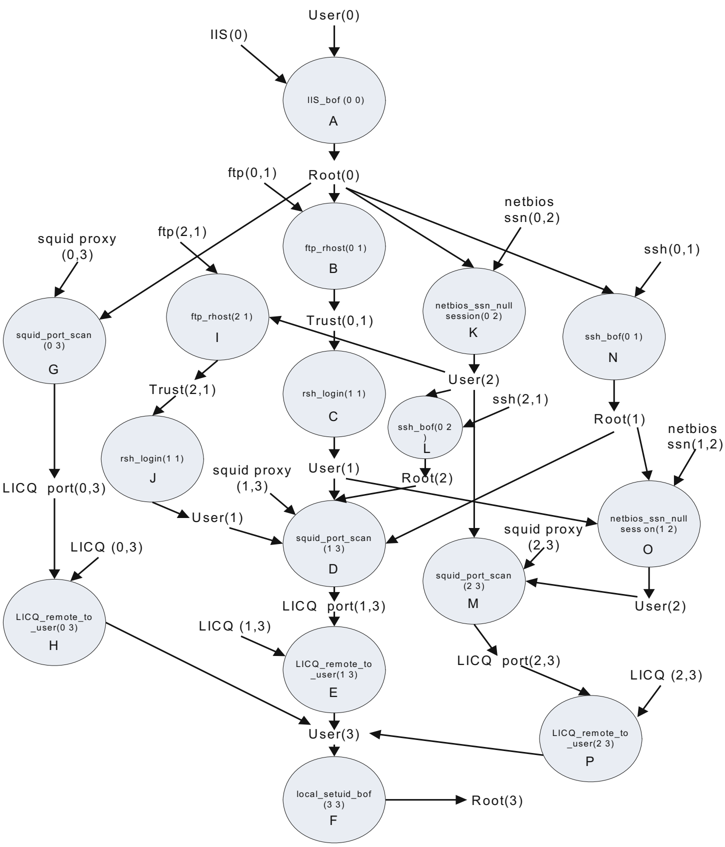


Fig. 7 Attack Graph

Using algorithm 2, the *attack graph* shown in Figure 7 can be generated. The circles represent the *nodes* in the *attack graph* that contain the *exploits*, which the attacker has utilized in different stages of the attack. The texts in the *attack graph* represent the *conditions* obtained by utilizing *exploits* or vice-versa.

4.5 Analysis of the Proposed Approach

In section 2, it has been stated that the time complexity for generating a t–level *plan-graph* at any action–level is $O(mn^k)$, where the notations have their usual meanings. In the *domain.pddl*, it may be noted that we have used only three formal parameters viz. A, S, and T to represent *an attacker, a source, and a destination*, respectively. These three parameters are sufficient to realize any *action*. So, k in this case is a constant. Again, the number of *STRIP* operators that have been used for generating the attack paths is bounded by the number of *generic* vulnerabilities existing in the network. Therefore, the time complexity to generate attack paths in any action–level is $O(mn^3)$, where n is the number of objects used in the *fact.pddl*, i.e., mainly the number of *hosts* in the network and m is the number of *generic* vulnerabilities present in the hosts of the network.

The algorithm for generating the *attack graph* (refer to algorithm 2) is dependent upon the number of *nodes* that constitutes the set of generated attack paths. Again, each node in the *attack graph* represents an *instantiated exploit*. Therefore, the running time of *attack graph building* algorithm is always bounded by $O(e^2)$, where e is the total number of *instantiated exploits*. Hence, the worst-case complexity of generating attack paths and collapsing them into *attack graph* is given as $O(mn^3+e^2)$.

5 Conclusions

Securing a network from intrusions and actual attacks has become a primary concern for a network administrator. Mere isolation of an enterprise network from the Internet is not an option in most of the organizations. Therefore, an administrator needs to analyze the network in such a way, that it becomes sufficiently secure as well as operational. *Attack graph* is a complete graph that describes how multi-stage, multi-host attacks may be correlated to compromise a *goal* node. Such correlated attacks are potentially more harmful than single-point attacks. But, the problem with such a *complete attack graph* is its scalability as well as time-complexity for generation. In this work, a method for enumerating *shortest* attack paths and then, collapsing these paths to form a *minimal attack graph* has been proposed. For this purpose, an *artificial intelligence* technique, called *planner*, has been deployed. The motivation behind using a *planner* as a *model-checker* is its robustness against *state space explosion* problem. Beside this, it provides richer specification language viz. *PDDL* to express complex state space domains relatively easier than custom-built analysis engines. A customized *attack path enumeration* algorithm has been proposed, which does automatic invalidation of *planner's fact* files to generate a succinct set of *shortest attack paths*. These paths are collapsed to form a *shortest attack paths* by a customized *graph building algorithm*. The proposed approach may be extended to wireless network, where generation of attack paths in timely efficient manner is a challenging problem due to its dynamic nature.

References

1. Bugtraq vulnerability database, `http://www.securityfocus.com/archive/` (accessed on June 2009)
2. National vulnerability database. automating vulnerability management, security measurement, and compliance checking, `http://nvd.nist.gov/` (accessed on June 2009)
3. Nessus. open source vulnerability scanner project, `http://www.nesssus.org` (accessed on June 2009)
4. Network associates: Cybercop scanner, `http://www.nai.com` (accessed on June 2009)

5. Nmap, `http://www.insecure.org/nmap/index.html` (accessed on June 2009)
6. Retina. network security scanner, `http://www.eeye.com/html/products/Retina` (accessed on June 2009)
7. Smv information and software on web, `http://www.cs.cmu.edu/~modelcheck` (accessed on June 2009)
8. Ammann, P., Pamula, J., Ritchey, R., Street, J.: A host-based approach to network attack chaining analysis. In: Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC 2005), December 5-9 (2005)
9. Ammann, P., Wijesekera, D., Kaushik, S.: Scalable, graph-based network vulnerability analysis. In: Proceedings of CCS 2002: 9th ACM Conference on Computer and Communications Security, pp. 217–224. ACM Press, New York (2002)
10. Artz, M.: NetSPA: A Network Security Planner. PhD thesis, Massachusettes Institute of Technology (May 2002)
11. Blum, A.L., Furst, M.L.: Fast planning through planning graph analysis. Journal of Artificial Intelligence, 281–300 (February 1997)
12. Chen, Y., Boehm, B., Sheppard, L.: Value driven security threat modeling based on attack path analysis. In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS) (January 2007)
13. Chen, Y., Hsu, C., Wah, B.: Temporal planning using subgoal partitioning and resolution in sgplan. Journal of Artificial Intelligence Research, 323–369 (2006)
14. Dacier, M., Deswarte, Y., Kaaniche, M.: Quantitative assessment of operational security: Models and tools. In: Proceedings of the LAAS Research Report 96493 (May 1996)
15. Dantu, R., Kolan, P., Akl, R., Loper, K.: Classification of attributes and behavior in risk management using bayesian networks. In: Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI), May 23-24, pp. 71–74 (2007)
16. Dawkins, J., Hale, J.: A systematic approach to multi-stage network attack analysis. In: Proceedings of the Second IEEE Internation Information Assurance Workshop (IWIA 2004). IEEE Computer Society, Los Alamitos (2004)
17. Feng, C., Ju-Shu, S.: A flexible approach to measuring network security using attack graphs. In: Proceedings of the International Symposium on Electronic Commerce and Security (ISECS), August 3-5, pp. 426–431 (2008)
18. Fox, M., Long, D.: Pddl 2.1: An extension to pddl for expression temporal planning domains. Journal of Artificial Intelligence Research, 61–124 (2003)
19. Frigault, M., Wang, L.: Measuring network security using bayesian network-based attack graphs. In: Proceedings of the 32nd Annual IEEE International Conference on Computer Software and Applications (COMPSAC 2008), July 28–August 1, pp. 698–703 (2008)
20. Ingols, K., Lippmann, R., Piwowarski, K.: Practical attack graph generation for network defense. In: Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC 2006), December 2006, pp. 121–130 (2006)
21. Jajodia, S., Noel, S., O'Berry, B.: Topological analysis of network attack vulnerability. In: Proceedings of Managing Cyber Threats: Issues, Approaches and Challenges. Kluwer Academic Publisher, Dordrecht (2003)
22. Moskowithz, I.S., Kang, M.H.: An insecurity flow model. In: Proceedings of the 6th New Security Paradigms Workshop, September 1997, pp. 61–74 (1997)
23. Noel, S., Jajodia, S.: Managing attack graph complexity through visual hierarchical aggregation. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security. ACM Press, New York (2004)
24. Noel, S., Jajodia, S., O'Berry, B., Jacobs, M.: Efficient minimum-cost network hardening via exploit dependency graph. In: Proceedings of 19th Annual Computer Security Applications Conference, ACSAC 2003 (2003)

25. Ortalo, R., Deswarte, Y., Kanniche, M.: Experimenting with quantitative evaluation tools for monitoring operational security. In: Proceedings of the IEEE Transactions on Software Engineering, pp. 633–650 (1999)
26. Ou, X., Boyer, W.F., McQueen, M.A.: A scalable approach to attack graph generation. In: Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS), 30 October - 3 November, pp. 336–345 (2006)
27. Ou, X., Govindavajhala, S., Appel, A.W.: Mulval: A logic-based network security analyzer. In: Proceedings of the 14th USENIX Security Symposium, July 31 – August 5 (2005)
28. Phillips, C., Swiler, L.P.: A graph-based system for network-vulnerability analysis. In: Proceedings of the Workshop on New Security Paradigms (NSPW), September 22-26, pp. 71–79 (1998)
29. Ritchey, R., O'Berry, B., Noel, S.: Representing tcp/ip connectivity for topological analysis of network security. In: Proceedings of the 18th Annual Computer Security Applications Conference, ACSAC 2002 (2002)
30. Ritchey, R.W., Ammann, P.: Using model checking to analyze network vulnerabilities. In: Proceedings of the 2000 IEEE Symposium on Security and Privacy, May 2000, pp. 156–165 (2000)
31. Schneier, B.: Attack Trees: Modeling Security Threats. Dr. Dobb's Journal, 21–29 (December 1999)
32. Sheynar, O.: Scenario Graphs and Attack Graphs. PhD thesis, Carnegei Mellon University, USA (April 2004)
33. Sheynar, O., Jha, S., Wing, J.M., Lippmann, R.P., Haines, J.: Automated generation and analysis of attack graphs. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy, pp. 273–284 (2002)
34. Sheynar, O., Wing, J.M.: Tools for generating and analyzing attack graphs. In: Proceedings of the Workshop on Formal Methods for Components and Objects (FMCO), November 2-5 (2004)
35. Swiler, L.P., Phillips, C., Ellis, D., Chakerian, S.: Computer-attack graph generation tool. In: Proceedings of the 2nd DARPA Information Survivability Conference & Exposition (DISCEX II), vol. II, pp. 307–321. IEEE Computer Society, Los Alamitos (2001)
36. Templeton, S., Levitt, K.: A requires/provides model for computer attacks. In: Proceedings of the 2000 Workshop on New Security Paradigms. ACM Press, New York (2001)
37. Tidwell, T., Larson, R., Fitch, K., Hale, J.: Modelling internet attacks. In: Proceedings of the Second Annual IEEE SMC Information Assurance Workshop, pp. 54–59. IEEE Press, Los Alamitos (2001)
38. Tupper, M., Zincir-Heywood, A.N.: Vea-bility security metric: A network security analysis tool. In: Proceedings of the Third International Conference on Availability, Reliability, and Security, March 4-7, pp. 950–957 (2008)
39. Wang, L., Islam, T., Long, T., Singhal, A., Jajodia, S.: An attack graph-based probabilistic security metric. In: Atluri, V. (ed.) DAS 2008. LNCS, vol. 5094, pp. 283–296. Springer, Heidelberg (2008)
40. Wang, L., Singhal, A., Jajodia, S.: Measuring the overall security of network configurations using attack graphs. In: Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec), July 8-11, pp. 98–112 (2007)
41. Wang, Y.M., Liu, Z.L., Cheng, X.Y., Zhang, K.J.: An analysis approach for multi-stage network attacks. In: Proceedings of the 4th International Conference on Machine Learning and Cybernetics (ICMLC), August 18-21 (2005)
42. Zhang, T., Hu, M.Z., Li, D., Sun, L.: An effective method to generate attack graph. In: Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC), August 18-21 (2005)

10

High Dimensional Neural Networks and Applications

B.K. Tripathi and P.K. Kalra

Abstract. Intelligent systems are emerging computing systems developed based on intelligent techniques. These techniques take advantage of artificial neural networks to emulate intelligent behavior. Extensive studies carried out during the past several years have revealed that neural networks enjoy numerous practical advantages over conventional methods. They are more fault-tolerant, less sensitive to noise and mostly used for their human-like characteristics (learning and generalization). They have been accepted as powerful tools for correlating data without making strong assumptions about the problems. Traditional neural networks's parameters are usually real numbers for dealing with real-valued data. However, high-dimensional data also appear in practical applications and consequently, high-dimensional neural networks have been proposed. They have also presented improved results even in case of real-valued problems. As a prelude, we provide a brief overview of the existing methodologies in high-dimensional neural computation. Our particular point of view is to describe several real-world applications, in which the use of these techniques really helps in achieving the goals of intelligent system.

1 Features of Artificial Neuron

A neuron cell in nervous system has very complex structure and extremely complex physiological properties and operations. Artificial neurons, in neural network models, are usually considered as simplified models of biological neurons, i.e., real nerve cells, and the connection weights between nodes resemble to synapses between the neurons. But, for the time being, it is far from clear how much of this simplicity is justified because, as yet, we have only poor understanding of neuronal functions in biological networks. Pre-synaptic signals impinging on the neuron are spread in space and time. Computation performed by a neuron is the

B.K. Tripathi
Research Scholar
107-ACES, Indian Institute of Technology, Kanpur, India
e-mail: abkt@iitk.ac.in

P.K. Kalra
Director
Indian Institute of Technology, Rajasthan, India
e-mail: kalra@iitk.ac.in

D.K. Pratihar and L.C. Jain (Eds.): Intelligent Autonomous Systems, SCI 275, pp. 215–233.
springerlink.com

reflection of spatial and temporal integration of synapses on dendritic tree. This integration of synaptic inputs or information processing is characterized by an aggregation function, as shown in Fig.1. Starting from linear aggregation proposed by McCulloch-Pitts model [1] in 1943 to higher order non-linear aggregation [2, 35, 22], a variety of architectures of the neurons have been proposed in literature. The activation function limits the amplitude of the neuron output to some finite range. Further, these neurons have been extended to deal with higher dimensional signals for applications in broad manifolds of computational intelligence.

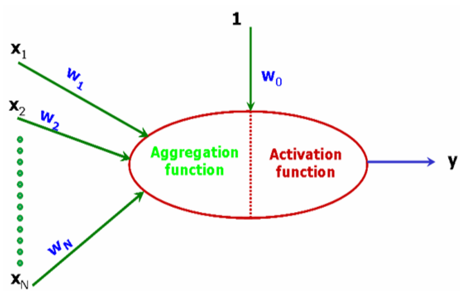


Fig. 1 An artificial Neuron Model

Scientific interest in neuron models or artificial neural networks mainly arises from their potential ability to perform interesting computational tasks. This chapter discusses the functionality of neurons with high dimensional parameters (complex-valued neuron and 3-D vector-valued neuron) and their contribution in design and development of new computational tools to solve high-dimensional problems.

2 Learning and Acquisition of Knowledge

The gradient descent-based error back-propagation method is a very popular learning algorithm for feed-forward neural networks. This conventional back-propagation learning procedure (real-BP) has been extended to 2D BP (complex-BP) in [11, 15] and 3D-BP (three-dimensional vector version of the back-propagation) in [21, 7]. Back-propagation and its variations work by calculating the first order partial derivative of the overall error with respect to each weight. This may be too slow for many applications and it scales up poorly, as the tasks become larger and more complex. To overcome such problems some modifications and variations to this algorithm, like addition of momentum term, Delta-Bar-Delta algorithm [18], modified error function [30, 31] were suggested. A direct adaptive method for faster back-propagation learning was proposed in [5]. Resilient back-propagation (RPROP) is a local adaptive learning scheme, performing supervised batch learning in multilayer neural network. It is basically aimed at eliminating the harmful influence of the size of the partial derivative on the weight step. In RPROP, only the sign of the derivative is considered to indicate the direction of the weight update and the size of the weight change is exclusively determined by a weight-specific update value. The complex-RPROP algorithm can be derived by extending the real RPROP to the complex domain. The results obtained so far are very promising and confirm the quality of the complex-RPROP algorithm with respect to both convergence time and robustness. This chapter explores the high dimensional versions of training algorithms in variety of applications.

3 PCA/ICA

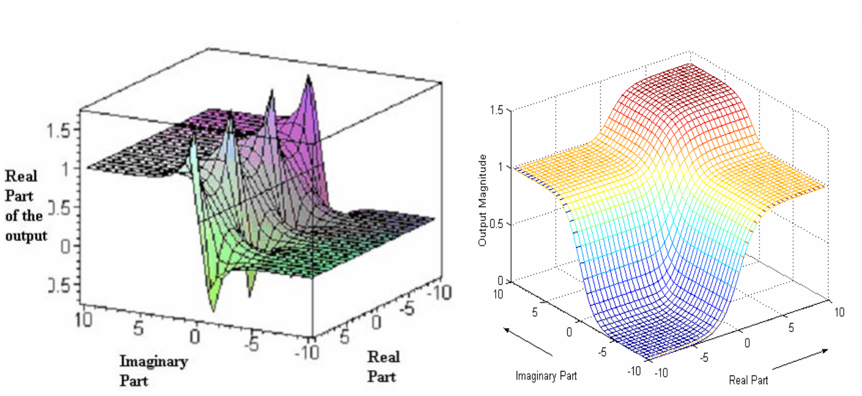
Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are statistical and computational techniques for revealing hidden factors that underlie sets of random variables, measurements or signals. They extract useful information from large amounts of data. PCA is a standard de-correlation technique and following its application, one derives an orthogonal projection basis that directly leads to dimensionality reduction [23]. Its aim is to find a set of mutually orthogonal basis functions that capture the direction of maximum variance in the data and for which, the coefficients are pair-wise de-correlated. PCA is used to describe face images in terms of a set of basis functions (or eigenfaces). ICA is a method of finding a linear non-orthogonal coordinate system in any multivariate data set [24]. The directions of the axes of this coordinate system are determined by both the second and higher order statistics of the original data. The goal is to perform a linear transformation, which makes the resulting variables as statistically independent from each other as possible. High dimensional neural networks can be integrated with these statistical methods to analyze the collected data to recognize reconnaissance patterns.

4 Real Domain Neural Network

In the development of computational intelligent systems, artificial neural networks have been used to build mathematical models that mimic the computing power of the human brain. This powerful processing capability has been demonstrated in various applications of single dimension (real domain). Still there are many applications, which deal with high dimensional signals. The easiest solution would be to consider a conventional real domain neural network, where high dimensional signals are replaced by pairs of independent real-valued signals. An alternative is to introduce a high dimensional neural network. This approach yields more efficient solution both in terms of computational complexity and performance.

5 Complex Domain Neural Network

Complex domain neural network is the extension of real domain neural network, in which all the weights, biases, inputs and outputs are complex numbers [36]. They can directly process two-dimensional information. The complex number is directly related to two-dimensional plane. The complex number comprises of two real numbers and comes with phase information embedded into it. Complex domain neural networks are useful in areas, where there is a need of capturing phase information in signals and must be retained all through the problem. Also, the action potential in human brain may have different pulse patterns and the distance between pulses may be different. This justifies the introduction of complex numbers representing phase and amplitude into neural networks. Complex domain neural networks have been extensively used in processing complex-valued signals and they provide efficient solution for complex-valued problems. By encoding real-valued signals into complex numbers, they have also shown better solution than real-valued neural networks in solving real-valued problems. For example in [17], two-layered complex domain neural networks can successfully solve the XOR problem [9, 22], which cannot be solved by two-layered real-valued neural networks. The complex domain neural network framework attracts not only many neural network researchers but also engineers in general, who work in the field of radars, remote sensing, communication, speech processing, image processing and robotics, mainly because of its high functionality in processing real-world information.



(a) (b)

Fig. 2 Complex-valued activation function : (a) holomorphic functions, (b) split-activation function

5.1 Complex Activation Function

It is noteworthy that there is a wide direction in complex-valued neurons with the different activation functions and there are different complex-valued neural networks. Historically, the first complex-valued activation function for a neuron was a phase dependent activation function and it was proposed in 1971 by Aizenberg et al. [8], which has been later developed as multi-valued neuron (MVN) and universal binary neuron (UBN) in [9, 10]. Some other authors had also addressed various complex-valued neurons with different activation functions. A few of them [14, 11] developed fully complex domain neural network based on holomorphic functions (refer to Fig. 2(a)). It is important to mention that in many cases such networks never converged, the cause might be non-boundedness of holomorphic functions. In the theory of complex-valued functions, Liouville's theorem states that if a function is holomorphic and bounded, then it is a constant function.

To have function bounded in a domain, we cannot keep it holomorphic. Many other researchers [12, 15] had adopted non-holomorphic activation function (refer to Fig. 2(b)), for complex-valued neurons. This is a 2-D extension (split-activation function) of real activation function, $fC(V) = f(\Re(V)) + j * f(\Im(V))$; where f is a non-linear function. Because of simplicity, boundedness and ability to handle amplitude and phase of complex signals properly, this activation function is popularly used in complex domain neural network.

5.2 Learning in Complex Domain

To directly process complex values by artificial neural networks, various gradient-based learning algorithms in complex domain had been developed in [14, 11, 12, 15, 13]. The theoretical aspects of different complex-BP have diverse viewpoints depending on the complex-valued activation functions explained in section 5.1. The operation of activation function in split complex-BP algorithm [12, 15] is split into real and imaginary parts, and this makes the split complex-BP algorithm to avoid the occurrence of singular points in the adaptive training process. In all its applications, this chapter analyzes the dynamics of complex domain neural network with split (real-imaginary type) activation function.

The unit of learning is complex valued signals and learning in complex domain neural network is adjusting 2D motion [19]. With split activation function, the update rules are linear combination of derivatives of real and imaginary components of an activation function. Both real and imaginary parts of the weights are modified as the function of real and imaginary parts of signals, which reduces the probability of standstill [15] in complex-BP as compared to real-BP. Decision boundary of complex neuron consists of two hyper surfaces, which intersect orthogonally and divide a decision region into four sections [16]. Average learning speed of complex BP algorithm is several times more than that of real-BP and number of learning parameters needed is almost half of that of the real-BP [15].

6 Complex Domain Neural Network-Based Intelligent Systems

Intelligent techniques, such as artificial neural networks have emerged as powerful tools for many real-world applications. In recent times, complex domain neural networks have featured prominently in the areas of intelligent system design. These neural networks have been frequently used for solving the problems dealing with 2-D information (complex-valued data). They have also shown their high functionality in case of real-valued problems. We shall not enlist a complex domain neural network to perform arithmetic operations, although we have designed and tested an approximation machine, which can implement basic vector operations (vector addition, subtraction, multiplication and division). Applications in new technologies, such as robotics, manufacturing, communication engineering, space technology, medical instrumentation, ocean engineering, signal and image processing as well as those in older technologies, namely control and prediction problems are creating a wide spectrum of examples, in which non-linearities, uncertainty and complexities play a major role. Furthermore, it is acknowledged that neural networks contribute to the development and spread of intelligent systems for machine-vision applications. Here are just a few examples of problems to which complex domain neural networks have been applied. Neural network-based intelligent systems strongly depend on the existence of technology that provides computers with high computing performance for processing a large amount of information in reasonable time.

6.1 Conformal Mapping

Conformal mapping preserves angles both in magnitude and phase [20], when a problem is mapped from a domain D_z to other domain D_κ in which the solution is sought. As described in [15, 19], the complex-valued signals flowing through complex domain network are the unit of learning, which enable to learn 2-D motion of signals. In contrast, neural network in real domain administers 1-D motion of signals. This is the main reason as to why neural network extended to complex domain can learn mapping Ψ, while equivalent real domain network cannot. One of the novel applications in conformal mapping, which is used to map complicated regions conformally onto simpler, standard regions, where boundary value problems are easier to solve. Bilinear transformation is an important class of conformal mapping which is expressed as the quotient of two linear expressions:

$$\kappa = \Psi = (\mathrm{a\ z + b}) / (\mathrm{c\ z + d}), \qquad (1)$$

where $a,b,c,d \in C$ are complex constant with the restriction that $ab \neq cd$. Such a mapping on complex plane preserves the angles between oriented curves and the phase of each point on the curve is also maintained during transformation. Our experiments confirmed that complex domain neural network has the ability to learn and generalize this transformation with a small error.

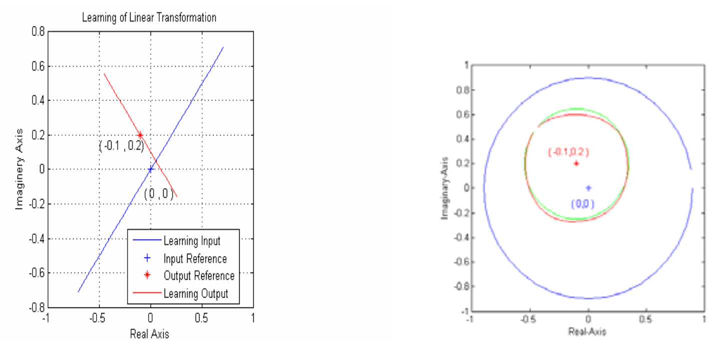
6.1.1 Linear Transformation

If $\Psi(z)$ is an arbitrary bilinear transformation given by Eq. (1), $d = 1$ and $c = 0$, then $\Psi(z)$ reduces to linear transformation (*Scaling, Rotation and Translation*) of the form:

$$\Psi(z) = a\,z + b, \qquad (2)$$

where $z, a, b \in \mathbf{C}$. Evidently, this is an expansion or contraction by a factor $|a|$ and rotation through an angle equal to $Arg(a)$ in counterclockwise direction, followed by translation in a direction defined by the $Arg(b)$ through a distance equal to $|b|$.

Example 1. In order to validate the performance of neural network with two dimensional parameters, we train a 2-3-2 network with complex-BP learning algorithm for mapping Ψ. The input training patterns are set of points of radius vector $r\,e^{j\theta}$, lying on a straight line with equal intervals. Line passes through a reference (mid of line), making an angle θ with real axis and the corresponding output patterns are set of points ($\alpha\, r\, e^{j(\theta+\tau)} + \beta\, e^{j\sigma}$) representing the transformation Ψ, where $a = \alpha\ e^{j\tau}$ and $b = \beta\ e^{j\sigma}$. All z's in training and testing points are within a circle of unit radius centered at origin ($0 \le r \le 1$) and all angles vary from 0 to 2 π. Fig. 3 (a) explains the training input-output mapping ($\alpha = 1/2$, $\tau = 3\pi/4$,$b = -0.1+ j \times 0.2$) and Fig. 3 (b) presents the generalization ability of trained network over circle.



(a) **(b)**

Fig. 3 (a) Learning and (b) Generalization of mapping $\Psi(z)$

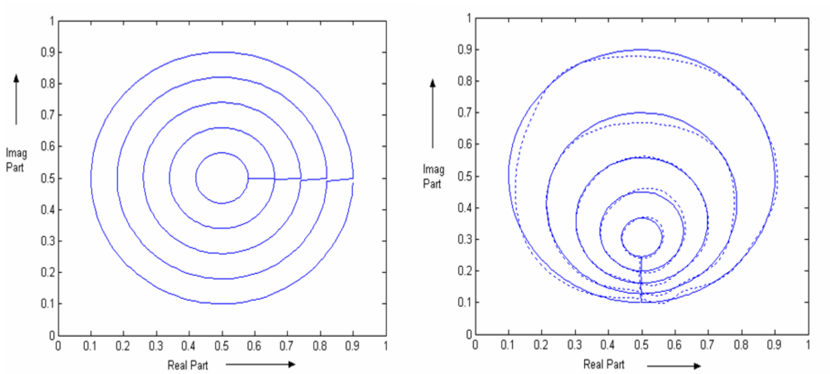
6.1.2 Bilinear Transformation

Mobius transformation (Bilinear transformation) is an important class of elementary mappings studied by Augustus Ferdinand Mobius (1790-1868). These mappings are conveniently expressed as the quotient of two linear expressions and are commonly known as linear fractional or bilinear transformations.

Example 2. Mobius transformation is considered as a linear transformation followed by reciprocal transformation. This conformally maps one-one onto from z plane to k plane. Following mapping maps a disk $Dz : |z| < 1$ one to one and onto the upper half plane $Im(k) > 0$.

$$\kappa = j\times (1 - z) / (1 + z). \tag{3}$$

A 2-5-2 network of complex-GPN neurons proposed in [22] is trained with 180 points (First input) on the circumference of five concentric disk (36 points on each) from radius 0.1 to 0.5 and all disks are referenced at origin (Second input). The output patterns are corresponding values of k defined by Eq. (3). All input-output points in data set are normalized in between 0 to 1. Fig. 4 (a) shows the training input patterns and Fig. 4 (b) displays the desired output (solid lines) and the network output (dotted lines) for training patterns.



(a) (b)

Fig. 4 Bilinear Transformation: (a) Training input data points, (b) Desired output data points (solid lines) and network output (dotted lines) for training data after learning

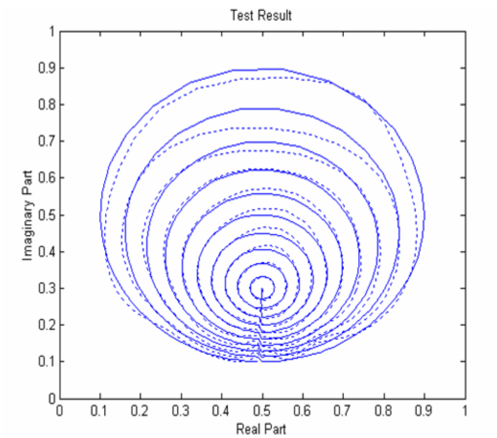


Fig. 5 Bilinear Transformation : Test results, Actual (dotted lines) and Desired (dark lines) outputs

The trained network generalizes this mapping from z plane to k plane over varying values of radius of disks from 0.05 to 0.5 in steps of 0.05 at ten regular intervals. Fig. 5 presents the transformation results of these test patterns with complex GPN- based networks. The dark lines denote the desired output and dotted lines indicate the output of the network. In our experiments, the complex-BP failed to train any network for bilinear transformation, as the saturation is observed in training. However, with complex-RPROP algorithm, training and generalization are successfully achieved.

6.2 Communication Channel Equalization

Neural network equalizers have been recently receiving considerable attention especially in severely non-linear distorted and rapidly varying signals. In high speed communication networks with non-linear active and passive devices, transmission channels and inter symbol interference (ISI) introduce severe non-linear distortions in transmitting symbols. Communication signals with a variable envelope modulation are more efficient in transmission from spectral point of view but they are affected in phase and amplitude by above distortions. The problem of equalization is to determine an estimation of the input signals using received signals and desired delayed signals. Complex domain neural networks are well known for their ability of performing classification tasks by forming non-linear decision boundaries; therefore, they are used in channel equalization.

Example 3. The objective of equalization becomes the separation of the received signals in the output signal space, whose optimal decision region boundaries are generally highly non-linear. An example of highly complex non-linear channel model is suggested in [3, 4], which is as follows:

$$y(n) = O(n) + 0.1\ [O(n)]^2 + 0.05\ [O(n)]^3 + \nu_n \qquad (4)$$

$$\nu_n \sim N(0,0.01),$$

where ν_n is the white Gaussian noise with mean 0 and variance 0.01.

$$O(n) = (\ 0.34 - j\times 0.27)z(n) + (0.87 + j\times 0.43)z(n-1) + (0.34 - j\times 0.21)z(n-2) \qquad (5)$$

6.3 Time-Series Prediction Problems

Neural networks can be used to produce a powerful intelligent system for time series prediction, which is a key problem of function approximation. Various neural network architectures and learning methodologies have been used in literature for efficient solution of these problems. We have proposed multiplicative neuron in [27] and generalized product neuron in [22], which solved variety of time-series problems using single neuron of these models.

6.4 Radar and Sonar Signal Classification

Neural network can distinguish among various types of radar returns (weather, birds, aircraft) with a greater accuracy than that achieved by conventional systems. Ionosphere database was collected by a system in Goose Bay, Labrador and is available in [33]. These radar returns belong to two classes, *Good* radar returns are those showing evidence of some type of structure in the ionosphere and *Bad* returns are those that do not, their signals pass through the ionosphere. The targets were free electrons in the ionosphere. We have used a heterogeneous structure of complex valued neural network [29] containing two conventional *SUM* and one *RBF* neurons in hidden layer and obtained 96% success rate.

A neural network has been used to distinguish between sonar returns from a rock and those from a steel cylinder. Various attempts have been made to solve sonar data problem defined in [32, 28] with the smallest possible network. The best success rate reported using a network of conventional neurons having two such neurons at hidden layer, is 85▷7% [28]. To increase it up to 89▷2%, twenty four hidden neurons are required. Recently, Aizenberg in [9] used 60-2-1 network of multi-valued neurons to solve this problem and reported 88-93% success rate in 400 to 2400 epochs. We have also solved this problem in complex domain network of our newly proposed two neuron models and obtained the success rate of 88-91% using 60-1-1 network in 2500 to 3500 epochs. Thus, our result is obtained using the smallest possible network, which is found to be comparable with that obtained using multi-valued neurons [9].

6.5 2D Face Recognition for Biometric Applications

This section reviews basic issues on neural network-based face recognition systems for biometric interpretation. We, human beings, have natural ability to recognize persons at a glance. Design of artificial intelligent systems is an attempt to simulate this ability in machines. The machine recognition of human faces is a challenging problem due to changes in the face identity, poses, expressions, occuladed and variation among images of the same due to illumination and viewing direction. The issues are how the features are adopted to represent a face under environmental changes and how we classify a new face image based on the chosen representation. Intelligent systems that recognize human faces have been applied in many applications, such as security system, identity verification and access control. Biometrics, in general and facial recognition in particular, are examples of popular applications for artificial intelligent systems.

6.5.1 Biometric System

Neural network-based intelligent systems can provide useful assistance to biometric system for rapid identification in real time. Biometrics can be defined as the automated use of physiological or behavioral characteristics to determine or verify identity. Two main types are involved in the measurement of either physiological or behavioral characteristics. Examples of each measurement type are -

1. Physiological measurements : finger scans, hand scans, retinal scans, iris scans, facial scans.
2. Behavioral measurements : voice scans, keystroke scans and signature scans.

Two main uses of biometrics are in verification and identification. In the verification (one to one), the person claims to be someone and this claim is verified by ensuring the provided biometric sufficiently close to the data stored for that person. Identification (one to many) is a more difficult scenario, in which the person is searched in a database. For the purpose of better security applications or human-computer interactions, the variety of biometrics can be classified into three levels:

• *Level 1* (lowest level of security) - something we have, such as a photo ID.

• *Level 2* (second level of security) - something we know, such as a password to access a computer or a personal identification number (PIN) to access funds at an ATM.

Level 3 (highest level of security) - something we do or something we are, which comprises physiological and/or behavioral biometrics, including facial identity, finger-prints, voice-prints, signatures, etc. For a highly secured system, the combination of these physical properties will give the most realistic biometric system (refer to Fig.6).

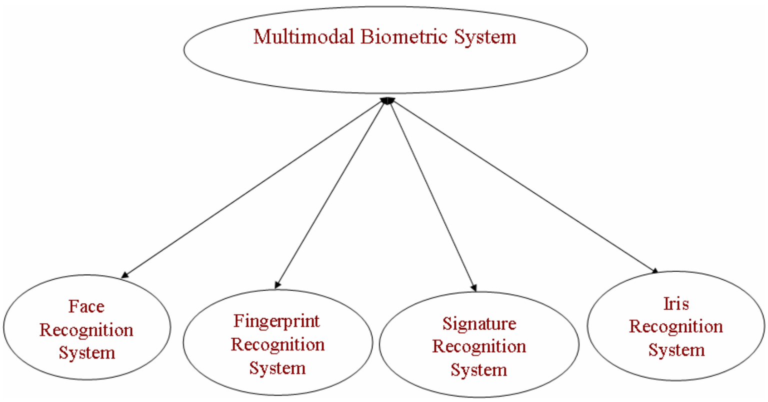


Fig. 6 Diagrammatic representation of multi-modal biometrics systems

6.5.2 Face Recognition

People are good at face recognition. They can identify faces even after years of separation in spite of changes due to viewing condition, aging, expression and distraction (glasses, beards, hairstyle). How are faces encoded or decoded by the human brain? The main challenges are in developing computational model of face recognition, quantization of facial features and extracting relevant features from facial images, then computer may be able of recognizing faces. There are many works emerging every year suggesting different methods for face recognition; most of them follow either of two basic approaches for simulating face recognition in machines:

Feature Extraction-Based Approach
It deals with extracting feature vectors from basic parts of the face (such as eyes, nose, mouth, chin). With the help of deformable templates and extensive mathematics key information from basic parts of face is converted into a feature vector.

Information Theory-Based Approach
Here, the most relevant information that best describe a face is derived from the entire face image. It uses information theory approach for coding and decoding of face images. PCA (M. Turk A. Pentland) and ICA (Bartlett, Movellan, Sejnowski) techniques are popularly used to capture the variation among face images using extracted information from the collection of face images. They also perform dimensionality reduction to allow fast and robust face recognition. Neural networks are used as classifier with emphasis on real time computation and adaptation.

Complex-PCA and Complex-ICA
The development of an intelligent face recognition system requires sufficient information and meaningful data during machine learning of the visual inputs (facial images). This article uses information theory-based approach that search for certain global representation of a face to extract meaningful information. PCA and ICA decompose face images into a small set of characteristic feature images or basis images. Basis image developed by the PCA depends only on second-order image statistics. In face recognition task, much of important information may be contained in the high order relationships among image pixels. Better basis images may be found by a method sensitive to high order statistics : ICA. These methods [23, 24] find a set of basis images, which are used to find a new descriptor of each image. This chapter presents face recognition by complex PCA and complex ICA [25, 26]. The complex PCA and ICA give complex valued new descriptor of each image. Further, intelligent system is implemented with complex domain neural network, whose task is to simulate the function of recognition. Complex domain neural networks perform this task with a much smaller network topology as compared to real domain neural network. The aim of the proposed system is to improve the system's ability to identify faces, while decreasing the storage size of synaptic weights assigned to neural networks. The applications on different standard image databases have verified the enhanced capability of complex domain neural network to make accurate identification. Furthermore, with these techniques it is possible to examine various cases, intensity images, colored, possibly at a distance and thus, minimize the shortcomings of previous techniques.

Example 4. ORL face database [34] has 40 persons and 10 different images for each person, 92x112 pixels with slightly varying lightning conditions, pose, scale, face expression and presence or absence of glasses for gray scale images. Four images of each person are used for training and the rest is for testing. Complex-PCA and -ICA find basis images and complex domain neural networks with three hidden neurons are used as classifier. This system has given 98-99% success rate in 1000 epochs, whereas eight hidden neurons are needed in real domain network to get approximately the same accuracy on an average in 1000 epochs. Since we need a neural network for each person and weights are stored for future testing, complex domain neural networks are preferred, as these are more compact and require much less storage space for processing a large database. It is worth mentioning that the performance with complex-ICA is found to be better than that with complex-PCA.

Example 5. We have conducted experiments on different occluded faces of ORL face database. Fig.7 presents some sets of faces which are recognized by both complex-PCA and -ICA-based recognition method. Fig.8 shows a few occluded images which complex-ICA-based procedure was able to recognize but those were not recognized by complex-PCA.

Fig. 7 Correctly recognized through complex-ICA and complex-PCA

Fig. 8 Correctly recognized through complex-ICA but not complex-PCA

Example 6. IITK database of Indian Faces for colored images of resolution 640x480 with much different orientations have been used in this experiment. Complex-PCA and -ICA find basis images and complex domain neural networks with four hidden neurons are used as the classifier. This system has given 89-94% success rate on an average in 2000 epochs, whereas nine hidden neurons are needed in real domain network to get approximately the same accuracy in 3000 epochs. Complex domain neural networks are seen to be more compact and require much less storage space for future testing.

7 3D Vector Valued Neural Network

In 3D vector valued neural network [21], the parameters like threshold values and input-output signals are 3D real valued vectors and weights associated with connections are 3D orthogonal matrices. All the operations in such neural models are scalar matrix operations. The corresponding 3D vector valued back-propagation algorithm proposed in [7] is a natural extension of complex valued back-propagation algorithm [15, 12] and it has the ability to learn 3D motion [6] as complex-BP learned 2D motion. The net internal potential of a 3D vector-valued neuron is 3D real valued vector, $V = [Vx, Vy, Vz]^T$. The activation function for this neuron is defined as 3D extension of real activation function:

$$Y = f(V^T) = [f(Vx), f(Vy), f(Vz)]^T \tag{6}$$

The back-propagation learning algorithm for conventional real-valued neuron (one dimension) has been extended to higher dimensional neuron models [15, 7]. 3D vector version of back-propagation algorithm for multilayer network is based on scalar product operation among network parameters. The synaptic weights are three dimensional orthogonal matrices. The learning rules have been obtained in [21, 7] using gradient descent on error function, which enables the neuron or network to learn 3D vector patterns naturally by minimizing squared error function -

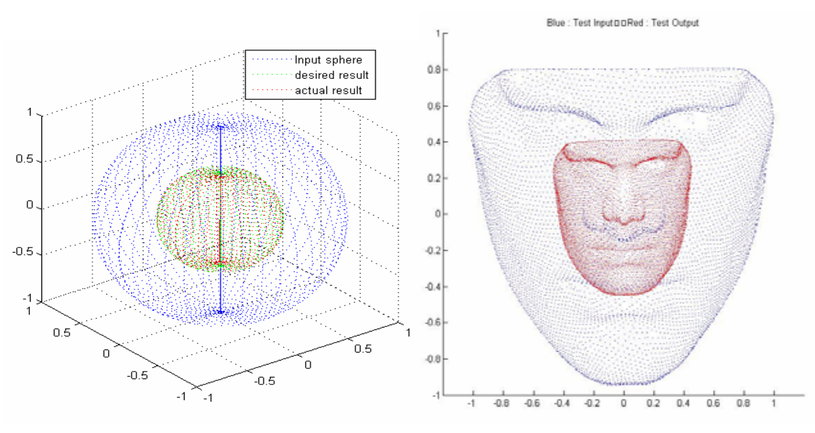
$$E = 1/2\,|e|^2\,, \tag{7}$$

where $|e| = \sqrt{(e_x^2 + e_y^2 + e_z^2)}$ and $e = [e_x, e_y, e_z]^T$. Derivations for updating various learning parameters of multilayered network are given in [15, 7].

7.1 Generalized Mapping in 3D

The three basic classes of transformation: *scaling, rotation and translation* convey dominant geometric characteristics of mapping. In [21], author considered these transformations individually. This article presents mappings for different combinations of above three transformations. We used a 2-4-2 three dimensional vector-valued network, which transforms every input point (x,y,z) into another point (x', y', z') in 3D space. The proposed network generates a value within the range -1≤x', y', z' ≤1. In all simulations, the learning input patterns are the set of points lying on a straight line within a sphere of unit radius, centered at origin and generalization ability is tested over standard geometric figures. First input of input layer takes a set of points lying on surface of curve and second input is a reference point of input curve. Similarly, the first neuron of output layer gives the surface of transformed figure and second output is a reference point of transformed figure.

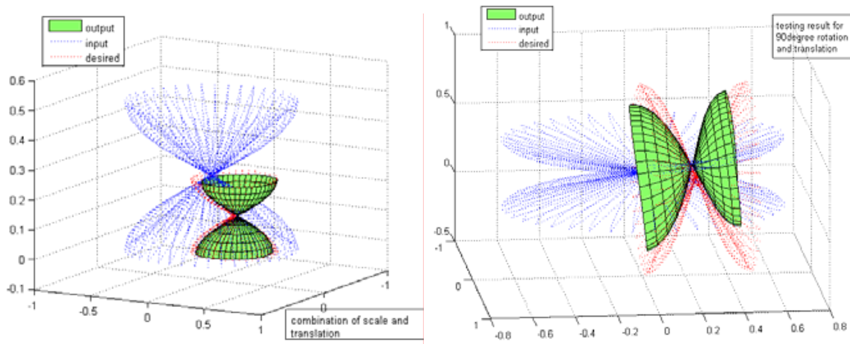
Example 7. In our first experiment, the network is trained for input-output mapping over straight line in 3D for similarity transformation (scaling factor 1◁2). The generalization ability of such trained network is tested over sphere and 3D face of point cloud data, as shown in Fig. 9.



(a) (b)

Fig. 9 Similarity transformation in 3D

Example 8. Fig. 10 (a) presents the generalization behavior of a network, which learned the composition of scaling and translation transformations. All patterns in 3D are contracted by factor 1/2 and displaced by (0,0,0.2).



(a) (b)

Fig. 10 (a) Scaling and Translation in 3D, (b) Rotation and Translation in 3D

Example 9. Fig. 10 (b) presents the generalization behavior of a network, which learned the composition of rotation and translation. All patterns in 3D are rotated over π/2 radians and displaced by (0,0,0.2).

7.2 3-D Face Recognition for Biometric Applications

The face recognition technologies have come a long way in the last twenty years for automatic verification of identity information for secure transactions, surveillance, security task and access control. But, a little attention was given in the design of authentication system, which can process 3D patterns. This section focuses on 3D pattern classification using neural network. In our experiments, we have performed various simulations over point cloud data of the two sets of 3D faces. A 1-2-1 network of vector valued neurons was used in following two experiments.

Example 10. The network was trained by a face displayed in Fig. 11(e), from the first set of face data displayed in Fig. 11. This face data set contains five faces of different persons. All five faces are tested in trained network. Table 1 demonstrates that testing error for four other faces is much higher in comparison with that of the face used for training.

Example 11. The network was trained by a face displayed in Fig. 12(a), from the second set of face data shown in Fig. 12. This face data set contains five faces of the same person with different orientation and poses. All five faces are tested in trained network. The testing error for four other faces is seen to be minimum and comparable to that of the face used for training, as shown in Table 2.

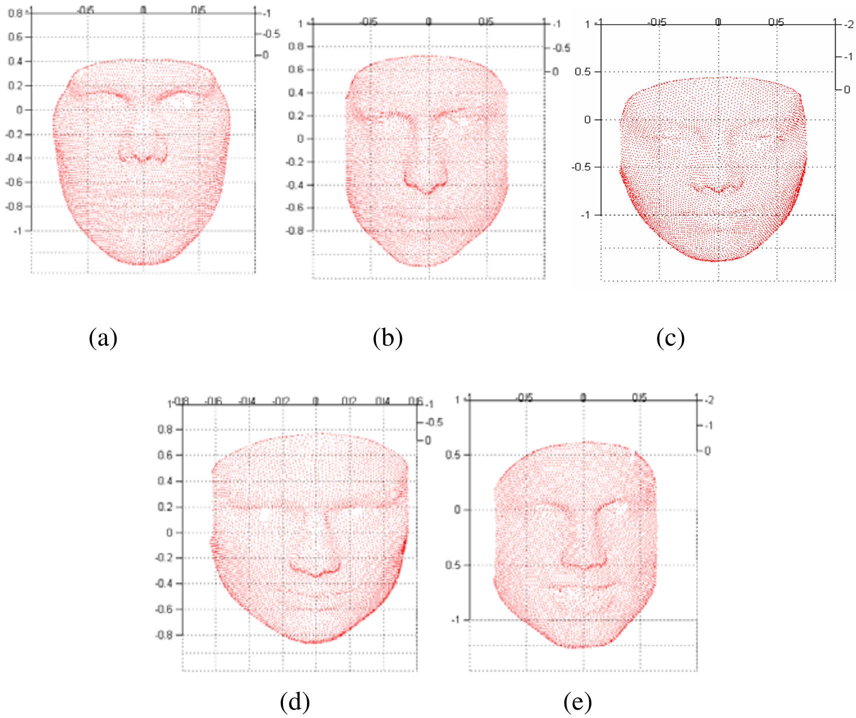


Fig. 11 The set of five different faces.

Table 1 Comparison of testing error of first set (refer to Fig. 11) of face data

Target error, MSE = 0.00005

S. No.	Face	Test Error
1	11 (a)	2.6099×10^{-01}
2	11 (b)	9.4852×10^{-01}
3	11 (c)	2.2289×10^{-01}
4	11 (d)	5.7973×10^{-01}
5	11 (e)	7.7973×10^{-05}

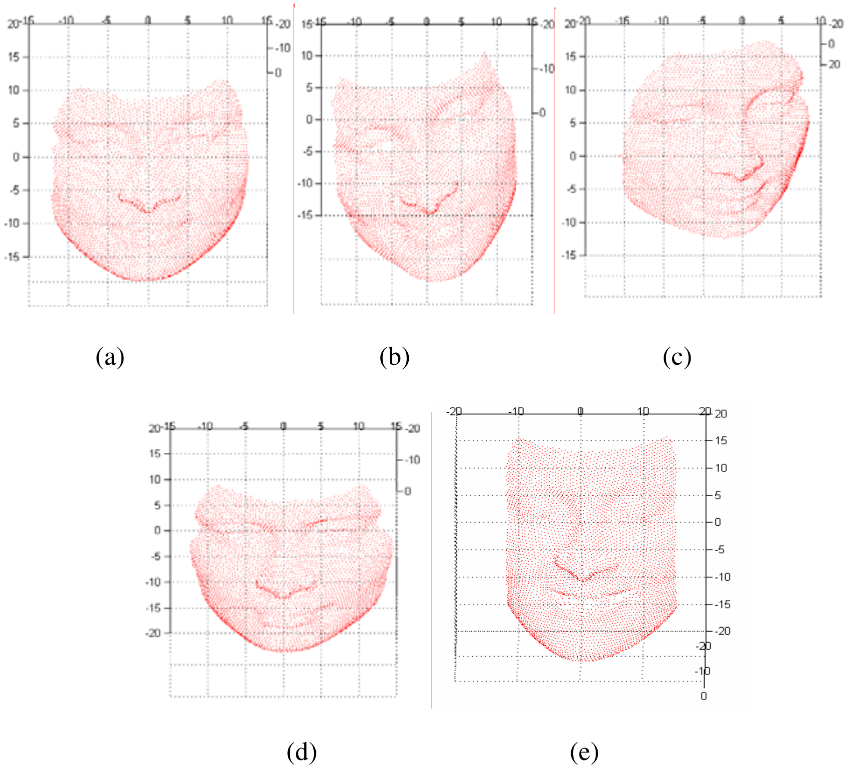


Fig. 12 The set of five faces of same person with different orientation and poses.

Table 2 Comparison of testing performance of second set (refer to Fig. 12) of face data

Target error, MSE = 0.0001

S. No.	Face	Test Error
1	12 (a)	2.9611×10^{-04}
2	12 (b)	6.0055×10^{-03}
3	12 (c)	6.0055×10^{-03}
4	12 (d)	5.50019×10^{-04}
5	12 (e)	5.30099×10^{-04}

8 Conclusions

In this chapter, we have concentrated on describing issues related to the development and use of high dimensional artificial neural network-based intelligent systems for a variety of applications. Research in intelligent systems to-date remains centered on technological issues and is mostly application driven. The successful implementation of intelligent systems should not only rely on their technical feasibility and effectiveness but also on the successful integration of the technology with the organizational and social context within which it is applied. All these issues are critical in applications because they ultimately reflect on the quality of system provided.

References

1. McCulloch, W.S., Pitts, W.: A logical calculation of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133 (1943)
2. Giles, C.L., Maxwell, T.: Learning, invariance and generalization in a high-order neural network. Applied Optics 26(23), 4972–4978 (1987)
3. Cha, H.: Channel equalization using adaptive complex radial basis function networks. IEEE J. Select. Areas Communication 9, 122–131 (1995)
4. Jianping, D., Sundararajan, N., Saratchandran, P.: Communication channel equalization using complex-valued minimal radial basis function neural networks. IEEE Trans. on Neural Networks 13(3), 687–696 (2002)
5. Igel, C., Husken, M.: Empirical evaluation of the improved Rprop learning algorithms. Neurocomputing 50, 105–123 (2003)
6. Watanabe, A., Miyauchi, A., Miyauchi, M.: A method to interpret 3D motion using neural networks. IEICE Trans. on Fundamentals of Electronics and Computer Science E77-A(8), 1363–1370 (1994)
7. Nitta, T.: A 3D vector version of the back-propagation algorithm. In: Proc. of IJCNN 1992, Beijing, vol. 2, pp. 511–516 (1992)
8. Aizenberg, N.N., Ivaskiv, Y.L., Pospelov, D.A.: About one generalization of the threshold function. Doklady Akademii Nauk SSSR (The Reports of the Academy of Sciences of the USSR) 196(6), 1287–1290 (1971) (in Russian)
9. Aizenberg, I., Moraga, C.: Multilayer feed-forward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm. Soft Computing 11(2), 169–183 (2007)
10. Aizenberg, I., Aizenberg, N., Vandewalle, J.: Multi-valued and universal binary neurons: theory, learning, applications. Kluwer Academic Publishers, Dordrecht (2000)
11. Kim, T., Adali, T.: Approximation by fully complex multilayer perceptrons. Neural Computation 15, 1641–1666 (2003)
12. Piazza, F., Benvenuto, N.: On the complex back-propagation algorithm. IEEE Transaction on Signal Processing 40(4), 967–969 (1992)
13. Goh, S.L.: A complex-valued RTRL algorithm for recurrent neural networks. Neural Computation 16, 2699–2713 (2003)
14. Leung, H., Haykin, S.: The complex back-propagation algorithm. IEEE Transaction on Signal Processing 39(9), 2101–2104 (1991)

15. Nitta, T.: An extension of the back-propagation to complex numbers. Neural Networks 10(8), 1391–1415 (1997)
16. Nitta, T.: Orthogonality of decision boundaries in complex-valued neural networks. Neural Computation 16(1), 73–97 (2004)
17. Nitta, T.: Solving the XOR problem and the detection of symmetry using a single complex-valued neuron. Neural Networks 16, 1101–1105 (2003)
18. Jacobs, R.A.: Increased rates of convergence through learning rate adaptation. Neural Networks 1(4), 295–307 (1988)
19. Nitta, T.: An analysis of the fundamental structure of complex-valued neurons. Neural Processing Letters 12, 239–246 (2000)
20. Brown, J.W., Churchill, R.V.: Complex Variables and Applications, 7th edn. McGraw-Hill, New York (2003)
21. Nitta, T.: Three-dimensional vector valued neural network and its generalization abil-Ity. Neural Information Processing – Letters and Reviews 10(10), 237–242 (2006)
22. Tripathi, B.K., Chandra, B., Kalra, P.K.: The generalized product neuron model in complex domain. In: Proc. of International Conference on Neural Information Processing, Auckland, New Zealand (2008)
23. Turk, M., Pentaland, A.P.: Face recognition using eigenfaces. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591 (1991)
24. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face recognition by independent component analysis. IEEE Transaction on Neural Networks 3(6), 1450–1464 (2002)
25. Rattan, S., Hsieh, W., Ruessink, B.G.: Nonlinear complex principal component analysis and its applications. In: Proceedings of International Joint Conference on Neural Networks, Montreal, Canada (2005)
26. Calhoun, V., Adali, T.: Complex ICA for FMRI analysis: performance of several approaches. In: Proc. of ICASSP (2003)
27. Kalra, P.K., Yadav, R.N., John, J.: Time series prediction with single multiplicative neuron model. Applied Soft Computing 7, 1157–1163 (2007)
28. Gorman, R.P., Sejnowski, T.J.: Analysis of hidden units in a layered network trained to classify sonar targets. Neural Networks 1, 75–89 (1988)
29. Tripathi, B.K., Kalra, P.K.: A complex valued heterogeneous neural network. In: Proc. of a Workshop on Complex Valued Neural Network, International Joint Conference on Artificial Intelligence, Hyderabad, India (2007)
30. Van Ooyen, A., Nienhuis, B.: Improving the convergence of the back-propagation algorithm. Neural Networks 5(3), 465–472 (1992)
31. Chen, X., Tang, Z., Vairappan, C., Li, S., Okada, T.: A modified error back-propagation algorithm for complex-value neural networks. International Journal of Neural System 15(6), 435–443 (2005)
32. `http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/neural/bench/cmu`
33. Blake, C.L., Merz, C.J.: UCI Repository of machine learning database. University of California, Department of Information and computer science (1998), `http://www.ics.uci.edu/mealrn/MLRepository.html`
34. ORL database, `http://www.cam-orl.co.uk/facedatabase.html`
35. Gupta, M.M., Homma, N.: Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory. Wiley, Chichester (2003)
36. Hirose, A.: Complex-Valued Neural Networks. Springer, New York (2006)

Author Index

Bepari, Bikash 87
Beyer, Jörg 123
Bhaumik, Subhasis 87
Bibel, Wolfgang 5

Chattopadhyay, Subhagata 143

Ghosh, Nirnay 187
Ghosh, S.K. 187

Hauptmann, Werner 123
Heesche, Kai 123
Hui, Nirmal Baran 29

Jain, Lakhmi C. 1

Kalra, P.K. 167, 215
Kruse, Rudolf 123

Otte, Clemens 123

Pratihar, Dilip Kumar 1, 29, 57, 143

Ray, Ranjit 87

Tripathi, B.K. 215

Vundavilli, Pandu Ranga 57

Yadav, Sandeep Kumar 167

Appendix

The following resources may prove useful for exploring the field of intelligent autonomous systems further. This list is neither complete, nor exclusive. It can be seen as a starting point to explore the field further.

1 Journals

Advanced Engineering Informatics, Elsevier, The Netherlands.
http://www.elsevier.com/wps/find/journaldescription.cws_home/622240/description

AI Magazine, AAAI Press, USA.
http://www.aaai.org/Magazine/magazine.php

Fuzzy Optimization and Decision Making, Springer, USA.
http://www.springer.com/math/journal/10700

IEEE Intelligent Systems, IEEE Press, USA.
http://www.computer.org/portal/site/intelligent/

IEEE Transactions on Neural Networks, IEEE Press, USA.
http://ieee-cis.org/pubs/tnn/

IEEE Transactions on Evolutionary Computing, IEEE Press, USA.
http://www.ieee-cis.org/pubs/tec/

IEEE Transactions on Fuzzy Systems, IEEE Press, USA.
http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=91

IEEE Computational Intelligence Magazine, IEEE Press, USA.
http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=10207

Intelligent Decision Technologies: An International Journal, IOS Press,
The Netherlands.
http://www.iospress.nl/loadtop/load.php?isbn=18724981

International Journal of Hybrid Intelligent Systems, IOS Press, The Netherlands.
http://ijhis.hybridsystem.com/

International Journal of Intelligent and Fuzzy Systems, IOS Press,
The Netherlands.
http://www.iospress.nl/loadtop/load.php?isbn=10641246

International Journal of Knowledge-Based and Intelligent Engineering Systems, IOS Press, The Netherlands.
http://www.iospress.nl/loadtop/load.php?isbn=13272314

Machine Learning, Springer, USA.
http://www.springer.com/computer/artificial/journal/10994

Neural Computing and Applications, Springer, London.
http://www.springer.com/computer/mathematics/journal/521

Neurocomputing, Elsevier, The Netherlands.
http://www.elsevier.com/locate/neucom

International Journal of Intelligent Defence Systems
http://www.inderscience.com/IJIDSS

International Journal of Knowledge and web intelligence
http://www.inderscience.com/ijkwi

International Journal of Advanced Intelligence Paradigms
http://www.inderscience.com/IJAIP

International Journal of Reasoning-based Intelligent Systems
www.inderscience.com/ijris

International Journal of Computational Intelligence Studies
http://www.inderscience.com/ijcistudies

International Journal of Knowledge Engineering and Soft Data Paradigms
http://www.inderscience.com/ijkesdp

International Journal of Artificial Intelligence and Soft Computing
http://www.inderscience.com/ijaisc

International Journal of Granular Computing, Rough Sets and Intelligent Systems
http://www.inderscience.com/sample.php?id=315

International Journal of Innovative Computing, Information and Control
http://www.ijicic.org/

Interactive Technology and Smart Education, Troubador Publishing
http://www.troubador.co.uk/itse/

International Journal of Computational Intelligence Systems
http://www.atlantis-press.com

International Journal of Biomedical Soft Computing and Human Sciences, Biomedical Fuzzy Systems Association, Japan
http://www.f.waseda.jp/watada/BMFSA/journal-IJ/

Multiagents and Grid Systems: An International Journal

Journal of Universal Computer Science

International Journal of Artificial Intelligence, USA

Machine Intelligence and Robotic Control, An International Journal, Cyber Scientific, Japan

Applied Soft Computing, Elsevier Science

International Transactions on Systems Science and Applications

IEEE Transactions on Systems, Man and Cybernetics, Part B, IEEE Press, USA

International Journal of Pattern Recognition and Artificial Intelligence

Computer and Information Systems, Europe

2 Conferences

- KES International Conference Series
  www.kesinternational.org/
- AAAI Conference on Artificial Intelligence
  www.aaai.org/aaai08.php

3 Books

Some books listed in this section may also appear in Section 7 due to their association with the book series.

- Bianchini, M., Maggini, M., Scarselli, F. and Jain, L.C., Innovations in Neural Information Processing Paradigms, Springer-Velag, Germany, 2010.
- Lim, C.P., Jain, L.C. and Satchidananda, D., Innovations in Swarm Intelligence, Springer-Velag, Germany, 2009.
- Tolk, A. and Jain, L.C. (Editors), Complex Systems in Knowledge-based Environments, Springer-Verlag, Germany, 2009.
- Nguyen, N.T. and Jain, L.C. (Editors), Intelligent Agents in the Evolution of Web and Applications, Springer-Verlag, Germany, 2009.
- Jain, L.C. and Nguyen, N.T. (Editors), Knowledge Processing and Decision Making in Agent-Based Systems, Springer-Verlag, Germany, 2009.
- Teodorescu, H.N., Watada, J. and Jain, L.C. (Editors), Intelligent Systems and Technologies, Springer-Verlag, Germany, 2009.
- Nicoletti, M.D.C. and Jain, L.C. (Editors), Computational Intelligence Techniques for Bioprocess Modelling, Supervision and Control, Springer-Verlag, Germany, 2009.
- Castellano, G.; Jain, L.C. and Fanelli, A.M. (Editors), Web Personalization in Intelligent Environments, Springer-Verlag, Germany, 2009.
- Pan, J.S., Huang, H.C. and Jain, L.C. (Editors), Information Hiding and Applications, Springer-Verlag, Germany, 2009.
- Damiani, E., Jeong, J., Howlett, R.J. and Jain, L.C. (Editors), New Directions in Intelligent Interactive Multimedia Systems and Services – 2, Springer-Verlag, Germany, 2009.
- Rakus-Andersson, E., Yager, R.R., Ichalkaranje, N. and Jain, L.C. (Editors), Recent Advances in Decision Making, Springer-Verlag, Germany, 2009.
- Wang, F.H., Pan, J.S. and Jain, L.C., Innovations in Digital Watermarking Techniques, Springer-Verlag, Germany, 2009.
- Finn, A.R. and Jain, L.C. (Editors), Innovation in Defence Support Systems, Springer-Verlag, Germany, 2010.
- Jarvis, J., Ronnquist, R, Jarvis, D. and Jain, L.C., Holonic Execution: A BDI Approach, Springer-Verlag, 2008.

- Pratihar, D.K., Soft Computing, Narosa Publishing House, New Delhi, India, 2008
- Jain, L.C., Sato, M., Virvou, M., Tsihrintzis, G., Balas, V. and Abeynayake, C. (Editors), Computational Intelligence Paradigms: Volume 1 – Innovative Applications, Springer-Verlag, 2008.
- Phillips-Wren, G., Ichalkaranje, N. And Jain, L.C. (Editors), Intelligent Decision Making-An AI-Based Approach, Springer-Verlag, 2008.
- Fulcher, J. and Jain, L.C. (Editors), Computational Intelligence: A Compendium, Springer-Verlag, 2008.
- Sordo, M., Vaidya, S. and Jain, L.C.(Editors), Advanced Computational Intelligence Paradigms in Healthcare 3, Springer-Verlag, 2008.
- Virvou, M. And Jain, L.C.(Editors) , Intelligent Interactive Systems in Knowledge-Based Environments, Springer-Verlag, 2008.
- Sommerer, C., Jain, L.C. and Mignonneau, L. (Editors), The Art and Science of Interface and Interaction Design, Volume 1, Springer-Verlag, 2008.
- Nayak, R., Ichalkaranje, N. and Jain, L.C. (Editors), Evolution of the Web in Artificial Intelligence Environments, Springer-Verlag, 2008.
- Tsihrintzis, G. and Jain, L.C. (Editors), Multimedia Services in Intelligent Environments, Springer-Verlag, 2008.
- Holmes, D. and Jain, L.C. (Editors), Innovations in Bayesian Networks, Springer-Verlag, 2008.
- Magnenat-Thalmann, N., Jain, L.C. and Ichalkaranje, N., New Advances in Virtual Humans, Springer-Verlag, 2008.
- Tsihrintzis, G., Virvou, M., Howlett, R.J. and Jain, L.C. (Editors), New Directions in Intelligent Interactive Multimedia, Springer-Verlag, 2008.
- Jain, L.C., Srinivasan, D. and Palade, V. (Editors), Advances in Evolutionary Computing for system Design, Springer-Verlag, 2007.
- Baba, N., Jain, L.C. and Handa, H. (Editors), Advanced Intelligent Paradigms in Computer Games, Springer-Verlag, 2007.
- Chahl, J.S., Jain, L.C., Mizutani, A. and Sato-Ilic, M. (Editors), Innovations in Intelligent Machines 1, Springer-Verlag, 2007.
- Jain, L.C, Tedman, R. and Tedman, D. (Editors), Evolution of Teaching and Learning in Intelligent Environment, Springer-Verlag, 2007.
- Zharkova, V. and Jain, L.C. (Editors), Artificial Intelligence in Recognition and Classification of Astrophysical and Medical Images, Springer-Verlag, 2007.

- Pan, J-S. Huang, H-C, Jain, L.C. and Fang, W-C (Editors), Intelligent Multimedia Data Hiding, Springer-Verlag, 2007.
- Yoshida, H., Jain, A., Ichalkaranje, A. Jain, L.C. and Ichalkaranje, N.. (Editors), Advanced Computational Intelligence Paradigms in Healthcare 1, Springer-Verlag, 2007.
- Vaidya, S., Jain, L.C. and Yoshida, H. (Editors), Advanced Computational Intelligence Paradigms in Healthcare 2, Springer-Verlag, 2007.
- Sato, M. and Jain, L.C., Innovations in Fuzzy Clustering, Springer-Verlag, 2006.
- Ichalkaranje, N., Ichalkaranje, A. and Jain, L.C., Intelligent Paradigms for Assistive and Preventive Healthcare, Sprinter-Verlag, 2006.
- Patnaik, S., Jain, L.C., Tzafestas, S.G., Resconi, G. and Konar, A. (Editors), Innovations in Robot Mobility and Control, Springer-Verlag, 2006.
- Apolloni, B., Ghosh, A., Alpaslan, F, Jain, L.C. and Patnaik, S. (Editors), Machine Learning and Robot Perception, Springer-Verlag, 2006.
- Palade, V., Bocaniala, C.D. and Jain, L.C. (Editors), Computational Intelligence in Fault Diagnosis , Springer-Verlag, 2006.
- Holmes, D. and Jain, L.C. (Editors), Innovations in Machine Learning, Springer-Verlag, 2006.
- Chen, S.H., Jain, L.C. and Tai, A.C. (Editors), Computational Economics, IGP, USA, 2006.
- Seiffert, U., Jain, L.C. and Schweizer, P. (Editors), Bioinformatics Using Computational Intelligence Paradigms, Springer-Verlag, ISBN 3-54022-901-9, 2005.
- Konar, A. and Jain, L.C., Cognitive Engineering, Springer-Verlag, ISBN 1-85233-975-6, 2005.
- Khosla, R., Ichalkaranje, N. and Jain, L.C. (Editors), Design of Intelligent Multi-Agent Systems, Springer-Verlag, Germany, 2005.
- Ghosh, A. and Jain, L.C. (Editors), Evolutionary Computation in Data Mining, Springer-Verlag, Germany, 2005.
- Phillips-Wren, G. and Jain, L.C. (Editors), Intelligent Decision Support Systems in Agent-Mediated Environments, IOS Press, The Netherlands, 2005.
- Silvermann, B., Jain, A., Ichalkaranje, A. and Jain, L.C. (Editors), Intelligent Paradigms in Healthcare Enterprises, Springer-Verlag, Germany, 2005.

- Ghaoui, C. Bannore, V. Jain, M. and Jain, L.C. (Editors), Knowledge-Based Virtual Education, Springer-Verlag, Germany, 2005.
- Abraham, A., Jain, L.C. and Goldberg, R. (Editors), Evolutionary Multi-objective Optimization, Springer-Verlag, London, 2005
- Pal, N. and Jain, L.C. (Editors), Advanced Techniques in Knowledge Discovery and Data Mining, Springer-Verlag, London, 2005
- Abraham, A., Jain, L.C. and van der van Zwagg, B. (Editors), Innovations in Intelligent Systems, Springer-Verlag, Germany, 2004.
- Tonfoni, G. and Jain, L.C., Visualizing Document Processing, Mouton De Gruyter, Germany, 2004.
- Fulcher, J. and Jain, L.C. (Editors), Applied Intelligent Systems, Springer-Verlag, Germany, 2004.
- Damiani, E., Jain, L.C. and Madravio, M. (Editors), Soft Computing in Software Engineering, Springer-Verlag, Germany, 2004.
- Resconi, G. and Jain, L.C., Intelligent Agents: Theory and Applications, Springer-Verlag, Germany, 2004.
- Pan, J-S., Huang, H.C. and Jain, L.C. (Editors), Intelligent Watermarking Techniques, World Scientific Publishing Company Singapore, 2004.
- Abraham, A., Jain, L.C. and Kacprzyk, J. (Editors), Recent Advances in Intelligent Paradigms and Applications, Springer-Verlag, Germany, 2003.
- Tonfoni, G. and Jain, L.C. (Editors), Innovations in Decision Support Systems, AKI, 2003.
- Shapiro, A. and Jain, L.C.., Intelligent and Other Computational Techniques in Insurance, World Scientific Publishing Company Singapore, 2003.
- Tonfoni, G. and Jain, L.C., The Art and Science of Documentation Management, Intellect, UK, 2003.
- Howlett, R., Ichalkaranje, N., Jain, L.C. and Tonfoni, G. (Editors), Internet-Based Intelligent Information Processing, World Scientific Publishing Company Singapore, 2002.
- Seiffert, U. and Jain, L.C. (Editors), Self-Organising neural Networks, Springer-Verlag, Germany, 2002.
- Jain, L.C. and Kacprzyk, J. (Editors), New Learning Paradigms in Soft Computing, Springer-Verlag, Germany, 2002.
- Jain, L.C., Chen. Z. and Ichalkaranje, N. (Editors), Intelligent Agents and Their Applications, Springer-Verlag, Germany, 2002.

- Jain, L.C., Ichalkaranje, N. and Tonfoni, G. (Editors), Advances in Intelligent Systems for Defence, World Scientific Publishing Company Singapore, 2002.
- Jain, L.C., Howlett, R.J., Ichalkaranje, N., and Tonfoni, G. (Editors), Virtual Environments for Teaching and Learning, World Scientific Publishing Company Singapore, 2002.
- Schmitt, M., Teodorescu, H.N., Jain, A, Jain, A., Jain, S. and Jain, L.C. (Editors). Computational Intelligence Processing in Medical Diagnosis, Springer-Verlag, 2002.
- Jain, L.C. and De Wilde, P. (Editors), Practical Applications of Computational Intelligence Techniques, Kluwer Academic Publishers, USA, 2001.
- Howlett, R.J. and Jain, L.C. (Editors), Radial Basis Function Networks 1, Springer-Verlag, Germany, 2001.
- Howlett, R.J. and Jain, L.C. (Editors), Radial Basis Function Networks 2, Springer-Verlag, Germany, 2001.
- Teodorescu, H.N., Jain, L.C. and Kandel, A. (Editors), Hardware Implementation of Intelligent Systems, Springer-Verlag, Germany, 2001.
- Baba, N. and Jain, L.C. (Editors), Computational Intelligence in Games, Springer-Verlag, 2001.
- Jain, L.C. (Editor), Innovative Teaching and Learning, Springer-Verlag, Germany, 2000.
- Jain, L.C., Lazzerini, B. and Halici, U. (Editors), Innovations in ART Neural Networks, Springer-Verlag, Germany, 2000.
- Jain, A., et al. (Editors), Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis, World Scientific Publishing Company, Singapore, 2000.
- Jain, L.C. and Fanelli, A.M. (Editors), Recent Advances in Artificial Neural Networks: Design and Applications, CRC Press, USA, 2000.
- Medsker, L., and Jain, L.C. (Editors) Recurrent Neural Networks: Design and Applications, CRC Press, USA, 2000.
- Jain, L.C., Halici, U., Hayashi, I., Lee, S.B. and Tsutsui, S. (Editors) Intelligent Biometric Techniques in Fingerprint and Face Recognition, CRC Press, USA, 2000.
- Jain, L.C. (Editor), Evolution of Engineering and Information Systems, CRC Press USA, 2000.
- Dumitrescu, D., Lazzerini, B., Jain, L.C. and Dumitrescu, A., Evolutionary Computing and Applications, CRC Press USA, 2000.

- Lazzerini, B., Dumitrescu, D. and Jain, L.C., Fuzzy Sets and their Applications to Clustering and Training, CRC Press USA, 2000.
- Jain, L.C. and De Silva, C.W. (Editors), Intelligent Adaptive Control, CRC Press USA, 1999.
- Jain, L.C. and Martin, N.M. (Editors), Fusion of Neural Networks, Fuzzy Logic and Evolutionary Computing and their Applications, CRC Press USA, 1999.
- Jain, L.C. and Lazzerini, B., (Editors), Knowledge-Based Intelligent Techniques in Character Recognition, CRC Press USA, 1999.
- Teodorescu, H.N., Kandel, A. and Jain, L.C. (Editors), Soft Computing Techniques in Human Related Science, CRC Press USA, 1999.
- Jain, L.C. and Vemuri, R. (Editors), Industrial Applications of Neural Networks, CRC Press USA, 1998.
- Jain, L.C., Johnson, R.P., Takefuji, Y. and Zadeh, L.A. (Editors), Knowledge-Based Intelligent Techniques in Industry, CRC Press USA, 1998.
- Teodorescu, H.N., Kandel, A. and Jain, L.C. (Editors), Fuzzy and Neuro-fuzzy Systems in Medicine, CRC Press USA, 1998.
- Jain, L.C. and Fukuda, T. (Editors), Soft Computing for Intelligent Robotic Systems, Springer-Verlag, Germany, 1998.
- Jain, L.C. (Editor), Soft Computing Techniques in Knowledge-Based Intelligent Engineering Systems, Springer-Verlag, Germany, 1997.
- Sato, M., Sato, Y. and Jain, L.C., Fuzzy Clustering Models and Applications, Springer-Verlag, Germany, 1997.
- Jain, L.C. and Jain, R.K. (Editors), Hybrid Intelligent Engineering Systems, World Scientific Publishing Company, Singapore, 1997.
- Vonk, E., Jain, L.C. and Johnson, R.P., Automatic Generation of Neural Networks Architecture Using Evolutionary Computing, World Scientific Publishing Company, Singapore, 1997.
- Van Rooij, A., Jain, L.C., and Johnson, R.P., Neural Network Training Using Genetic Algorithms, World Scientific Publishing Company, Singapore, December 1996.

4 Conference Proceedings

- Håkansson, A., Nguyen, N.T., Hartung, R., Howlett, R.J. and Jain, L.C. (Editors), Agents and Multi-Agents Systems: Technologies and Applications, Lecture Notes in Artificial Intelligence, Springer-Verlag, Germany, 2009.
- Velasquez, J.D., Rios, S.A., Howlett, R.J. and Jain, L.C. (Editors), Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, LNAI 5711, Volume 1, Springer-Verlag, Germany, 2009.
- Velasquez, J.D., Rios, S.A., Howlett, R.J. and Jain, L.C. (Editors), Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, LNAI 5712, Volume 2, Springer-Verlag, Germany, 2009.
- Pan, J.S., Chen, Y.W. and Jain, L.C. (Editors), Intelligent Information Hiding and Multimedia Signal Processing, IEEE Computer Society Press, USA, 2009.
- Nakamatsu, K., Phillips-Wren, G., Jain, L.C. and Howlett, R.J. (Editors), New Advances in Intelligent Decision Technologies, Springer-Verlag, 2009.
- Nguyen, N.T., Jo, G.S., Howlett, R.J. and Jain, L.C. (Editors), Agents and Multi-Agents Systems: Technologies and Applications, Lecture Notes in Artificial Intelligence, Springer-Verlag, Germany, 2008.
- Lovrek, I., Howlett, R.J. and Jain, L.C. (Editors), Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, LNAI 5177, Volume 1, KES 2007, Springer-Verlag, Germany, 2008.
- Lovrek, I., Howlett, R.J. and Jain, L.C. (Editors), Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, LNAI 5178, Volume 2, KES 2007, Springer-Verlag, Germany, 2008.
- Lovrek, I., Howlett, R.J. and Jain, L.C. (Editors), Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, LNAI 5179, Volume 3, KES 2007, Springer-Verlag, Germany, 2008.
- Pan, J.S., Niu, X.M., Huang, H.C. and Jain, L.C. (Editors), Intelligent Information Hiding and Multimedia Signal Processing, IEEE Computer Society Press, USA, 2008.
- Jain, L.C., Lingras, P., Klusch, M., Lu, J., Zhang, C., Cercone, N. and Cao, L. (Editors), Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2008, IEEE Computer Society, USA, 2008.

- Jain, L.C., Gini, M., Faltings, B.B., Terano, T., Zhang, C., Cercone, N. and Cao, L. (Editors), Proceedings of the 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2008, IEEE Computer Society, USA, 2008.
- Apolloni, B., Howlett, R.J. and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, Volume 1, LNAI 4692, KES 2007, Springer-Verlag, Germany, 2007.
- Apolloni, B.,Howlett, R.J.and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, Volume 2, LNAI 4693, , KES 2007, Springer-Verlag, Germany, 2007.
- Apolloni, B.,Howlett, R.J.and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, Volume 3, LNAI 4694, KES 2007, Springer-Verlag, Germany, 2007.
- Nguyen, N.T., Grzech, A., Howlett, R.J. and Jain, L.C., Agents and Multi-Agents Systems: Technologies and Applications, Lecture Notes in Artificial Intelligence, Springer-Verlag, Germany, 2007, 2007, 1046 pp., ISBN: 978-3-540-72829-0.
- Liao, B.-H., Pan, J.-S., Jain, L.C., Liao, M., Noda, H. and Ho, A.T.S., Intelligent Information Hiding and Multimedia Signal Processing, Volume 1, IEEE Computer Society Press, USA, 2007. ISBN: 0-7695-2994-1.
- Liao, B.-H., Pan, J.-S., Jain, L.C., Liao, M., Noda, H. and Ho, A.T.S., Intelligent Information Hiding and Multimedia Signal Processing, Volume 2, IEEE Computer Society Press, USA, 2007. ISBN: 0-7695-2994-1.
- Howlett, R.P., Gabrys, B. and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, KES 2006, Springer-Verlag, Germany, Vol. 4251, 2006.
- Howlett, R.P., Gabrys, B. and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, KES 2006, Springer-Verlag, Germany, Vol. 4252, 2006.
- Howlett, R.P., Gabrys, B. and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, KES 2006, Springer-Verlag, Germany, Vol. 4253, 2006.
- Khosla, R., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, KES 2005, Springer-Verlag, Germany, Vol. 3682, 2005.

- Skowron, A., Barthes, P., Jain, L.C., Sun, R., Mahoudeaux, P., Liu, J. and Zhong, N.(Editors), Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Compiegne, France, IEEE Computer Society Press, USA, 2005.
- Khosla, R., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, KES 2005, Springer-Verlag, Germany, Vol. 3683, 2005.
- Khosla, R., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, KES 2005, Springer-Verlag, Germany, Vol. 3684, 2005.
- Khosla, R., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence, KES 2005, Springer-Verlag, Germany, Vol. 3685, 2005.
- Negoita, M., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Engineering Systems, KES 2004, Lecture Notes in Artificial Intelligence, Vol. 3213, Springer, 2004
- Negoita, M., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Engineering Systems, KES 2004, Lecture Notes in Artificial Intelligence, Vol. 3214, Springer-Verlag, 2004
- Negoita, M., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Engineering Systems, KES 2004, Lecture Notes in Artificial Intelligence, Vol. 3215, Springer-Verlag, 2004
- Murase, K., Jain, L.C., Sekiyama, K. and Asakura, T. (Editors), Proceedings of the Fourth International Symposium on Human and Artificial Intelligence Systems, University of Fukui, Japan, 2004.
- Palade, V., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Engineering Systems, Lecture Notes in Artificial Intelligence, Vol. 2773, Springer-Verlag, 2003
- Palade, V., Howlett, R.P., and Jain, L.C. (Editors), Knowledge-Based Intelligent Engineering Systems, Lecture Notes in Artificial Intelligence, Vol. 2774, Springer-Verlag, 2003.
- Damiani, E., Howlett, R.P., Jain, L.C. and Ichalkaranje, N. (Editors), Proceedings of the Fifth International Conference on Knowledge-Based Intelligent Engineering Systems, Volume 1, IOS Press, The Netherlands, 2002.
- Damiani, E., Howlett, R.P., Jain, L.C. and Ichalkaranje, N. (Editors), Proceedings of the Fifth International Conference on Knowledge-Based Intelligent Engineering Systems, Volume 2, IOS Press, The Netherlands, 2002

- Baba, N., Jain, L.C. and Howlett, R.P. (Editors), Proceedings of the Fifth International Conference on Knowledge-Based Intelligent Engineering Systems (KES'2001), Volume 1, IOS Press, The Netherlands, 2001.
- Baba, N., Jain, L.C. and Howlett, R.P. (Editors), Proceedings of the Fifth International Conference on Knowledge-Based Intelligent Engineering Systems (KES'2001), Volume 2, IOS Press, The Netherlands, 2001.
- Howlett, R.P. and Jain, L.C. (Editors), Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems, IEEE Press, USA, 2000. Part 1.
- Howlett, R.P. and Jain, L.C. (Editors), Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems, IEEE Press, USA, 2000. Part 2.
- Jain, L.C. (Editor), Proceedings of the Third International Conference on Knowledge-Based Intelligent Engineering Systems, IEEE Press, USA, 1999.
- Jain, L.C. and Jain, R.K. (Editors), Proceedings of the Second International Conference on Knowledge-Based Intelligent Engineering Systems, Volume 1, IEEE Press, USA, 1998.
- Jain, L.C. and Jain, R.K. (Editors), Proceedings of the Second International Conference on Knowledge-Based Intelligent Engineering Systems, Volume 2, IEEE Press, USA, 1998.
- Jain, L.C. and Jain, R.K. (Editors), Proceedings of the Second International Conference on Knowledge-Based Intelligent Engineering Systems, Volume 3, IEEE Press, USA, 1998.
- Jain, L.C. (Editor), Proceedings of the First International Conference on Knowledge-Based Intelligent Engineering Systems, Volume 1, IEEE Press, USA, 1997.
- Jain, L.C. (Editor), Proceedings of the First International Conference on Knowledge-Based Intelligent Engineering Systems, Volume 2, IEEE Press, USA, 1997.
- Narasimhan, V.L. and Jain, L.C. (Editors), The Proceedings of the Australian and New Zealand Conference on Intelligent Information Systems, IEEE Press, USA, 1996.
- Jain, L.C. (Editor), Electronic Technology Directions Towards 2000, ETD2000, Volume 1, IEEE Computer Society Press, USA, May 1995.
- Jain, L.C. (Editor), Electronic Technology Directions Towards 2000, ETD2000, Volume 2, IEEE Computer Society Press, USA, May 1995.

5 Special Issue of Journals

- Nguyen, N.T., Lim, C.P., Jain, L.C. and Balas, V.E. (Guest Editors), Theoretical Advances and Applications of Intelligent Paradigms, Journal of Intelligent and Fuzzy Systems, IOS Press, Volume 20, Number 1&2, 2009.
- Lovrek, I., Howlett, R.J., Lim, C.P. and Jain, L.C. (Guest Editors), Knowledge-based Intelligent Systems and Their Applications, Journal of Intelligent & Fuzzy Systems, 2009.
- Howlett, R.J., Lovrek, I., Jain, L.C., Lim, C.P. and Gabrys, B., Advances in Design and Application of Neural Networks, Neural Computing and Applications, Springer-Verlag, 2010.
- Lim, C.P., Jain, L.C., Tsihrintzis, G.A., Virvou, M. and Balas, V.E. (Guest Editors), Intelligent Modelling and Data Analysis Techniques, International Journal of Intelligent Defence Support Systems, Volume 2, Number 2, 2009.
- Abeynayake, C., Lim, C.P., Castellano, G. and Jain, L.C., Tsihrintzis, G.A., Virvou, M. and Balas, V.E. (Guest Editors), Special Issue on Selected Papers from the KES2008 Conference, International Journal of Intelligent Defence Support Systems, Volume 2, Number 1, 2009.
- Lim, C.P., Sunde, J. and Jain, L.C. (Guest Editors), Special Issue on Intelligent Knowledge Processing and Decision Making Techniques, International Conference, Intelligent Decision Technologies: An International Journal, Volume 3, Number 2, 2009.
- Lim, C.P., Howlett, R.J., Jain, L.C. and Sato-Ilic (Guest Editors), Special Issue on Advances and Applications of Knowledge-Based Systems, International Journal of Advanced Intelligence Paradigms, Volume 2, Number 1, 2009.
- Jain, L.C., Lim, C.P. and Nguyen, N.T. (Guest Editors), Recent Advances in Intelligent Paradigms Fusion and Their Applications, International Journal of Hybrid Intelligent Systems, Volume 5, Issue 3, 2008.
- Lim, C.P., Jain, L.C., Nguyen, N.T. and Balas, V. (Guest Editors), Advances in Computational Intelligence Paradigms and Applications, An International Journal on Fuzzy Optimization and Decision Making, Springer, Volume 7, Number 3, 2008.
- Abraham, A., Jarvis, D., Jarvis, J. and Jain, L.C. (Guest Editors), Special issue on Innovations in agents: An International Journal on Multiagent and Grid Systems, IOS Press, Volume 4, Issue 4, 2008.

- Abraham, A., Smith, K., Jain, R. and Jain, L.C. (Guest Editors), Special issue on Network and Information Security: A Computational Intelligence Approach, Journal of Network and Computer Applications, Elsevier Publishers, Volume 30, Issue 1, 2007.
- Ghosh, A., Seiffert, U. and Jain, L.C., (Guest Editors), Special Issue on Evolutionary Computation in Bioinformatics, Journal of Intelligent and Fuzzy Systems, IOS Press, The Netherlands, Volume 18, Number 6, 2007.
- Palade, V. and Jain, L.C. (Guest Editors), Practical Applications of Neural Networks, Journal of "Neural Computing and Applications", Springer-Verlag, Germany, Volume 14, No. 2, 2005.
- Abraham, A. and Jain, L.C. (Guest Editors), Computational Intelligence on the Internet, Journal of Network and Computer Applications, Elsevier Publishers, Volume 28, Number 2, 2005.
- Abraham, A., Thomas, J., Sanyal, S. and Jain, L.C. (Guest Editors), Information Assurance and Security, Journal of Universal Computer Science, Volume 11, Issue 1, 2005.
- Abraham, A. and Jain, L.C. (Guest Editors), Special issue on Optimal Knowledge Mining, Journal of Fuzzy Optimization and Decision Making, Kluwer Academic Publishers, Volume 3, Number 2, 2004.
- Palade, V., Ghaoui, C. and Jain, L.C. (Guest Editors), Special issue on Intelligent Instructional Environments, Journal of Interactive Technology and Smart Education,Troubador Publishing Ltd, UK, Volume 1, Issue 3, August 2004.
- Palade, V., Ghaoui, C. and Jain, L.C. (Guest Editors), Special issue on Engineering Applications of Computational Intelligence, Journal of Intelligent and Fuzzy systems, IOS Press, Volume 15, Number 3, 2004.
- Alahakoon, D., Abraham, A. and Jain, L.C. (Guest Editors), Special issue on Neural Networks for Enhanced Intelligence, Neural Computing and Applications, Springer-Verlag, UK, Volume 13, No. 2, June 2004.
- Abraham, A., Jonkar, I., Barakova, E., Jain, R. and Jain, L.C. (Guest Editors), Special issue on Hybrid Neurocomputing, Neurocomputing, Elsevier, The Netherlands, Volume 13, No. 2, June 2004.
- Abraham, A. and Jain, L.C., (Guest Editors), Special issue on Knowledge Engineering, Journal of Intelligent and Fuzzy Systems, The IOS Press, The Netherlands, Volume 14, Number 3, 2003.
- Jain, L.C. (Guest Editor), Special issue on Fusion of Neural Nets, Fuzzy Systems and Genetic Algorithms in Industrial Applications, IEEE Transactions on Industrial Electronics, USA, December 1999.

- De Silva, C. and Jain, L.C. (Guest Editors), Special Issue on Intelligent Electronic Systems, Engineering Applications of Artificial Intelligence, an international journal, USA, January 1998.
- Jain, L.C. (Guest Editor), Special issue on Intelligent Systems: Design and Applications, Journal of Network and Computer Applications (An International Journal published by Academic Press, England). Vol. 2, April 1996.
- Jain, L.C. (Guest Editor), Special issue on Intelligent Systems: Design and Applications, Journal of Network and Computer Applications (An International Journal published by Academic Press, England). Vol. 1, January, 1996.

6 Book Chapters

- Jain, L.C. and Lim, C.P., Advances in Decision Making, Chapter 1, Springer-Verlag, 2009, pp. 1-6.
- Castellano, G., Fanelli, A.M., Torsello, M.A. and Jain, L.C., Innovations in Web Personalization, Chapter 1, Springer-Verlag, 2009, pp. 1-26.
- Khazab, M., Tweedale, J. and Jain, L.C., Dynamic Applications using Multi-agent Systems, Castellano, G., Fanelli, A.M., Torsello, M.A. and Jain, L.C., Innovations in Web Personalization, Chapter 4, Springer-Verlag, 2009, pp. 67-79.
- Rakus-Andersson, E. and Jain, L.C., Computational Intelligence in Medical Decision Making, Chapter 9, Springer-Verlag, 2009, pp. 145-159.
- Jain, L.C., Lim, C.P. and Nguyen, N.T., Innovations in Knowledge Processing and Decision Making in Agent-Based Systems, Springer-Verlag, Germany, 2009, Chapter 1, pp. 1-18.
- Tweedale, J. and Jain, L.C., The Evolution of Intelligent Agents Within the World Wide Web, Springer-Verlag, Germany, 2009, Chapter 1, pp. 1-9.
- Tolk, A. and Jain, L.C., An Introduction to Complex Systems in Knowledge-based Environments, Volume 168, SCI Series, Springer-Verlag, Germany, 2009, pp. 1-5
- Tweedale, J., Bollenbeck, F., Jain, L.C. and Urlings, P., Agent Transport Layer Adaptation System, Chapter 11, SCI 170, Springer-Verlag, Germany, pp. 247-273, 2009.
- Lim, C.P. and Jain, L.C., Advances in Intelligent Methodologies and Techniques, Chapter 1, Springer-Verlag, Germany, pp. 3-28, 2009.
- Nicoletti, M.D.C., Jain, L.C. and Giordano, R.C., Computational Intelligence Techniques as Tools for Bioprocess Modelling, Optimization, Supervision and Control, Chapter 1, Springer-Verlag, Germany, pp. 1-23, 2009.
- Pedrycz, W., Ichalkaranje, N., Phillips-Wren, G., and Jain, L.C., Introduction to Computational Intelligence for Decision Making, Springer-Verlag, 2008, pp. 75-93, Chapter 3.
- Tweedale, J., Ichalkaranje, N., Sioutis, C., Urlings, P. and Jain, L.C., Future Directions: Building a Decision Making Framework using Agent Teams, Springer-Verlag, 2008, pp. 381-402, Chapter 14.

- Virvou, M. and Jain, L.C., Intelligent Interactive Systems in Knowledge-Based Environments: An Introduction, Springer-Verlag, 2008, pp. 1-8, Chapter 1.
- Tsihrintzis, G. and Jain, L.C., An Introduction to Multimedia Services in Intelligent Environments, Springer-Verlag, pp. 1-10, 2008, Chapter 1.
- Jain, L.C. and Lim, C.P., An Introduction to Computational Intelligence Paradigms, Springer-Verlag, pp. 1-15, 2008 Chapter 1.
- Nayak, R. and Jain, L.C., An Introduction to the Evolution of the Web in an Artificial Intelligence Environment, pp. 1-15, 2008, Chapter 1.
- Nayak, R. and Jain, L.C., Innovations in Web Applications using Artificial Intelligence Paradigms, pp. 17-40, 2008, Chapter 2.
- Zharkova, V.V. and Jain, L.C., Introduction to Paper Recognition and Classification in Medical and Astrophysical Images, Springer-Verlag, 2007, pp. 1-18, Chapter 1, ISBN: 10-3-540-47511-7.
- Yoshida, H., Vaidya, S. and Jain, L.C., Introduction to Computational Intelligence in Healthcare, Springer-Verlag, 2007, pp. 1-4, Chapter 1, ISBN: 978-3-540-72374-5.
- Huang, H.C., Pan, J.S., Fang, W.C. and Jain, L.C., An Introduction to Intelligent Multimedia Data Hiding, Springer-Verlag, 2007, Chapter 1, ISBN: 10-3-540-711686.
- Jain, L.C.,Quteishat, A. and Lim, C.P., Intelligent Machines :An Introduction, Springer-Verlag, 2007, pp. 1-9, ISBN : 978-3-540-72695-1
- Jain, L.C., Tan, S.C. and Lim, C.P., Introduction to Evolutionary Computing in System Design, Springer-Verlag, 2007, pp. 1-9, ISBN : 978-3-540-72376-9.
- Castellano, G., Castiello, C., Fanelli, M. and Jain, L.C., Evolutionary Neuro-Fuzzy Systems and Applications, Springer-Verlag, 2007, pp. 11-45, ISBN : 978-3-540-72376-9.
- Do, Q.V, Lozo, P. and Jain, L.C., Vision-Based Autonomous Robot Navigation, in Innovations in Robot Mobility and Control, Springer-Verlag, 2006, pp. 65-103, Chapter 2.
- Tran, C., Abraham, A. and Jain, L., Soft Computing Paradigms and Regression Trees in Decision Support Systems, in Advances in Applied Artificial Intelligence, Idea Group Publishing, 2006, pp. 1-28, Chapter 1.
- Jarvis, B., Jarvis, D. and Jain, L., Teams in Multi-Agent Systems, in IFIP International Federation for Information Processing, Vol. 228, Intelligent Information Processing III, Springer-Verlag, 2006, pp. 1-10, Chapter 1.

- Consoli, A., Tweedale, J. and Jain, L., The Link Between Agent Coordination and Cooperation, in IFIP International Federation for Information Processing, Vol. 228, Intelligent Information Processing III, Springer-Verlag, 2006, pp. 11-19, Chapter 2.
- Bannore, B. and Swierkowski, L. An Iterative Approach to Super Resolution, in IFIP International Federation for Information Processing, Vol. 228, Intelligent Information Processing III, Springer-Verlag, 2006, pp. 473-482.
- Abraham, A. and Jain, L.C., Industry, Evolutionary Multiobjective Optimization, in Springer-Verlag's book, ISBN 1-85233-787-7, 2005, pp. 1-6.
- Wang, F., Jain, L.C. and Pan, J., Genetic Watermarking on Spatial Domain, in Intelligent Watermarking Techniques, World Scientific, 2004, pp. 481-514, Chapter 17.
- Wang, F., Jain, L.C. and Pan, J., Watermark Embedding System based on Visual Cryptography, in Intelligent Watermarking Techniques, World Scientific, 2004, pp. 377-394, Chapter 13.
- Jain, L.C. and Chen, Z., Industry, Artificial Intelligence In, in Encyclopedia of Information Systems, Elsevier Science, USA, 2003, pp. 583-597.
- Teodorescu, H.N. and Jain, L.C., An Introduction to Computational Intelligence in Medical Diagnosis, Springer-Verlag, 2002, Chapter 1, pp. 1-11.
- Konar, A. and Jain, L.C., An Introduction to Computational Intelligence Paradigms, in Practical Applications of Computational Intelligence Techniques, Springer-Verlag, 2001, pp. 1-38, Chapter 1.
- Kuncheva, L.I., Wrench, J., Jain, L.C. and Al-zaidan, A., A Fuzzy Model of Heavy Metal Loadings in Marine Environment, in Soft Computing for Risk Evaluation and Management, Springer-Verlag, 2001, pp. 355-371.
- Tedman, D. and Jain, L.C., An Introduction to Innovative Teaching and Learning, in Teaching and Learning, Springer-Verlag, 2000, pp. 1-30, Chapter 1.
- Filippidis, A., Russo, M. and Jain, L.C., Novel Extension of ART2 in Surface Landmine Detection, Springer-Verlag, 2000, pp.1-25, Chapter 1.
- Jain, L.C. and Lazzerini, B., An Introduction to Handwritten Character and Word Recognition, in Knowledge-Based Intelligent Techniques in Character Recognition, CRC Press, 1999, 3-16.
- Filippidis, A., Jain, L.C. and Martin, N.N., “Computational Intelligence Techniques in Landmine Detection,” in Computing with Words in Information/Intelligent Systems 2, Edited by Zadeh, L. and Kacprzyk, J., Springer-Verlag, Germany, 1999, pp. 586-609.

- Halici, U., Jain, L.C. and Erol, A., Introduction to Fingerprint Recognition, in Intelligent Biometric Techniques in Fingerprint and Face Recognition, CRC Press, 1999, pp.3-34.

- Teodorescu, H.N., Kandel, A. and Jain, L., Fuzzy Logic and Neuro-Fuzzy Systems in Medicine: A historical Perspective, in Fuzzy and Neuro-Fuzzy Systems in Medicine, CRC Press, 1999, pp. 3-16.

- Jain, L.C. and Vemuri, R., An Introduction to Intelligent Systems, in Hybrid Intelligent Engineering Systems, World Scientific, 1997, pp. 1-10, Chapter 1.

- Karr, C. and Jain, L.C., Genetic Learning in Fuzzy Control, in Hybrid Intelligent Engineering Systems, World Scientific, 1997, pp. 69-101, Chapter 4.

- Karr, C. and Jain, L.C., Cases in Geno-Fuzzy Control, in Hybrid Intelligent Engineering Systems, World Scientific, 1997, pp. 103-132, Chapter 5.

- Katayama, R., Kuwata, K. and Jain, L.C., Fusion Technology of Neuro, Fuzzy, GA and Chaos Theory and Applications, in Hybrid Intelligent Engineering Systems, World Scientific, 1997, pp. 167-186, Chapter 7.

- Jain, L.C., Medsker, L.R. and Carr, C., Knowledge-Based Intelligent Systems, in Soft Computing Techniques in Knowledge-Based Intelligent Systems, Springer-Verlag, 1997, pp. 3-14, Chapter 1.

- Babri, H., Chen, L., Saratchandran, P., Mital, D.P., Jain, R.K., Johnson, R.P. and Jain, L.C., Neural Networks Paradigms, in Soft Computing Techniques in Knowledge-Based Intelligent Systems, Springer-Verlag, 1997, pp. 15-43, Chapter 2.

- Jain, L.C., Tikk, D. and Koczy, L.T., Fuzzy Logic in Engineering, in Soft Computing Techniques in Knowledge-Based Intelligent Systems, Springer-Verlag, 1997, pp. 44-70, Chapter 3.

- Tanaka, T. and Jain, L.C., Analogue/Digital Circuit Representation for Design and Trouble Shooting in Intelligent Environment, in Soft Computing Techniques in Knowledge-Based Intelligent Systems, Springer-Verlag, 1997, pp. 227-258, Chapter 7.

- Jain, L.C., Hybrid Intelligent System Design Using Neural Network, Fuzzy Logic and Genetic Algorithms - Part I, Cognizant Communication Corporation USA, 1996, Chapter 9.

- Jain, L.C., Hybrid Intelligent System Applications in Engineering using Neural Network and Fuzzy Logic – Part II, Cognizant Communication Corporation USA,1996, Chapter 10.

- Jain, L.C., Introduction to Knowledge-Based Systems, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 17-27, Chapter 1.
- Jain, L.C. and Allen, G.N., Introduction to Artificial Neural Networks, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 36-62, Chapter 2.
- Jain, L.C. and Karr, C.L., Introduction to Fuzzy Systems, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 93-103, Chapter 3.
- Jain, L.C. and Karr, C.L., Introduction to Evolutionary Computing Techniques, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 121-127, Chapter 4.
- Marwood, W. and Jain, L.C., Microelectronic Engineering, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 227-241, Chapter 6.
- Devanathan, R., Mital, D.P. and Jain, L.C., Control Instrumentation – A survey of Current Practice and Some Future Trends, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 277-293, Chapter 7.
- Marwood, W. and Jain, L.C., Electronic Design and Automation, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 335-341, Chapter 8.
- Sato, M., Jain, L.C. and Takagi, H., Electronic Design and Automation, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 402-430, Chapter 9.
- Furuhashi, T., Takagi, H. and Jain, L.C., Intelligent Systems using Artificial Neural Networks, fuzzy Logic and Genetic Algorithms in Industry, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 485-495, Chapter 10.
- Tanaka-Yamawaki, M. and Jain, L.C., Multimedia, Computer Simulations and Earth station, Electronic Technology Directions to the Year 2000, IEEE Computer Society Press USA, 1995, pp. 462-463, Chapter 11.

7. Book Series

7.1 Advanced Intelligence and Knowledge Processing, Springer-Verlag, Germany.

http://www.springer.com/series/4738

Series Editor: L.C. Jain and X. Wu

- Chen, M. and Mugnier, M.L., Graph-based Knowledge Representation, Springer-Verlag, London, 2009
- Chli, M. and de Wilde, P., Convergence and Knowledge Processing in Multi-agent Systems, Springer-Verlag, London, 2009
- Monekosso, D., et al. (Editors), Intelligent Environments, Springer-Verlag, London, 2009
- Narahari, Y., et al., Game Theoretic Problems in Network Economics and Mechanism Design Solutions, Springer-Verlag, London, 2009
- Zarri, G.P., Representation and Management of Narrative Information, Springer-Verlag, London, 2009
- Zudilova-Seinstra, E. et al. (Editors), Trends in Interactive Visualization, Springer-Verlag, London, 2009
- Camastra, F. and Vinciarelli, A., Machine Learning for audio, Image, and Video Analysis, Springer-Verlag, London, 2008
- Hu, C., et al., Knowledge Processing with Interval and Soft Computing, Springer-Verlag, London, 2008
- Kornai, A., Mathematical Linguistics, Springer-Verlag, London, 2008
- Meisels, A., Distributed Search by Constrained Agents, Springer-Verlag, London, 2008
- Nguyen, N.T., Advanced Methods for Inconsistent Knowledge Management, Springer-Verlag, London, 2008.
- Okada, A., et al. (Editors), Knowledge Cartography, Springer-Verlag, London, 2008
- Prokopenko, M. (Editor), Advances in Applied Self-Organising Systems, Springer-Verlag, London, 2008
- Simovici, D.A. and Djeraba, C., Mathematical Tools for Data Mining, Springer-Verlag, London, 2008
- Pierre, S. (Editor), E-Learning Networked Environments and Architectures, Springer-Verlag, London, 2007

- Scharl, A., Environmental Online Communication, Springer-Verlag, London, 2007
- Karny, M.(Editor), Optimized Bayesian Dynamic Advising, Springer-Verlag, London, 2006
- Liu, S. and Lin, Y., Grey Information: Theory and Practical Applications, Springer-Verlag, London, 2006
- Maloof, M.A. (Editor), Machine Learning and Data Mining for Computer Security, Springer-Verlag, London, 2006
- Manolopoulos, Y., et al., R-Trees: Theory and Applications, Springer-Verlag, London, 2006
- Abraham, A., Jain, L.C. and Goldberg, R. (Editors), Evolutionary Multiobjective Optimization, Springer-Verlag, London, 2005
- Bandyopadhyay, S., et al. (Editors), Advanced Methods for Knowledge Discovery from Complex Data, Springer-Verlag, London, 2005
- Chen-Burger, Y. and Robertson, D., Automatic Business Modelling,, Springer-Verlag, London, 2005
- Fyfe, C., Hebbian Learning and Negative Feedback Networks,, Springer-Verlag, London, 2005
- Grana, M., et al.(Editors), Information Processing with Evolutionary Algorithms, Springer-Verlag, London, 2005
- Husmeier, D., et.al. (Editors), Probabilistic Modelling in Bioinformatics and Medical Informatics, Springer-Verlag, London, 2005
- Konar, A. and Jain, L.C., Cognitive Engineering, Springer-Verlag, London, 2005
- Pal, N. and Jain, L., Advanced Techniques in Knowledge Discovery and Data Mining, Springer-Verlag, London, 2005
- Stuckenschmidt, H. and Harmelen, F.V., Information Sharing on the Semantic Web, Springer-Verlag, London, 2005
- Tan, K.C., et al., Multiobjective Evolutionary Algorithms and Applications, Springer-Verlag, London, 2005
- Wang, J.T.L. et al. (Editors), Data Mining in Bioinformatics, Springer-Verlag, London, 2005
- Wang, L. and Fu, X., Data Mining with Computational Intelligence, Springer-Verlag, London, 2005
- Gomez-Perez, et al., Ontological Engineering, Springer-Verlag, London, 2004

- Ko, C.C., Creating Web-based Laboratories, Springer-Verlag, London, 2004
- Scharl, A., Environmental Online Communication, Springer-Verlag, London, 2004
- Zhang, S., et al., Knowledge Discovery in Multiple Databases, Springer-Verlag, London, 2004
- Mentzas, G., et al., Knowledge Asset Management, Springer-Verlag, London, 2003
- Vazirgiannis, M., et al., Uncertainty Handling and Quality Assessment in Data Mining, Springer-Verlag, London, 2003.

7.2 Advanced Information Processing, Springer-Verlag, Germany.

http://www.springeronline.com/series/4785

Series Editor: L.C. Jain

Books Published

- Ishibuchi, H., Nakashima, T. and Nii, M., Classification and Modeling with Linguistic Information Granules, Springer-Verlag, Germany, 2005.
- Chen, S.H. and Wang, P.P.(Editors), Computational Intelligence in Economics and Finance, Springer-Verlag, Germany, 2004.
- Liu, J. and Daneshmend, L., Spatial Reasoning and Planning, Springer-Verlag, Germany, 2004.
- Ishida, Y., Immunity-Based Systems, Springer-Verlag, 2004.
- Ohsawa, Y. and McBurney, P.(Editors), Chance Discovery, Springer-Verlag, Germany, 2003.
- Deen, S.M.(Editor), Agent-Based Manufacturing, Springer-Verlag, Germany, 2003.
- Gasós J. and Hirsch B., e-Business Applications, Springer-Verlag, Germany, 2003.
- Harris, C., Hong, X. and Gan, Q., Adaptive Modelling, Estimation and Fusion from Data, Springer-Verlag, Germany, 2002.

7.3 Computational Intelligence and its Applications Series, IGI Publishing, USA.

http://www.igi-pub.com/bookseries/details.asp?ID=5&v=bs_releases

Series Editors: L.C. Jain, C. Fyfe, N.T. Nguyen

Books Published

- Chen, S.H., et al. (Editors), Computational Economics: A Perspectice from Computational Intelligence, 2006.
- Begg, R. and Palaniswami, M., Computational Intelligence for Movement Sciences:Neural Networks and Other Emerging Techniques, 2006.
- Zhang, D., et al. Biometric Image Discrimination Technologies, 2006
- Fulcher, J. (Editor), Advances in Applied Artificial Intelligence, 2006

7.4 Knowledge-Based Intelligent Engineering Systems Series, IOS Press, The Netherlands.

http://www.kesinternational.org/bookseries.php

Subseries Editor: L.C. Jain and R.J. Howlett

Books Published

- Masulli, M., et al. (Editors), Computational Intelligence and Bioengineering, IOS Press, The Netherlands, Vol. 196, 2009
- Apolloni, B., Bassis, S. and Marinaro, M. (Editors), New Directions in Neural Networks, IOS Press, The Netherlands, Vol. 193, 2009.
- Virvou, M. and Nakamura, T., Knowledge-Based Software Engineering, IOS Press, The Netherlands, Vol. 180, 2008.
- Velasquez, J.D. and Palade, V., Adaptive Web Sites, IOS Press, The Netherlands, Vol. 170, 2008.
- Zha, X.F. and Howlett, R.J., Integrated Intelligent System for Engineering Design, IOS Press, Vol. 149, 2006.
- Nakamatsu, K. and Abe, J.M. (Editors), Advances in Logic Based Intelligent Systems, IOS Press, Vol. 132, 2005.
- Phillips-Wren, G. and Jain, L. (Editors), Intelligent Decision Support systems in Agent-Mediated Environments, IOS Press, The Netherlands, Vol. 115, 2005.
- Turchetti, C., Stochastic Models of Neural Networks, IOS Press, The Netherlands, Vol. 102, 2004.

- Wang, K., Intelligent Condition Monitoring, IOS Press, Vol. 93, 2003.
- Lee, R.S.T. and Liu, J.H.K., Invariant Object Recognition Based on Elastic Graph Matching, IOS Press, The Netherlands, Vol. 86, 2003.
- Abraham, A., Koppen, M. and Franke, K. (Editors), Design and Applications of Hybrid Intelligent Systems, IOS Press, The Netherlands, Vol. 104, 2002.
- Abraham, A., et al. (Editors), Soft Computing Systems, IOS Press, The Netherlands, Vol. 87, 2002.
- Loia, V. (Editor), Soft Computing Agents, IOS Press, The Netherlands, Vol. 83, 2002.
- Damiani, E., et al. (Editors), Proceedings of the KES2002 Conference, IOS Press, The Netherlands, Vol. 82, Part 1 and Part 2, 2002.
- Motoda, H., Active Mining, IOS Press, The Netherlands, Vol. 79, 2002.
- Namatame, A., et al. (Editors), Agent-Based Approaches in Economic and Social Complex Systems, IOS Press, The Netherlands, Vol. 72, 2002.
- Baba, N., Howlett, R.J. and Jain L.C., Proceedings of the KES2001 Conference, IOS Press, The Netherlands, Vol. 69, Part 1 and Part 2, 2001.

7.5 The CRC Press International Series on Computational Intelligence, The CRC Press, USA.

http://www.crcpress.com/shopping_cart/products/product_series.asp?id=&series=747975&parent_id=&sku=1965&isbn=9780849319655&pc

Series Editor: L.C. Jain

Books Published

- Jiming Liu, Hong Kong Baptist University, Multiagent Robot Systems. ISBN 0-8493-2288-X, 2001
- Marco Russo, University of Messina; and Lakhmi Jain (Editors), Fuzzy Learning and Applications. ISBN 0-8493-2269-3, 2001
- Horia-Nicolai Teodorecu, University of South Florida; and L.C. Jain (Editors), Intelligent Systems and Techniques in Rehabilitation Engineering. ISBN 0-8493-0339-7, 2001
- Iluminada Baturone, A. Barriga, C. Jimenez, D. Lopez and S. Sanchez, Instituto de Microelectronica de Sevilla, Microelectronic Design of Fuzzy Logic-Based Systems. ISBN 0-8493-0633-7, 2000

- Lakhmi Jain; and Anna Maria Fanelli, University of Bari (Editors), Recent Advances in Artificial Neural Networks: Design and Applications. ISBN 0-8493-2268-5, 2000
- Lakhmi Jain, Ugur Halici, Middle East Technical University; I. Hayashi,Hannan University; Sang-Bae Lee, Korea Maritime University; and Shigeyoshi Tsutsui, Hannan University (Editors), Intelligent Biometric Techniques in Fingerprint and Face Recognition. ISBN 0-8493-2055-0, 2000
- Beatrice Lazzerini, University of Pisa; D. Dumitrescu, University of Pisa; and Lakhmi Jain, Fuzzy Sets and their Application to Clustering and Training. ISBN 0-8493-0589-6, 2000
- Beatrice Lazzerini, University of Pisa; D. Dumitrescu, University of Pisa; L.C. Jain; and A. Dumitrescu, University de Savoie, Evolutionary Computing and Applications. ISBN 0-8493-0588-8, 2000
- Larry Medsker, American University Washington; and Lakhmi Jain (Editors), Recurrent Neural Networks: Design and Applications. ISBN 0-8493-7181-3, 2000
- Toyoaki Nishida, Nara Institute of Science and Technology, Dynamic Knowledge Interaction. ISBN 0-8493-0314-1, 2000
- Lakhmi Jain; and Clarance de Silva, University of British Columbia (Editors), Intelligent Adaptive Control: Industrial Applications. ISBN 0-8493-9805-3, 1999
- Lakhmi Jain; and Beatrice Lazzerini, University of Pisa (Editors), Knowledge-Based Intelligent Techniques in Character Recognition. ISBN 0-8493-9807-X, 1999
- Lakhmi Jain; and Noel Martin, University of South Australia (Editors), Fusion of Neural Networks, Fuzzy Systems, and Genetic Algorithms: Industrial Applications. ISBN 0-8493-9804-5, 1999
- Charles Karr, Practical Applications of Computational Intelligence for Adaptive Conrol, University of Alabama (Editors), Industrial Applications of Genetic Algorithms. ISBN 0-8493-9801-0, 1999
- Charles Karr and L.M. Freeman, University of Alabama (Editors), Industrial Applications of Genetic Algorithms. ISBN 0-8493-9801-0, 1999
- Horia-Nicolai Teodorescu, Abraham Kandel, University of South Florida; and Lakhmi Jain (Editors), Soft Computing Techniques in Human-Related Science. ISBN 0-8493-1635-9, 1999
- Lakhmi Jain; Ray Johnson, Defence Science and Technology Organisation; Yoshiharu Takefuji, Keio University; and Lotfi Zadeh, University of California-Berkeley (Editors), Knowledge-Based Intelligent Techniques in Industry. ISBN 0-8493-9803-7, 1998

- Lakhmi Jain; and Rao Vemuri ,University of California-Davis (Editors), Industrial Applications of Neural Networks. ISBN 0-8493-9802-9, 1998
- Horia-Nicolai Teodorescu, Abraham Kandel, University of South Florida; and Lakhmi Jain (Editors), Fuzzy and Neuro-Fuzzy Systems in Medicine. ISBN 0-8493-9806-1, 1998
- Lakhmi Jain (Editor), Evolution of Engineering and Information Systems and their Applications, 1999. ISBN 0-8493-1965-X, 1997
- Zhengxin Chen, University of Nebraska at Omaha, Computational Intelligence for Decision Support. ISBN 0-8493-1799-1
- Horia-Nicolai Teodorescu, Abraham Kandel, University of South Florida, Dynamic Fuzzy Systems and Chaos Applications. ISBN 0-8493-2079-8

7.6 Series on Innovative Intelligence, World Scientific, Singapore.

http://www.worldscientific.com.sg/books/series/sii_series.shtml

Series Editor: L.C. Jain

Books Published

- Vol. 1: Virtual Environments for Teaching and Learning, 2002, Edited by L C Jain, R J Howlett, N S Ichalkaranje and G Tonfoni
- Vol. 2: Advances in Intelligent Systems for Defence, 2002, Edited by L C Jain, N S Ichalkaranje and G Tonfoni
- Vol. 3: Internet-Based Intelligent Information Processing Systems, 2003, Edited by R J Howlett, N S Ichalkaranje, L C Jain and G Tonfoni
- Vol. 4: Neural Networks for Intelligent Signal Processing, 2003, by A Zaknich
- Vol. 5: Complex-Valued Neural Networks, Theories and Applications, 2003, Edited by A Hirose
- Vol. 6: Intelligent and Other Computational Techniques in Insurance, 2003, Theory and Applications, Edited by A F Shapiro and L C Jain
- Vol. 7: Intelligent Watermarking Techniques (With CD-Rom), 2004, Edited by J-S Pan, H-C Huang and L C Jain
- Vol. 8: Learning Support Systems for Organizational Learning, 2004, by Joachim P Hasebrook and Hermann A Maurer

7.7 Intelligent Systems Reference Library

http://www.springer.com/series/8578

Series Editors: J. Kacprzyk and L.C. Jain

Books Published

Vol 1: Computational Intelligence: Collaboration, Fusion and Emergence, 2009, Edited by C. Mumford and L.C. Jain

Vol 2: Tree-Structure based Hybrid Computational Intelligence, 2010
By Y. Chen and A. Abraham

Vol 3: Developments and Challenges for Autonomous Unmanned Vehicles, 2010
By A. Finn and S. Scheding