NEURAL NETWORKS TECHNIQUE FOR THE CONTROL OF ARTIFICIAL MOBILE AGENTS

A PROJECT THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE

REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

In

Mechanical Engineering

By

Pratik Ranjan Bhanjdeo

(111ME0284)

Under Supervision of

Prof. Dayal Ramakrushna Parhi

Department of Mechanical Engineering

National Institute of Technology, Rourkela

CERTIFICATE

National Institute of Technology, Rourkela

This is to certify that the work contained in this thesis, titled "NEURAL NETWORKS TECHNIQUE FOR THE CONTROL OF ARTIFICIAL MOBILE AGENTS" submitted by Pratik Ranjan Bhanjdeo (111ME0284) is an authentic work that has been carried out by him under my supervision and guidance in partial fulfillment for the requirement for the award of Bachelor of Technology Degree in Mechanical Engineering at National Institute of Technology, Rourkela.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/ Institute for the award of any Degree or Diploma.

Place: Rourkela

Date: 11TH MAY, 2015

Dr. Dayal Ramakrushna Parhi

Professor

Department of Mechanical Engineering

National Institute of Technology

Rourkela – 769008

Acknowledgment

I am grateful to The Department of Mechanical Engineering for giving me the opportunity to carry out this project, which is an integral fragment of the curriculum in B. Tech at the National Institute of Technology, Rourkela.

I would like to express my heartfelt gratitude and regards to my project guide, Prof. Dayal Ramakrushna Parhi, Department of Mechanical Engineering, for being the corner stone of the project. It was his incessant motivation and guidance during periods of doubts and uncertainties that has helped me to carry on with this project.

I would like to thank Prof. SibaShankar Mohapatra, Head of the Department, Mechanical Engineering for his guidance, support and direction. I am also obliged to the staff of Mechanical Engineering Department for aiding me during the course of my project. I would also like to offer my sincere thanks to Mr. Anish Pandey from the Department of Mechanical Engineering, for his invaluable guidance throughout the entire project.

Pratik RanjanBhanjdeo

111me0284

ABSTRACT

This project presents a neural network approach to the motion control of an artificial mobile robot. The robot is required to move towards its goal in a cluttered environment and simultaneously avoiding collision with obstacles of any size and being randomly distributed. We have used feed-forward back propagation algorithm to control the robot motion. Our main focus is on the steering action of the robot. The algorithm takes input as angle to target, front obstacle distance, right obstacle distance and left obstacle distance. Sensors have been used for the purpose of getting the inputs on various obstacle distances. The network provides the angle through which the robot is to be steered to avoid obstacles. The steering is achieved by controlling the speed of motors and hence of wheels to whom they are attached to. The viability of the algorithm has been demonstrated through various simulations. The real-world experimental outcomes prove the efficacy of the algorithm to navigate the robot successfully.

Key words: Autonomous mobile robot, feed-forward back propagation algorithm, steering.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW

A mobile robot is an automatic robot that can change its position. Automatic guided vehicles are typical examples of mobile robots. AGVs operate in specially modified environment. They execute various pre-determined tasks along fixed paths. Since an AGV can operate in a pre-structured environment, they are inalterable. Changing the course of tasks is expensive, and any undetermined changes can lead to failure of task. An alternative to such mobile robots is to develop autonomous mobile robots.

The two primary definitions of autonomy are: i) undertaken without external supervision, and b) the faculty of self-government. In order to adapt with undetermined conditions and to cope up to altering environments, the faculty of self-government is necessary. Self-government means that the robot with the help of its own reasoning power can determine its path, instead of following any pre-coded programming sequence.

An autonomous mobile robot has the capability to navigate in its surroundings to execute a variety of activities. It is capable of adapting to changes in its environment. It learns from past experiences and modifies its behavior as per the situation. It can develop internal representations of the environment which will be useful for reasoning processes like navigation.

The faculty of a mobile robot to navigate around autonomously in its environment renders them such applications like: surveillance, inspection, transportation, guidance, etc. Particularly, such robots are utilized to carry out operations in environment that are either hostile or inaccessible to humans. Examples are underwater robots, planetary rovers.

They are used in the areas of cognitive science, psychology and artificial intelligence. They are excellent methods of testing hypotheses about intelligent behavior, perception and reasoning.

In order to navigate effectively consists of the capability to determine one’s position, target location and to find out a path in order to reach the target with minimum effort and time. This planning usually requires a brief knowledge about the environment and the ability to derive meaningful conclusions from it. For a robot to achieve this desired end, it must be able to learn and apply this learning in practical situations. In the current scenario various intelligent systems like fuzzy inference systems, artificial neural networks techniques, genetic algorithms, etc. have been applied for planning involved in the navigation and motion control of an autonomous robot. Artificial neural networks are nonlinear signal processing systems that are developed from interconnected elementary processing devices called neurons.

An Artificial Neural Network (ANN) is an information-processing system that is motivated from the functioning of biological neural networks. The basic unit of this system is the novel structure of the information processing system. It is comprised of a massive number of highly interconnected processing elements (neurons). These neurons work together to solve assigned tasks. ANN, like humans, learns by example. An ANN is developed for respective applications, such as data classification or pattern recognition. Learning in biological systems involves adjustment to the synaptic connections that exist between the neurons. This is true of ANNs as well.

ANN is a kind of artificial intelligence that tries to emulate the functioning of a human brain. Instead of using a digital model, where all calculations interpret ones and zeroes, a neural network functions by relationships between processing elements called the neurons. The output is determined by the structure and weights of various connections. Motivated from the working of biological neural networks, ANN is a massively parallel computing system comprising of an extremely huge number of simple processers with many interconnections. ANN structure tries to utilize some characteristic principles behind the functioning of our brain.

1.2 MOTIVATION

This project is a part of the rapidly and ever-increasing field of Robotics. It requires a combination of different disciplines like computer science, electrical and electronics engineering, mathematics and mechanical engineering. This project works on mobile robots which are different from industrial robots. Industrial robots are generally stationary, work in a controlled environment and have been pre-planned to perform certain activities. But, they have a little role to play in our day-to-day life. In contrast to them, mobile robots can play a significant role in our daily life. In our life things are not pre-planned. In order to cope with such situations it is highly essential to have mobile robots which can move and perform tasks on their own as per the situation. This independent thinking calls for learning and then the application of this learning in practical situations. Such kind of robots will be highly useful in space applications, military operations, hospitals, museums, libraries, super-markets, sports complexes, airports, railway stations, etc. Apart from these examples, we can make significant uses of neural networks approach to identify solutions to problems where no significant patterns may be observed. These algorithms can be used for a variety of purposes like pattern recognition, image processing, bioinformatics, knowledge extraction, forecasting, business applications, healthcare, etc.

1.3 OBJECTIVE

The aim of this project is the design of an autonomous mobile robot. This robot should be able to think and decide its path while navigating in a cluttered environment. It should traverse its path in the minimum possible route and at the same time must avoid collisions with obstacles on its way. It should be able to detect obstacles, plan its path and move toward its goal safely.

1.4 THESIS ORGANIZATION

The present thesis on “NEURAL NETWORKS TECHNIQUE FOR THE CONTROL OF ARTIFICIAL MOBILE AGENTS” has been divided into several chapters. This division has been done so as to learn the concepts easily. Also this division helps in separating theoretical concepts from the practical aspect of the project. The CHAPTER-1 presents a brief idea on mobile robots, artificial neural networks and its application in motion control. There has also been a mention about the motivation behind the project, its objective.

The CHAPTER-2 is about the literature review for this project. It covers the various contributions of different scholarly articles, journals in the development of this project.

The CHAPTER-3 gives a detailed description on mobile robot, its application. There is a significant description on robot architecture as this knowledge will be useful in the design and fabrication of robot.

The CHAPTER-4 focuses on neural networks, their applications, and their classification. There is also a detailed description about various neural networks.

The CHAPTER-5 is about MATLAB, specifically about Neural Network Toolbox. This will be required for simulations.

The CHAPTER-6 gives a brief note on the construction of robot – various parts used and their specifications.

The CHAPTER-7 is about training and simulation experiments. Also we will find a short description on practical training of robot based on simulations.

The CHAPTER-8 is on results obtained while the CHAPTER-9 concludes the thesis. Thereafter we will find a reference section mentioning the sources of knowledge which helped in the development of the robot.

CHAPTER-2

LITERATURE REVIEW

Artificial Neural Network is a kind of artificial intelligence. It tries to emulate the functioning of a human brain. Instead of using a digital model, where all calculations interpret ones and zeroes, it functions through developing relationships among processing units generally called as neurons. The output of the network is determined through the weights determined by activation functions.[3]

Motivated by the way a biological neural network functions, ANN consist of a massive parallel computing systems comprising a huge number of simple processers with many connections in between them. ANN structures try to utilize various characteristic principles on which our brain works.[17]

Neuron is the basic or elemental information-processing unit and is also called as node. This elemental unit of human consciousness engulfs some general faculties. Basically, biological neurons acquire inputs from other sources, integrates them as per a rule, executes a non-linear operation on the result, and then gives the final result as output.[10]

The organization of neurons into various layers and the way they are in and out a layer is generally called as the architecture of the network. There are either no connections or full connections between neurons belonging to a particular layer. The number of layers in the network is defined as the number of layers of weighted interconnections between the particular slabs of neurons. If 2 layers of interconnected weights are present, then there is presence of hidden layers. The various types of network architectures are: Radial basis, Cascaded, General Regression, Back propagation, Elman, etc. The method adopted for determining the weights is the process of training (or learning). The method used to modify the weights in the interconnections between the various layers of network with the aim of getting the desired response is called training of network. A neuron calculates its output response by using activation functions. The output is obtained by applying the total weighted input to an activation function. For neurons in the same layer, same activation functions are used. Activation function may be linear or non-linear.[11]

A single neuron cascaded (SNC) architecture provides compact structures which require less time for execution. Its unique growth in network allows automation in design. This network combines the strengths of multi-layer and single-layer neural networks.[22] The Elman network comprises an input layer, a hidden layer and an output layer. Thus it is similar to a 3-layer feed-forward neural network. Additionally, there is a context layer also.[23] This context layer is fed, without weighing, the output of the hidden layer. Generalized Regression neural network requires only a fraction of the training samples that a back-propagation algorithm would require.[24] A radial basis neural network is an artificial neural network whose activation functions are radial functions.[27] Feed forward was the first and the simplest among the artificial neural networks. In this network, the data passes forward in one direction from the input nodes through the hidden nodes and to the output nodes, only. Cycles and loops are absent. Each neuron in a layer has directed connections to all the neurons of the next layer and vice-versa.[37]. The weights associated with these connections are of varying strengths. Often the units are called as nodes.

The aim of planning of motion is to change the state of the robot by computing a sequence of admissible motions for the robot. A collision free path for a robot is calculated to traverse from the start position to the target avoiding obstacles. The robot has to continuously take in inputs through sensors and thereby using this information guide itself through the cluttered environment. For this action to be accomplished, it should be able to plan its path. Path-planning requires robot to traverse minimum possible route. This may call for certain complex algorithms. The robot should have the ability to maneuver successfully across the obstacles. The selection of algorithm depends on condition of environment, robot architecture, and resources available.[11] Navigation of robot comprises mapping of environment, path planning and path traversal while avoiding collisions.[2] Traditionally, path planning focuses on a single external constraint: physical obstacles in the environment. This is the only constraint considered in path planning. To help in such path planning problems, in recent years, ANN has come out as a powerful approach. It is simple to compute simple and efficient to implement.[4]

For the purpose of simulations a great help comes from the MATLAB Neural Network Toolbox. The toolbox gives a complete set of functions and a graphical user interface required for the design, implementation, visualization, and simulation of neural networks. One can use both supervised training as well as unsupervised training. Also does it support a variety of training functions.[3] The present work uses Feed-Forward Back Propagation Algorithm for network training being simple, fast training and easy to implement. The present method is cheap, easy, and relatively less complex owing to real-like online simulations. The toolbox's modular, open, and extensible design simplifies the creation of customized functions and networks. Various features of this toolbox are: GUI for creating, training, and simulating neural networks; visualization of functions for viewing performance of networks; comprehensive set of training and learning functions; support for the most commonly used supervised and unsupervised network architectures, automatic generation of SIMULINK models from neural network models.

CHAPTER-3

ROBOTS ARCHITECTURE

3.1 BACKGROUND

An autonomous mobile robot is a robot which can move in its workspace in order to execute the assigned activities. Its ability to learn from past experience helps it to take appropriate decisions depending on the situations. The two primary definitions of autonomy: i) work without external control, and b) self-government. In order to cope up undetermined situations and to adapt to altering environments, the faculty of self-government is required. Self-government means that the robot can take decisions of its own and change its actions as per the situations rather than following a fixed set of instructions supplied by the user as in industrial robots.

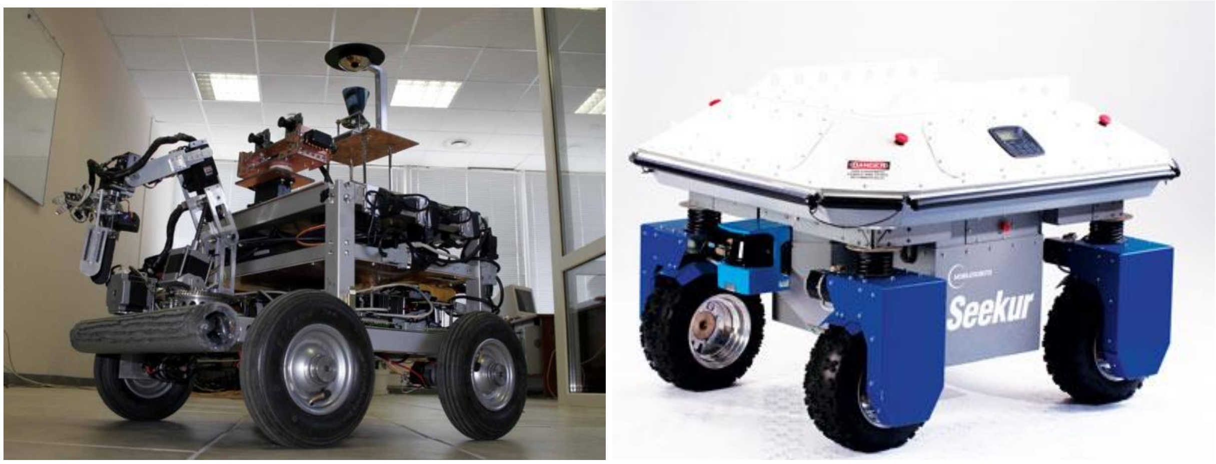


Figure1. Examples of Autonomous Mobile Robots

3.2 APPLICAIONS OF MOBILE ROBOTS

The various uses of an autonomous robot are mentioned as follows

1. In moving items in hospitals.
2. Warehouses uses.
3. Military purposes.
4. Space applications.

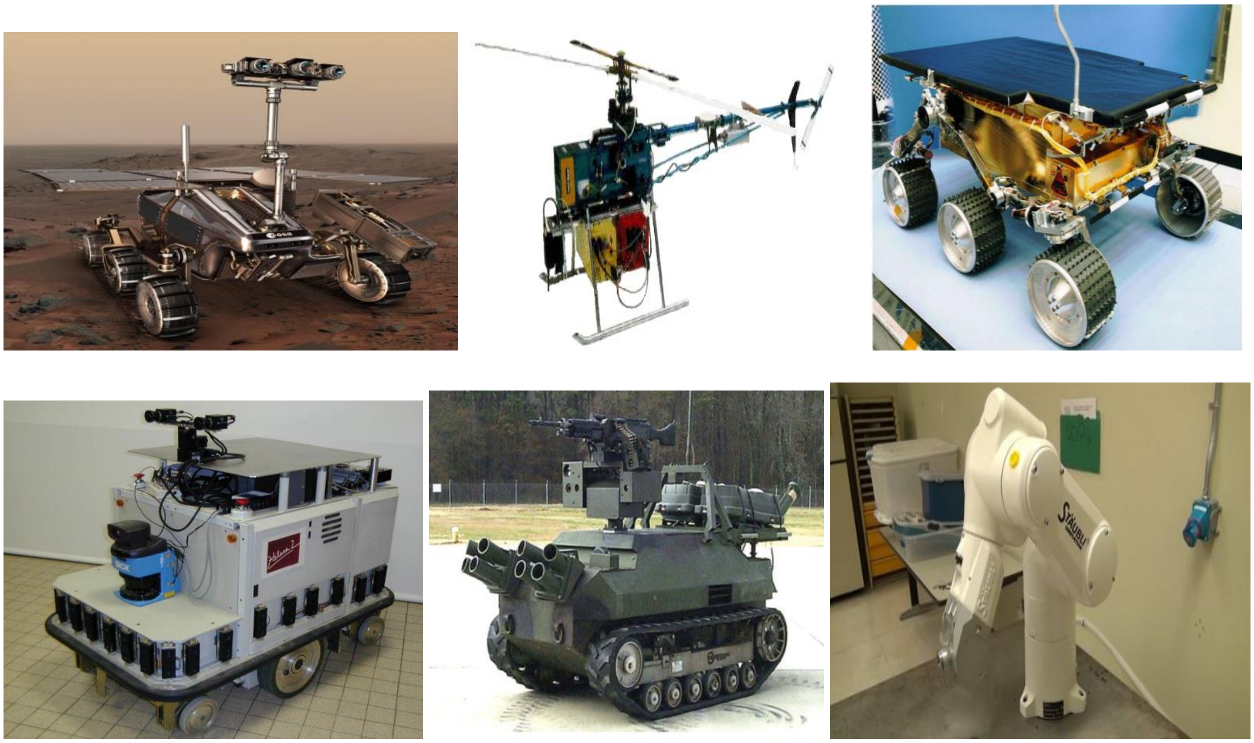


Figure2. Various applications of autonomous mobile robots

3.3 FEATURES OF MOBILE ROBOTS

A fully autonomous robot can:

- Work for long period of time without human interference.
- Avoid situations that are harmful.
- Gain information about the environment.
- Move either all parts or some parts depending upon surrounding situations without human help.

An autonomous robot may also learn new skills like new methods of executing its tasks or accommodating to changing environment.

Mobile robots need to answer three basic questions:

1. Where is it?
2. Where is it going?
3. How should it go there?

In order to answer these questions, a robot must perform the following:

- Make measurements
- Model the environment
- Localize itself
- Plan a path to its goal.

3.4 ROBOT ARCHITECTURE

In order to accomplish the above mentioned characteristics, we first need to look in the robot construction.

The figure below shows the robot architecture.

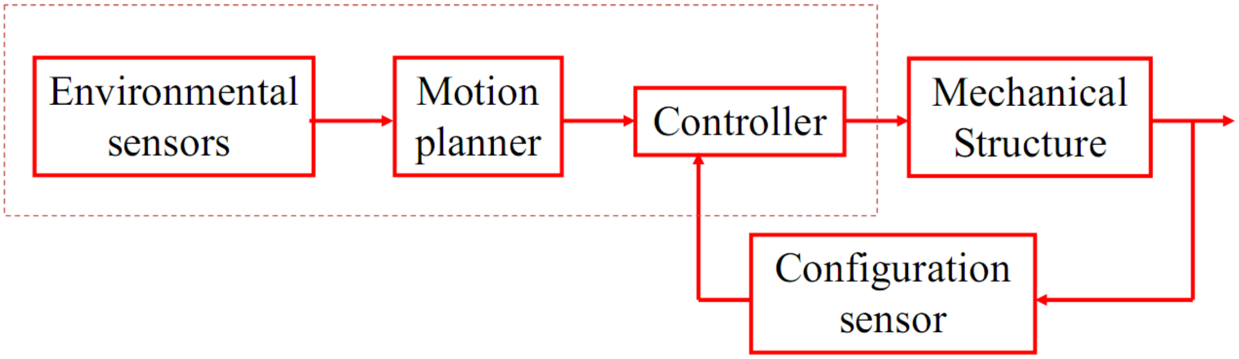


Figure3. Robot architecture

The essential parts are:

- Mechanical Structure – Kinematics & Dynamics
- Sensors
- Actuators
- Controllers
- Power conversion units.
- Programming Languages

3.4.1 SENSORS

Sensors are equipment that are used to measure certain properties of the surroundings, such as temperature, velocity, weight, size, distance, etc. Sensors are designed measure respective physical properties it has been designed to measure. These input data has a meaningful relationship with the properties of the surrounding. All sensors are characterized by certain properties describing their capabilities. Some of them are:

1) Accuracy
2) Linearity
3) Repeatability
4) Measurement Range
5) Resolution: smallest observable increment in output.
6) Response time
7) Sensitivity
8) Type of output.

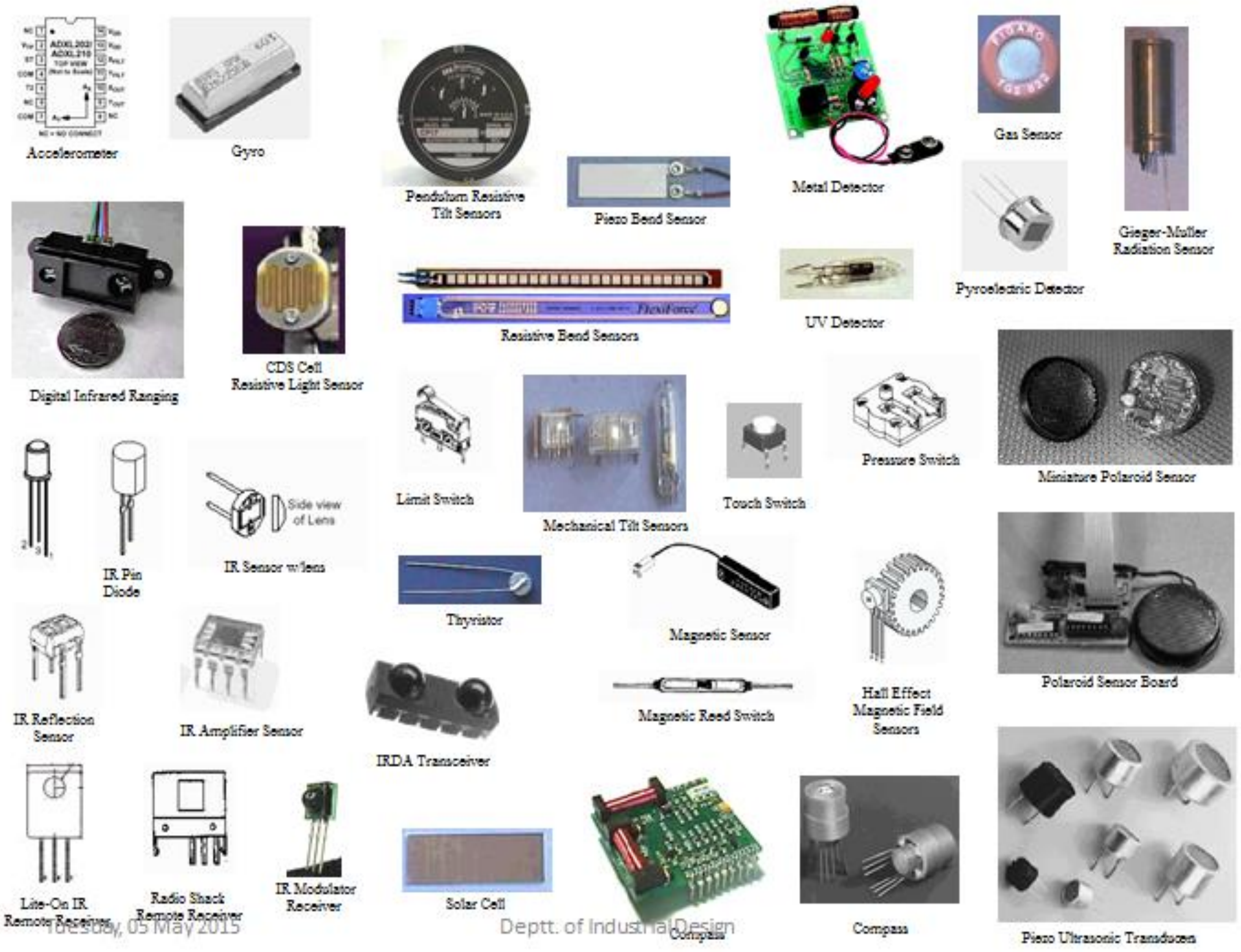


Figure4. Various types of sensors

Some important types of sensors are explained below.

Tactile Sensors

These sensors being designed on the sense of touch can detect physical properties caused due to contact with an object. The simplest tactile sensor is a micro switch.

Infrared Sensors

These sensors are a type of non-contact sensors. They are extensively utilized in mobile robots to detect obstacles. Their operation is: emit an infrared ray, and receiving the rays reflected from the obstacles. The robot-object distance is measured as per the rule– the intensity of the reflected light is inversely proportional to the square of the distance. Hence these sensors are used to measure short distances.

Ultrasonic Sensors

The fundamental principle of ultrasonic sensor is: a pulse of a particular frequency is emitted, reflected off surfaces and then detected by a receiver. By using the time elapsed and the speed of sound we can calculate the distance between the robot and the object.

Laser Sensors

Laser sensors are commonly used in mobile robots to measure the velocity, acceleration and distance of detected objects. The fundamental principle of this sensor is similar to a sonar sensor: a short light pulse is released, the time difference between emitted and detected pulse is used to calculate the distance. Typical maximum range of commercially available sensor may be of several hundred metres. As per requirements one may choose a shorter range one.

3.4.2 ACTUATORS

Actuators are mechanical devices that convert respective input energy into a mechanical motion. These forms of inputs can be either in the form of electrical energy, pneumatics, and hydraulic energy. The different types of actuators are described as follows:

Hydraulic Actuators

Hydraulic Actuators are devices used to get mechanical work by using hydraulic energy of the piston fluid. The mechanical motion may be rotary, linear or oscillatory. Hydraulic actuators produce high precise movements. A hydraulic actuator is generally a linear actuator producing linear displacement. It generally consists of a cylindrical tube, piston and high density oil. By varying the pressure, flow rate of the oil supplied the piston can be displaced by varying amounts. A hydraulic actuator is either operated manually or by means of pumps. It is generally used in cranes and excavators.

Pneumatic Actuators

Pneumatic Actuators utilize the pressure energy of fluid to produce mechanical work, linear or rotary motions. These actuators respond quickly to operations like start or stop. Also they can be used in place of hydraulic actuators which are prone to leaks. But, these types of actuators suffer from disadvantages like: occupy more space, lot of noise is produced and transportation issues.

Electric Actuators

These devices use motors to convert electrical energy to mechanical torque. As oil is not used, electric actuators are cleaner than hydraulic actuators. These devices are mostly used in engines. The servomotor is an example of an electric actuator.

Mechanical Actuators

Mechanical Actuators convert linear motions to rotary motions and vice-versa. Devices such as chain, pulley, rope-drives, gear-shaft, etc. are used for this purpose. As an example, the rotation of chain-sprocket arrangement, in a cycle, causes rotation of rear wheel and hence a translational motion is produced.

3.4.3 POWER CONVERSION UNITS

Power conversion units are composed of devices used for supplying power for functioning of robots. Small mobile machines may need small source of power. Power systems include batteries, flywheel and generators, super-capacitors, thermoelectric generators, flywheel and tethers. The various types of power sources are discussed hereafter.

Rechargeable Batteries

Rechargeable batteries generate electricity through reactions which can be reversed by passing a current in the reverse direction. Ni-Cd batteries are the most common in these types. Owing to their cost they are limited to low-power uses.

Lithium ions (Li+)

Lithium batteries have huge energy density over Ni-Cd. Li+ is a limited source of power and a high current is not allowable. Lithium batteries have to be carefully charged to avoid fire and burning because the heating effects during charging can ignite Li+.

We will use Li+ batteries in our robot. However there are other power sources like fuel cells, super-capacitors, generators, flywheel and tethers, not used in our robot, but can be used in systems as per conditions.

3.4.4. KINEMATICS & DYNAMICS OF ROBOT

3.4.4.1ROBOT KINEMATICS

Robot kinematics is mathematical and geometrical study of motion of a robot arm w.r.t. a fixed reference coordinate system. We do not consider the forces/moments responsible for the motion. The two main problems in robot arm kinematics are namely: forward kinematics and inverse kinematics problem. In forward kinematics we determine the coordinate of end-effector when we are provided the angles and coordinates of links. In inverse kinematics we determine configuration of links provided the configurations of end-effector are provided.

The kinematics problem is to find a transformation matrix co-relating body coordinate frame to a base coordinate frame. The relation between the body reference coordinate to base coordinate of reference is referred to by transformation matrix.

The transformation is represented by the following components:

a. Rotation Matrix
b. Translation vector

c. Perspective transformation

d. Scaling

The various rotation matrices are:

i. R(x, ø)= Rotation about x axis by an angle ø in anti-clockwise sense:

$$R(x, ø)= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos ø & -\sin ø \\ 0 & \sin ø & \cos ø \end{pmatrix} \text{-----------------------(i)}$$

ii. Rotation about y axis by an angle μ in anti-clockwise sense:

$$R(y, \mu)= \begin{pmatrix} \cos \mu & 0 & \sin \mu \\ 0 & 1 & 0 \\ -\sin \mu & 0 & \cos \mu \end{pmatrix} \text{-----------------------(ii)}$$

iii. Rotation about z axis by an angle r in anti-clockwise sense:

$$R(z, r)= \begin{pmatrix} \cos r & -\sin r & 0 \\ \sin r & \cos r & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{-------------------------(iii)}$$

If two co-ordinate frames are orthonormal and given by X and A, R to be the transformation that maps X co-ordinate into A co-ordinate,

Then, $X = RA$ ----------------(iv)

and A maps with X by the relation

$A = R^{-1}X = R^{T}X$ ---------------(v)

A general method for solving the kinematic equations of a robot manipulator with many joints is the use of “homogeneous transformations”. A generalized transformation is now described by a single matrix that combines the effects of translation and rotation.

The entries consist of a 3x3 sub-matrix responsible for the rotation and a 3x1 submatrix responsible for translation. Two other partitions may be identified as perspective transformation and scaling factor.

3.4.4.2 ROBOT DYNAMICS

Robot dynamics deals with the study of the forces/moments responsible for the robot arm motion. Newtonian and Lagrangian mechanics is generally utilized for computing the dynamics motion- dynamics equations for various manipulator joints.

3.4.5 CONTROLLERS

A control system is a representation or organization of components in a way as to regulate command or direct both itself as well as other system.

The elements of a control system are:

1. Controlled Variable –Quantity which controlled directly.
2. Command – The input which can be independently varied.
3. Indirectly controlled variable – Quantity controlled indirectly
4. Reference input – A standard signal used for comparison in the closed-loop system.
5. Actuating Signal –Difference between reference and actual signals.
6. Disturbance – Any signal other than the reference which affects the system performance.
7. System Error – Difference between actual and ideal values.

Some examples of control system applications are steering control of vehicle, speed control system, print wheel control system, etc.

Control systems are classified into the following two basic types:

1. Open-loop control system (unmonitored or non-feedback control systems)
2. Closed-loop control system (monitored or feedback control systems)

A closed-loop control system operating without human operator is called an automatic control system. Some examples are centrifugal watt governor, pressure control system, the lever control system, etc.

Advantages of automatic control system are:

1. Increased output
2. Improvement in the quality of products
3. Reduced effect of non-linearity and distortion.

4. Response is satisfactory over a wide range of input frequencies.

However, automatic control system has a tendency to overcorrect errors which may result in oscillations of constant or changing amplitude.

3.4.6 PROGRAMMING LANGUAGES

A robot program is used to define manipulator's path in its workspace. It combines this feature with peripheral actions to reinforce the work cycle.

Peripheral actions comprise:

1. Opening and closing of grippers.
2. Performing logical decision making.
3. Communicating to various robot parts.

Also, robot languages have been developed for the ease of control of motions of robots having different structures and geometric capabilities.

The various requirements of a robot programming language are:

1. Defining expected movement of manipulator.
2. Sensory control.
3. Robot-user interface.
4. The robot can be modeled by defining named variables.

CHAPTER-4

ARTIFICIAL NEURAL NETWORKS

4.1 INTRODUCTION

A neural network consists of a very large number of parallel-distributed processors that has ability to store experimental information and furnishing it whenever required to be applied. It similar to the brain in 2 ways:

1. The network learns through a learning process, and,
2. The information is stored by means of neuron-neuron connections called as synaptic weights.

Neural networks can also be defined as non-linear algorithms for image or signal or data processing.

ANN is an information-processing method. The information is processed by means of elements called as neurons. The outputs are passed through link connections. The output signal is realized by using activation functions on the net input.

An artificial neuron is characterized by:

1. Activation functions
2. Training and learning
3. Architecture

ANN models try to use some characteristic principles behind the working of the human brain.

Neural networks help in deriving conclusions from even imprecise data, complicated information, ascertain trends, and discover patterns for both computers as well as humans.

A neural network may become proficient in analyzing data if it is given training. It can predict solutions to that information it has not been trained into.

Other advantages include:

1. Self-Organization: Develops a structure of the data received while being taught.
2. Some network faculties may be kept when some significant network damage occurs.
3. Real-Time operation.

4. Adaptive learning: Ability to gain knowledge on performing tasks based on data provided while being trained.

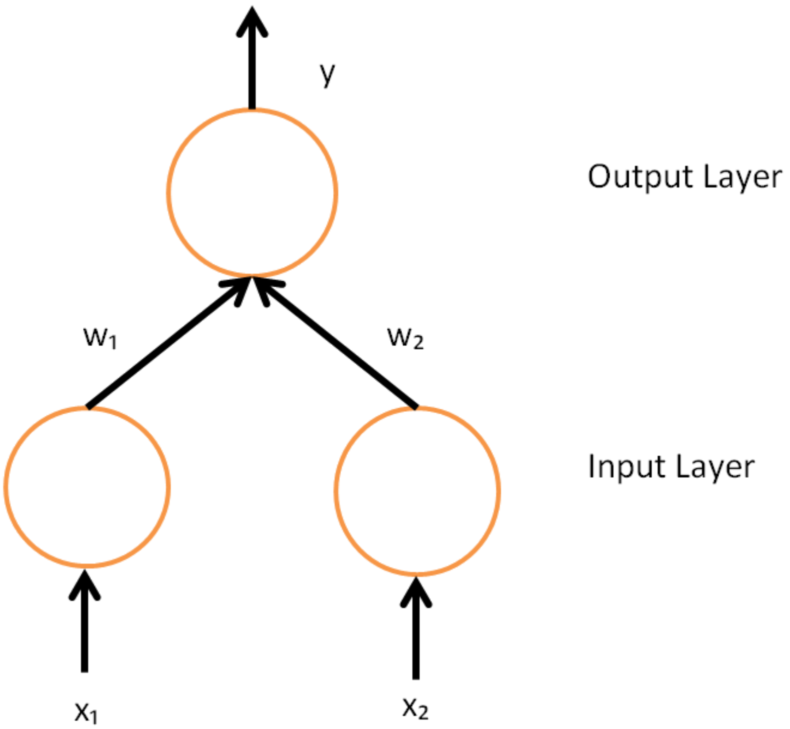


Figure5. A simple ANN

4.2 BIOLOGICAL NEURAL NETWORKS

A biological neural network or a nerve cell comprises the cell body, dendrites, the axon, and the synaptic weights. The building blocks are discussed as follows:

Dendrite –Acquiring information from other neurons.

Soma – Adds all the incoming information.

Axon –When input reaches a pre-determined threshold value, the information is passed onto subsequent neurons.

A neuron is the basic element of a neural network. A biological neuron acquires inputs from previous neurons, integrates them as per a rule, performs non-linear manipulation on the result, and then provides the outputs.

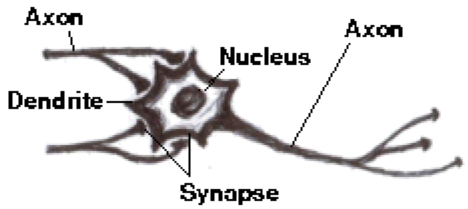


Figure6. A biological neuron

The properties of the biological neuron pose some features on the artificial neurons. They are:

1. Signals are received by the processing elements.
2. The weights at the receiving end have the capability to modify the incoming signal.
3. The neuron fires (transmits output), when sufficient input is obtained.
4. The output produced from one neuron may be transmitted to other neurons.
5. The processing of information is found to be local.
6. The weights can be modified by experience.
7. Neurotransmitters for the synapse may be excitatory or inhibitory.
8. Both artificial and biological neurons have inbuilt fault tolerance.

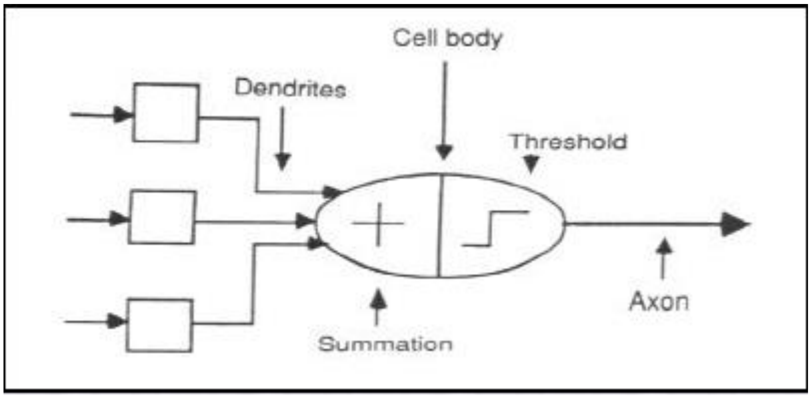


Figure7. A neuron model

Table1. Associated terminologies of Biological neural networks and Artificial neural networks

Biological Neural Network	Artificial Neural Network
Cell Body	Neurons
Dendrite	Weights
Soma	Net input
Axon	Output

4.3 COMPARISON BETWEEN ARTIFICIAL NEURAL NETWORK & BIOLOGICAL NEURAL NETWORK

The major differences between artificial and biological neural network is shown as below:

Table2. Comparison between artificial neural network and biological neural network

Characteristics	Artificial Neural Network	Biological Neural Network
Speed	Faster in processing data. One cycle is operated in about nanoseconds.	Slower in processing information. One cycle is operated in milliseconds.
Processing	Instructions are operated sequentially, one after another.	Parallel processing of information takes place.
Size and complexity	Difficult to perform complex pattern recognition.	Due to greater number of processing elements, more complex pattern recognition tasks can be accomplished.
Storage	New information in a memory location replaces the old information. It is strictly replaceable.	Old information and new information co-exist.
Control Mechanism	A control unit monitors all the computing activities.	No specific control mechanism.

4.4 BASIC BUILDING BLOCKS OF ARTIFICIAL NEURAL NETWORKS

The basic building blocks of the artificial neural network are:

1. Network architecture
2. Setting the weights
3. Activation function

4.4.1 NETWORK ARCHITECTURE

The structuring of neurons into various layers and the trend of relationships intra-layer and inter-layer is referred to as the network architecture. The neurons contained in a layer are either not interconnected or fully interconnected. There are various types of network architectures: Radial Basis, Cascaded, Back propagation etc.

ANNs can be of varying size and shape. The network behaves as a vector-valued function which takes in an input vector and returns another on the output. Various types of network architectures have been discussed later.

4.4.2 SETTING THE WEIGHTS

Training process is basically determining the values of the weights. Learning is the process of internally training a network. Generally the 3 types of learning methods are as follows:

1. **SUPERVISED TRAINING**

   In this process the network is provided with sample inputs and estimated outputs. The training sets values to weights and computes outputs which are then compared with the expected results. The learning runs till the output is equal to the expected results. These weighs are determined as per the learning algorithm. This method is used in pattern recognition. Some examples of supervised training are Back propagation network, Hebb network.

2. **UNSUPERVISED TRAINING**

   In unsupervised training the output vectors are not known for the respective input vectors. Unsupervised networks are found to be more complex and difficult to implement. It involves looping connections back into feedback layers and iterating through the process until some sort of stable recall can be achieved. Unsupervised networks are also called selflearning or self-organizing networks. Unsupervised training is used in the areas of adaptive resonance theory, self-organizing feature maps, etc.

3. **REINFORCEMENT TRAINING**

   In this method, the expected answer is not provided to the network. Instead, it is given an indication if the output wrong or correct. The network utilizes this data to refine its interpretation. If sufficient information is available, the reinforcement learning can readily handle a specific problem. In this method a trial and error method is used to map inputs with outputs. The network does not know the correct output even if it knows output to be right or wrong.

4.4.3 ACTIVATION FUNCIONS

The output to a neuron is obtained by using activation. For neurons in the same layer, same activation functions are used. Activation function may be linear or non-linear. Ex of a linear activation function f(x) = x; sigmoidal function is a non-linear function.

4.5 VARIOUS NEURAL NETWORKS

The major types of artificial neural networks are:

1. Cascaded Neural Network
2. Elman Neural Network
3. Feed-forward Back-propagation Neural Network
4. Generalized Regression Neural Network
5. Radial Basis Neural Network.

4.5.1 CASCADED NEURAL NETWORK

A single neuron cascaded (SNC) architecture provides compact and simpler models with less time consumed for performing tasks. Its unique network structure allows automation in design. This network combines the strengths of multi-layer and single-layer neural networks.

The Cascaded architecture comprises an input layer, hidden layers and an output layer. The input layer receives external information as input. Subsequent layers get external information as well as outputs of all preceding layers. As the input to a neuron comprises external inputs and outputs of all previous layers it is called as Cascaded. A cascaded neural network can have any number of neurons in each layer. The cascading of inputs improves its mapping capability.

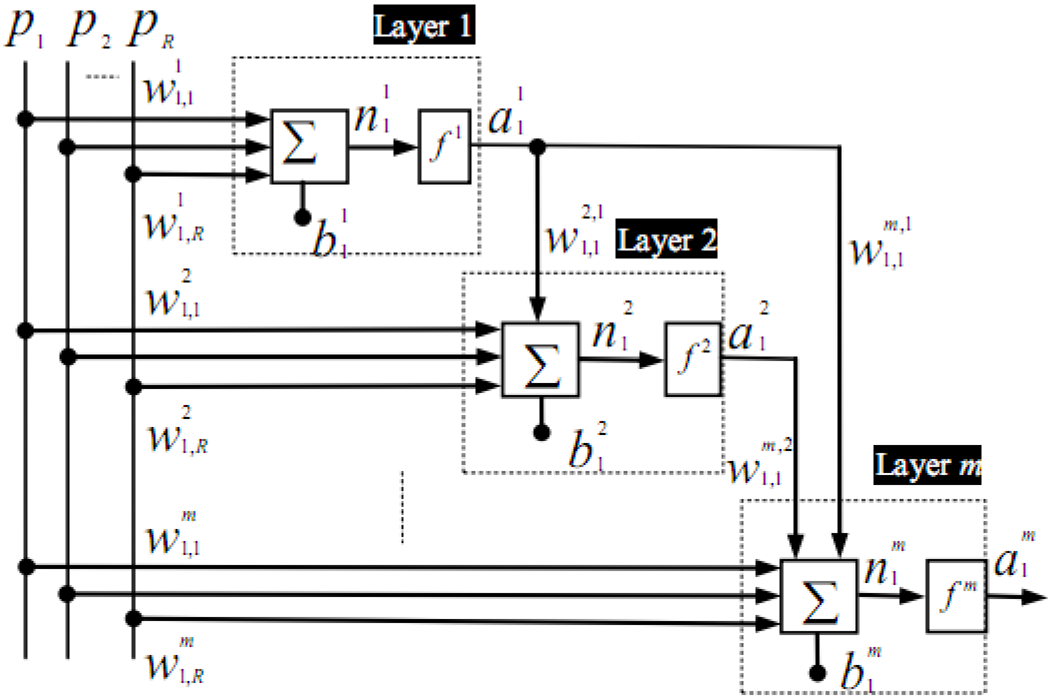


Figure8. A cascaded network

At the beginning we train only one hidden layer having only one neuron between input and output. To develop a multi-layer network, hidden layers are added sequentially.

The cascaded network possesses the advantages of multi-layer and single-layer networks. The result is a much simpler architecture that can undergo training and growth, of its own, to get the expected response. It is also executes faster.

4.5.2 ELMAN NEURAL NETWORK

The ENN is a simple recurrent neural network. It was created in 1990 by Jeffrey L. Elman. ENN comprises an input layer, a hidden layer and an output layer. Hence it is akin to a 3-layer feed-forward neural network. Additionally, it also consist a context layer. This context layer has its input as the output of the hidden layer. This output is then fed to the hidden layer in subsequent iterations. This allows for the formation of a short-term memory. ENNs are very helpful in forecasting time variation patterns. The image shows an ENN.

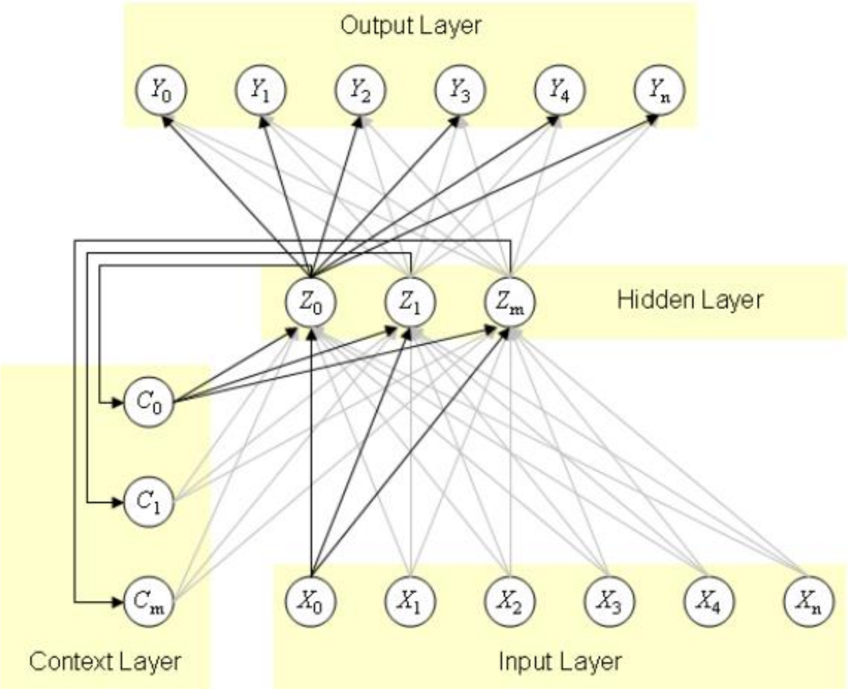


Figure9. An Elman structure

4.6.3 GENERAL REGRESSION NEURAL NETWORK

It was proposed by D.F. Specht. It is a type of probabilistic neural network. GRNN requires simply a portion of the training inputs that a feed-forward algorithm requires. This characterizes GRNN as a useful tool to predict as well as compare performance of system.

The probability distribution function used in Generalized Regression Neural Network is the Normal Distribution. GRNN can solve any function approximation. Unlike back-propagation, it does not need an iterative algorithm. A Generalized Regression Neural Network comprises four layers namely input, pattern-summation, summation layer and output.

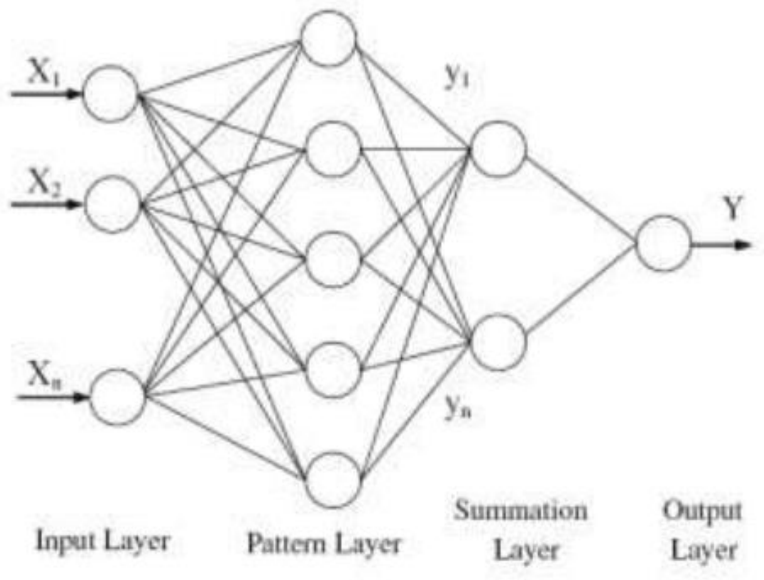


Figure10. A Generalized Regression Neural Network

4.5.4 RADIAL BASIS NEURAL NETWORK

This neural network is an ANN that uses radial basis functions as activation functions. The output of the network is a linear combination of these functions (functions dependent on the norm w.r.t. a center vector and are radially symmetric about the vector) of the inputs and the neuron parameters. RBFNs have applications in function approximation, forecasting time series. It was proposed by Lowe and Broomhead in 1998.

RBFs have three layers- input, hidden alongwith a non-linear RBF activation function and a linear output layer.

Euclidean distance and Gaussian function are taken to be the norm and activation function respectively.

The various features of RBF are:

1. They are 2-layer feed-forward networks- In the 1st layer weights from input to hidden layer is computed followed by the weights of hidden to output layer.
2. Hidden nodes execute a set of RBFs.

3. The output nodes execute linear simulation functions as in a multi-layer perceptron layer (MLP) network.
4. The training is very fast.
5. The network is adept at interpolations.

The properties of RBFs are attracting a great deal of interest due to their rapid training, generality and simplicity.

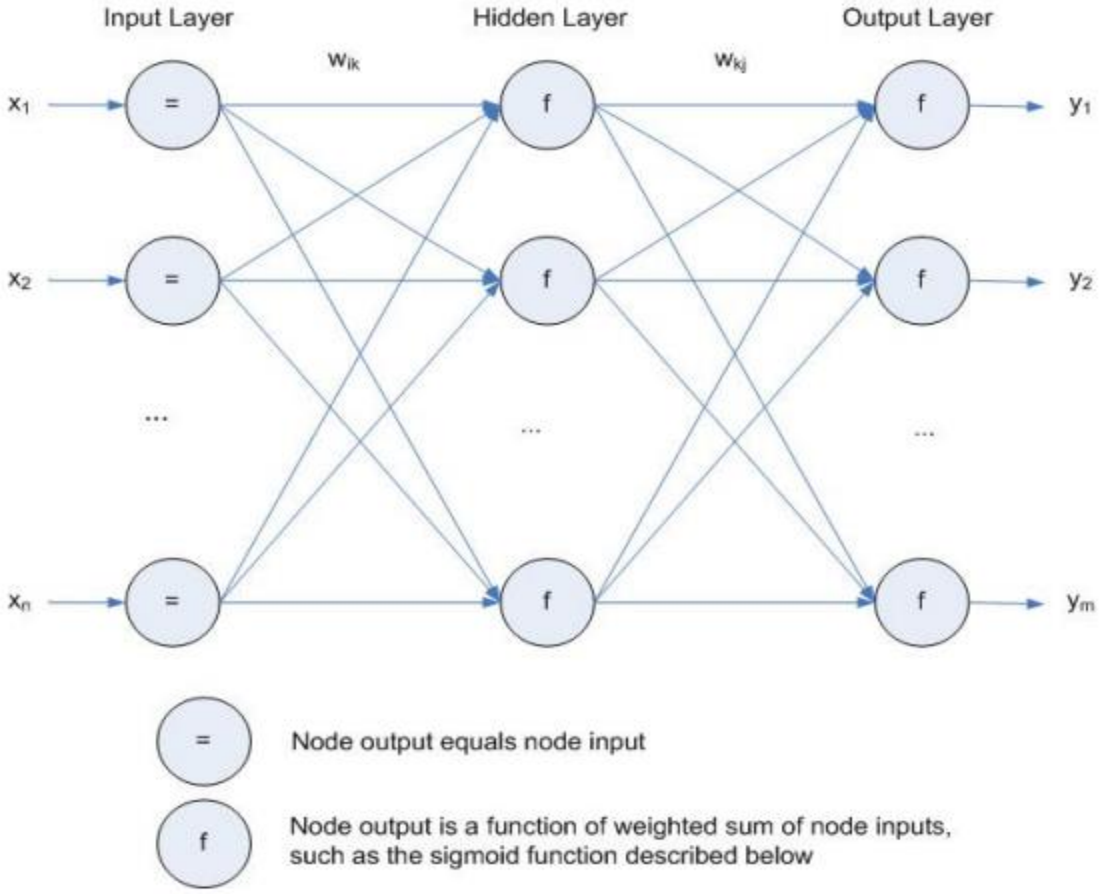

Input Layer
Hidden Layer
Output Layer
wik
wkj
x1
x2
...
xn
=
f
y1
y2
ym
Node output equals node input
Node output is a function of weighted sum of node inputs, such as the sigmoid function described below

Figure11. A RBF Network Structure

4.6.5 FEED-FORWARD BACK PROPAGATION NEURAL NETWORK

A FFNN has neurons only connected forward, but there are no backward connections. The term "feed forward" refers to the way network processes and recalls functions. The term "back propagation" refers to the way network is trained. It is an example of supervised learning. By utilizing the expected responses, the back propagation training algorithm makes calculations of errors and changes the weights of different layers traversing backwards-output layer to input layer.

The signals moves forward in one direction-input nodes_>hidden nodes_> output nodes, only. Cycles and loops are absent. Each node in a layer has directed relationships to the nodes of next layer. Every node in a layer is related to all the units in the preceding layer. These connections may be of varying weights.

The input layer is numbered as a layer with number 0. Any other layer that is not an output layer is a hidden layer. These have no connection to the outside world. In back propagation learning, every time an input, the output (o) is compared to the expected result (e). Then the squared error is calculated:

Error = $(d\text{-}o)^2$

The error indicates the deviation between the expected response for a particular input. In order to get the desirable outputs it is required to minimize the total Error.

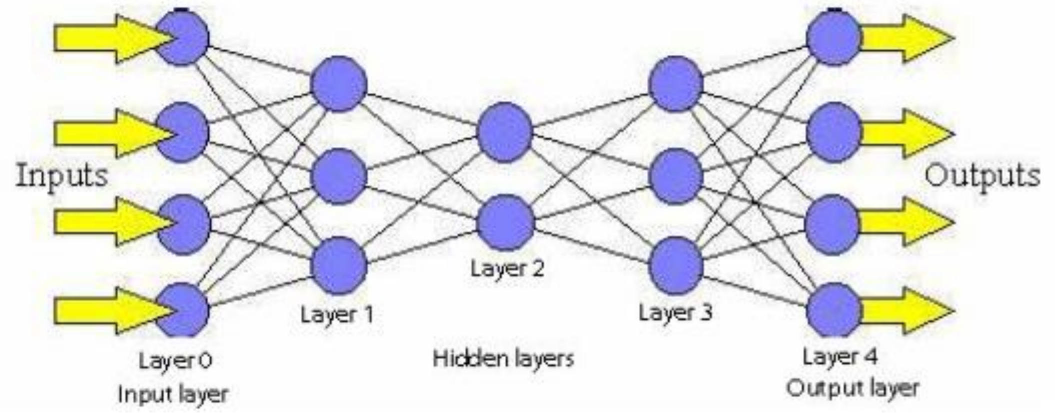


Figure12. A feed-forward back propagation neural network

An outline of this supervised learning algorithm:

1. Initially, set all the weights to some random values.
2. Repeat
    a) Feed the network with an input ‘x’ from the training state.
    b) Compute the network’s output.

    c) Change the weights of the nodes.

3. Stop when Error becomes small.

Complexity depends on the extent of the network, the quantity of patterns to be trained to, the number of epochs, the tolerance of the minimizer and the speed of the computer. Back propagation ANNs are extensively used in diverse applications, such as pattern recognition, location selection and performance evaluations.

Note: We shall be using Feed Forward Back Propagation Neural Network in our project.

CHAPTER-5

ROBOT CONSTRUCTION

As we have discussed in Chapter-3, the various elements of robot architecture are:

1. Sensors
2. Actuators
3. Mechanical Structure- Kinematics and dynamics
4. Controllers
5. Power Conversion Units
6. Programming Languages

A detailed description of each individual component used in the construction of robot has been given herewith.

5.1 SENSORS

INFRARED SENSORS

Name: SHARP GP2Y0A02YK0F 20cm to 150cm IR Range Sensor

Quantity – 1 piece

Table3. IR Sensor Characteristics

PARAMETER	Specifications
Distance measuring range	20 cm – 150 cm
Supply voltage	4.5 V – 5.5 V
Output terminal Voltage	0.25 V – 0.55 V
Average Current dissipation	Max. 50 mA
Difference of output voltage	1.8 V – 2.30 V

Figure13. IR sensor

ULTRASONIC SENSOR

Name: Ultrasonic Range Finder HC SR-04

Quantity – 2 pieces

Table4. Ultrasonic Sensor characteristics

PARAMETER	SPECIFICATIONS
Distance Measuring Range	2 cm – 400 cm
Working Voltage	5V DC
Measuring angle	$< 15^0$
Working frequency	40 Hz
Dimensions	45*20*15 mm
Working Current	15mA
Easy pin configuration	VCC, Trig, Echo, GND
Trigger Input Signal	10uS TTL pulse

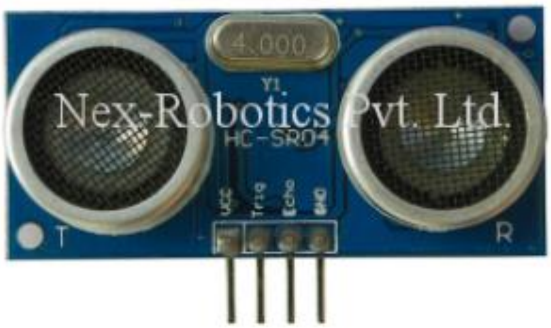


Figure14. Ultrasonic Range Finder HC SR-04

5.2 ACTUATORS

DC MOTOR

Name: 30 RPM Centre Shaft Economy Series DC Motor

Quantity – 4 pieces (1 piece per wheel)

Table5. DC Motor characteristics

PARAMETER	SPECIFICATIONS
RPM	30 at 12 V
Brush Type	Precious Metal
DC supply	4 V – 12V
Gear head diameter	37 mm
Shaft diameter	6 mm
Motor diameter	36 mm
Gear head length	21 mm
Shaft length	22 mm
Motor length	25 mm
Gear assembly	Spur
Total length	46 mm
Motor weight	100 gms

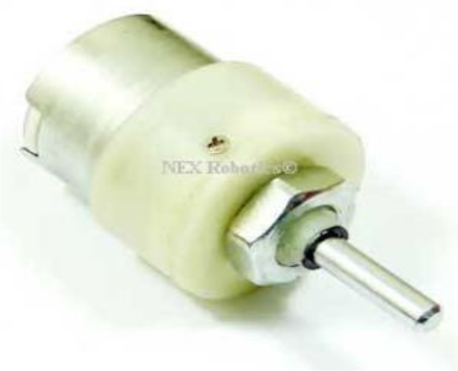


Figure15. 30 RPM Center Economy Series DC motor

5.3 CONTROLLERS

MICROCONTROLLER

Name: Arduino Mega 2560 R3

Quantity – 1 piece

Table6. Arduino Mega 2560 R3 Characteristics

PARAMETERS	SPECIFICATIONS
Microcontroller	ATmega2560
Input Voltage(recommended)	7 V-12V
DC Current per I/O Pin	40 mA
Input Voltage(limits)	6V-20V
DC Current for 3.3V Pin	50 mA
EEPROM	4KB
Operating Voltage	5V
SRAM	8KB
Flash Memory	256 KB of which 8KB used by bootloader
Digital I/O Pins	54 (15 provide PWM output)
Clock Speed	16 MHz
Analog Input pins	16

Figure16. Arduino Mega 2560 R3

MOTOR DRIVER

Name: L298 Motor Driver Kit

Quantity: 2 pieces (1 piece per 2 motors)

Table7. Motor driver Characteristics

PARAMETER	SPECIFICATIONS
Operating PWM frequency	Max. 20KHz
Bidirectional Control of 2 DC motors	Max. 2 A each
Operating Supply Voltage	Max. 46V

- Multi watt 15 package
- Current Sensing output
- Over temperature protection.

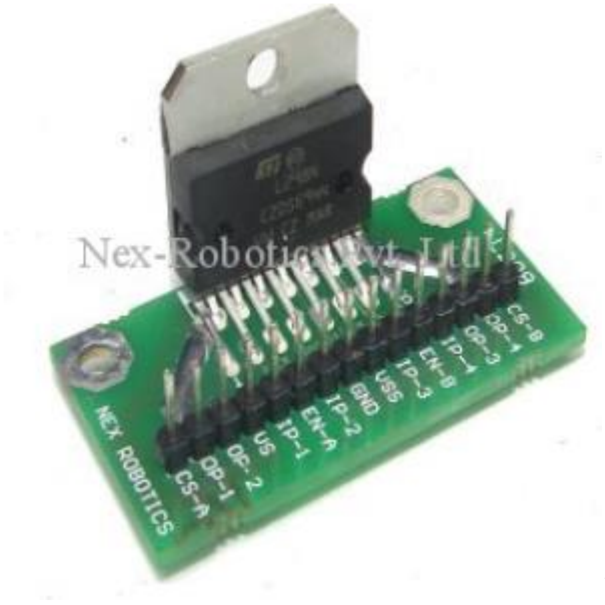


Figure17. L298 motor driver

5.4 POWER CONVERSION UNITS

BATTERY

Name: Lithium Polymer 3 Cell, 11.1V, 2000mAh, 20C discharge Battery

Quantity – 1 piece

Figure18. Li-Polymer battery

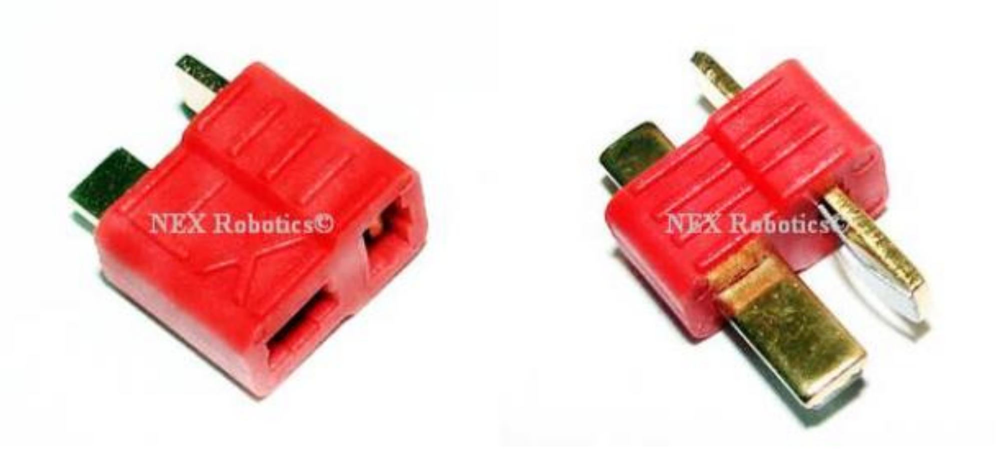


Figure19. ‘T’ type female and male power connectors

Table8. Battery characteristics

PARAMETER	SPECIFICATION
Battery Voltage	11.1V
Number of cells	3
Battery weight	195 gms
Battery capacity	2000 mAh
Dimensions	10.5 * 3.5*3 cm
Maximum constant discharge current	40 A
Recommended battery cut off voltage	9.9V
Maximum charging current	4.0 A

Note: All these components have been bought from Nex-Robotics, Navi Mumbai, Maharastra, India.

5.5 PROGRAMMING LANGUAGES

We have used MATLAB for the purpose of Simulation. C- Language has been used to run the robot.

5.6 MISCELLANEOUS PARTS

BREAD BOARD and JUMP WIRES.

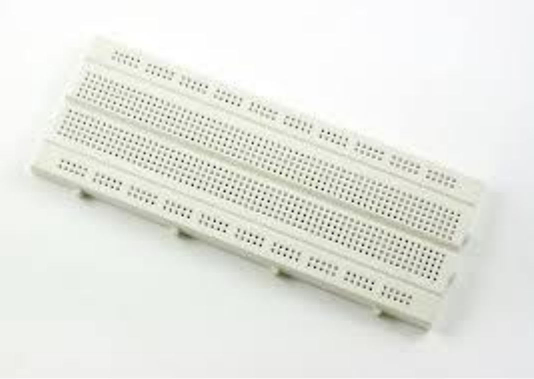


Figure20. A bread-board

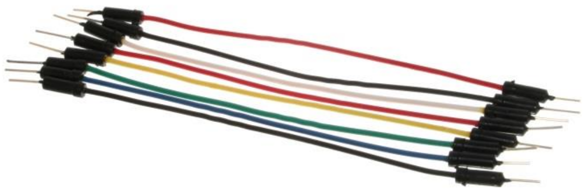


Figure21. Jump Wires

5.7 FINAL ROBOT ASSEMBLY

The pictures shown below are the assembled pictures of in-built mobile robot.

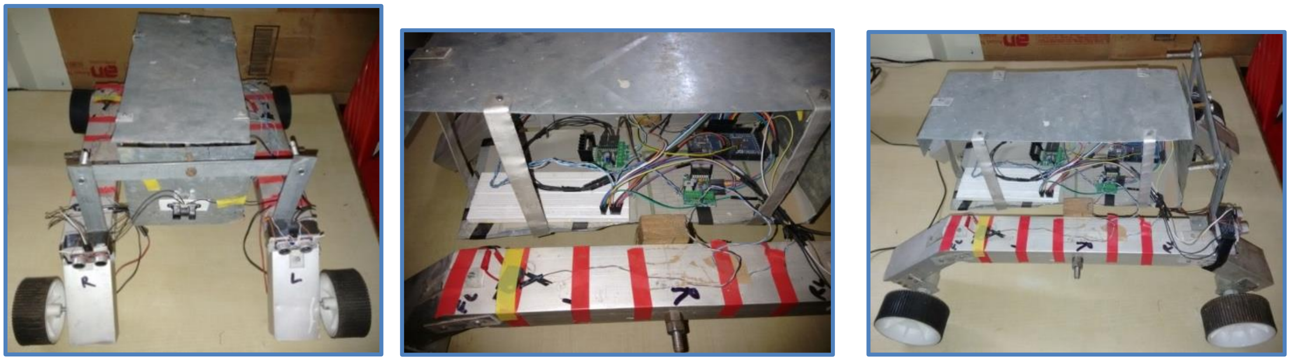


Figure22. Photos of in-built robot assembly

CHAPTER-6

SIMULATION & TRAINING

6.1 ONLINE TRAINING

As a robot moves in its environment towards the goal it is likely to encounter obstacles. The robot task in such a condition is to safely reach its goal without any collisions with the additional requirement of choosing the most optimum. A threshold distance (safe distance) needs to be maintained as it moves towards the target. The robot is being continuously fed with inputs-distance from obstacles, by means of sensors. By the available data it has to undergo either of the following operations on encountering an obstacle:

- If forward obstacle distance < threshold distance (safe distance) then look for other direction, left or right.
- If left obstacle distance< threshold distance turn right or move ahead

Table9. Input distances and output angle rotation

Sl.No.	f.o.d.(cm)	l.o.d.(cm)	r.o.d.(cm)	T.A.(degrees)	TURNING
1	16	80	16	90	left
2	16	16	80	-90	right
3	80	8	80	-30	right
4	20	60	40	55	left
5	30	80	50	55	left
6	12	80	80	90	left
7	20	40	80	-55	right
8	12	12	12	-85	right
9	40	12	12	0	straight
10	20	8	8	10	small left
11	80	12	12	0	straight
12	20	8	8	-10	small right
13	80	12	80	0	straight
14	80	12	12	0	straight
15	80	80	80	-85	right

f.o.d.: distance from front obstacle (cm) l.o.d.: distance from left obstacle (cm)

r.o.d.: distance from right obstacle (cm)

T.A.: the angle by which the robot turns on detecting an obstacle.

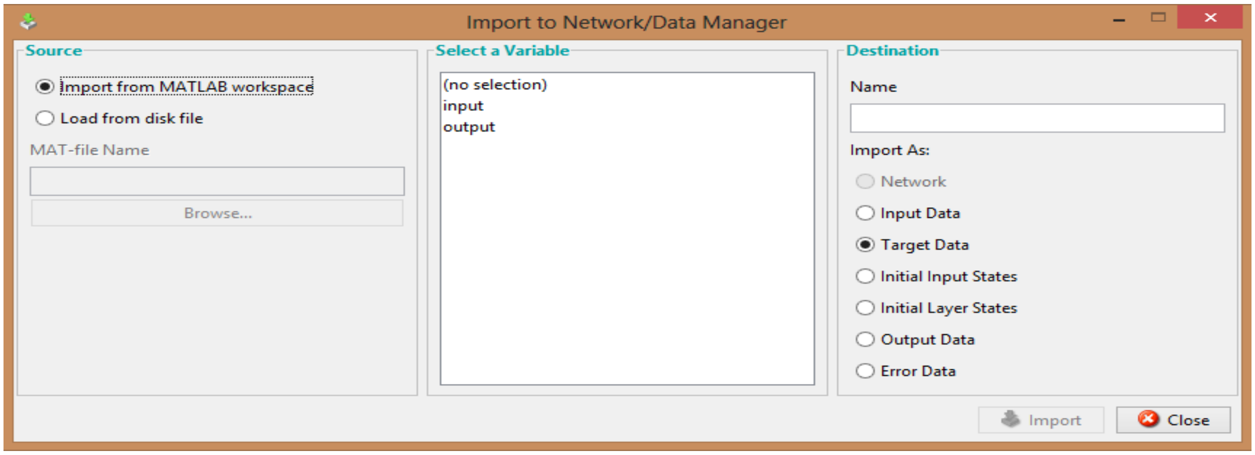
We will explain by means of an example. Let us take sl. no. 7 as an example. If f.o.d., l.o.d. and r.o.d. are 20 cms, 40 cms, 80 cms respectively then the robot will move in straight line towards the goal. If any of the distances is less than the safe distance then the robot will turn clockwise will by 55^0. (We have used the notation clockwise(-) for right turn and anti-clockwise(+) for left turn.) After turning through the robot continues to move towards the target unless it encounters another obstacle where the above procedure is again followed.

In the real-world the above conditions, as mentioned in table, rarely occur. In such a case the robot has to take decisions on its own based on the given data. For this training purpose will use the MATLAB neural network toolbox.

STEPS for TRAINING

1. We will first give the input values by creating a 15x3 matrix in the workspace named ‘input’ and then transpose it.
2. We will now provide the output values by creating a 15x1 matrix in the output named ‘output’ and then transpose it.
3. Open Command Window and then the toolbox by using ‘nntool’.

We will ‘Import’ inputs to this network. The ‘input’ will appear in Input data box. The ‘output’ is also fed by ‘Import’ and it will appear in Target data box.


Import to Network/Data Manager

Source

● Import from MATLAB workspace

○ Load from disk file

MAT-file Name

Select a Variable

(no selection)
input
output

Destination

Name

Import As:

○ Network

○ Input Data

● Target Data

○ Initial Input States

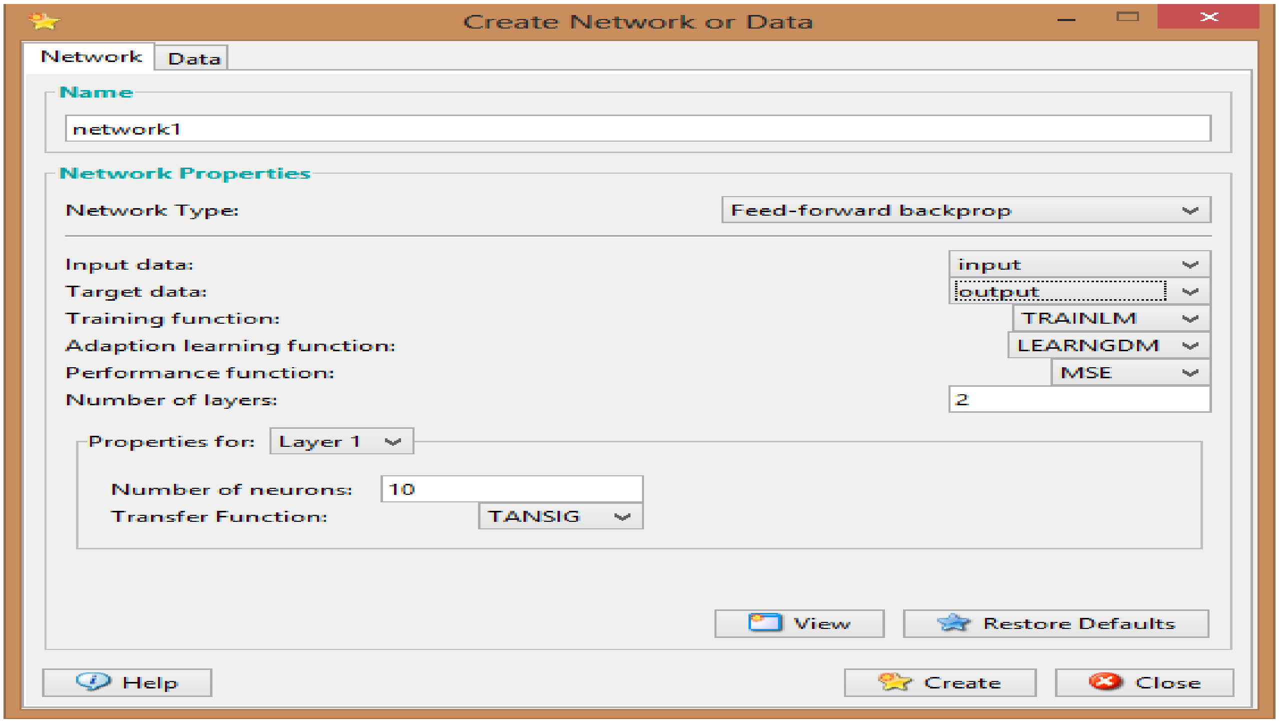
○ Initial Layer States

○ Output Data

○ Error Data

Figure23. ‘Import’ window.

4. A new network is created named ‘network1’ by clicking on ‘New’ button. We then give the Input data and Target data. We have set the number of hidden neurons to 10. Then click on ‘Create’. The ‘network1’ will appear in ‘Networks’ box.


Create Network or Data

Network | Data

Name

network1

Network Properties

Network Type: Feed-forward backprop

Input data: input

Target data: output

Training function: TRAINLM

Adaption learning function: LEARNGDM

Performance function: MSE

Number of layers: 2

Properties for: Layer 1

Number of neurons: 10

Transfer Function: TANSIG

Figure24. Create Network window

5. Double-click on ‘network1’ to generate the network. The output, evaluated by the network, is stored network1_outputs (appear in Output data box) and the error is stored in network1_errors (appear in Error data box).The figure below shows the structure of the network.

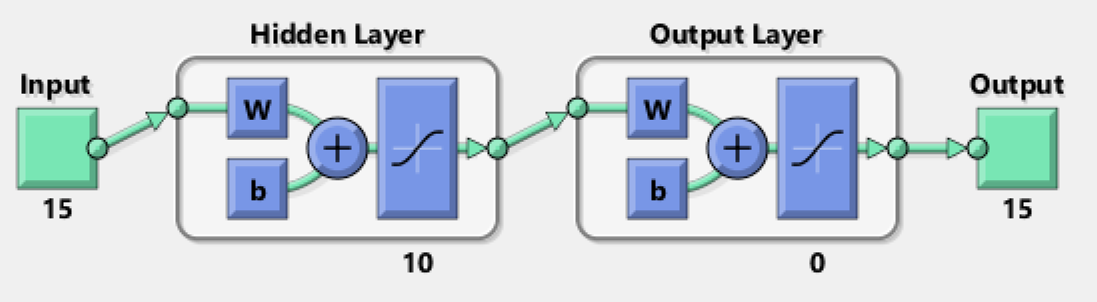
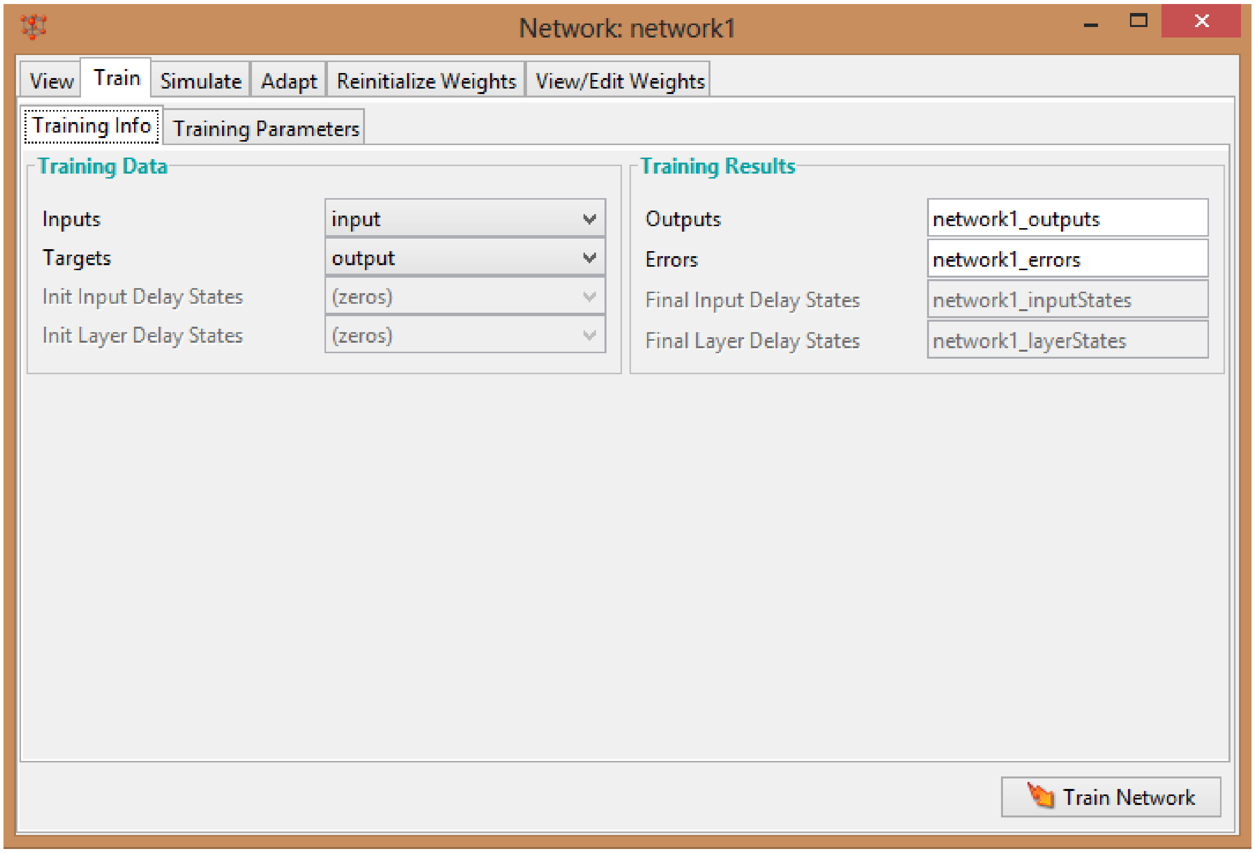


Figure25. Neural Network Structure.

6. Now the network is trained by clicking on ‘Train Network’



Training Data

Inputs	input
Targets	output
Init Input Delay States	(zeros)
Init Layer Delay States	(zeros)

Training Results

Outputs	network1_outputs
Errors	network1_errors
Final Input Delay States	network1_inputStates
Final Layer Delay States	network1_layerStates

Figure26. Training the network.

The training activity has to be performed repeatedly for more than 80 times to get lesser values of mean square error, regression=1, and desired output = actual output. The figure below shows the results of 1 training run.

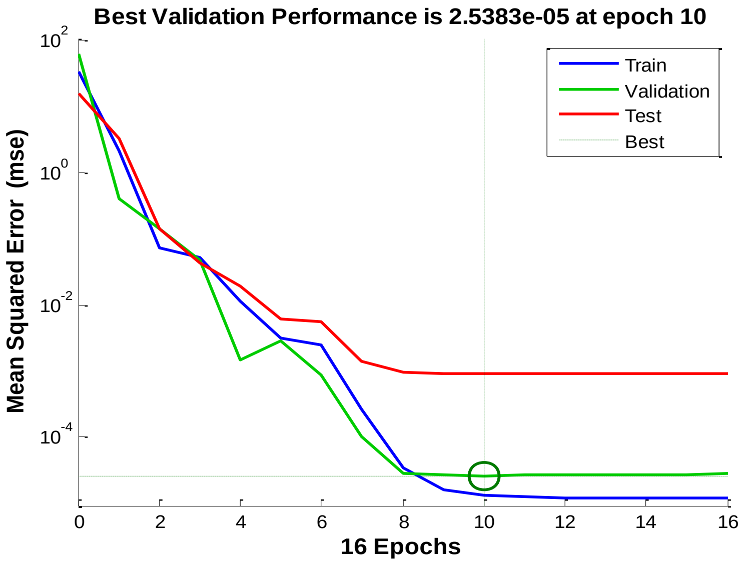


Figure27. Performance curve

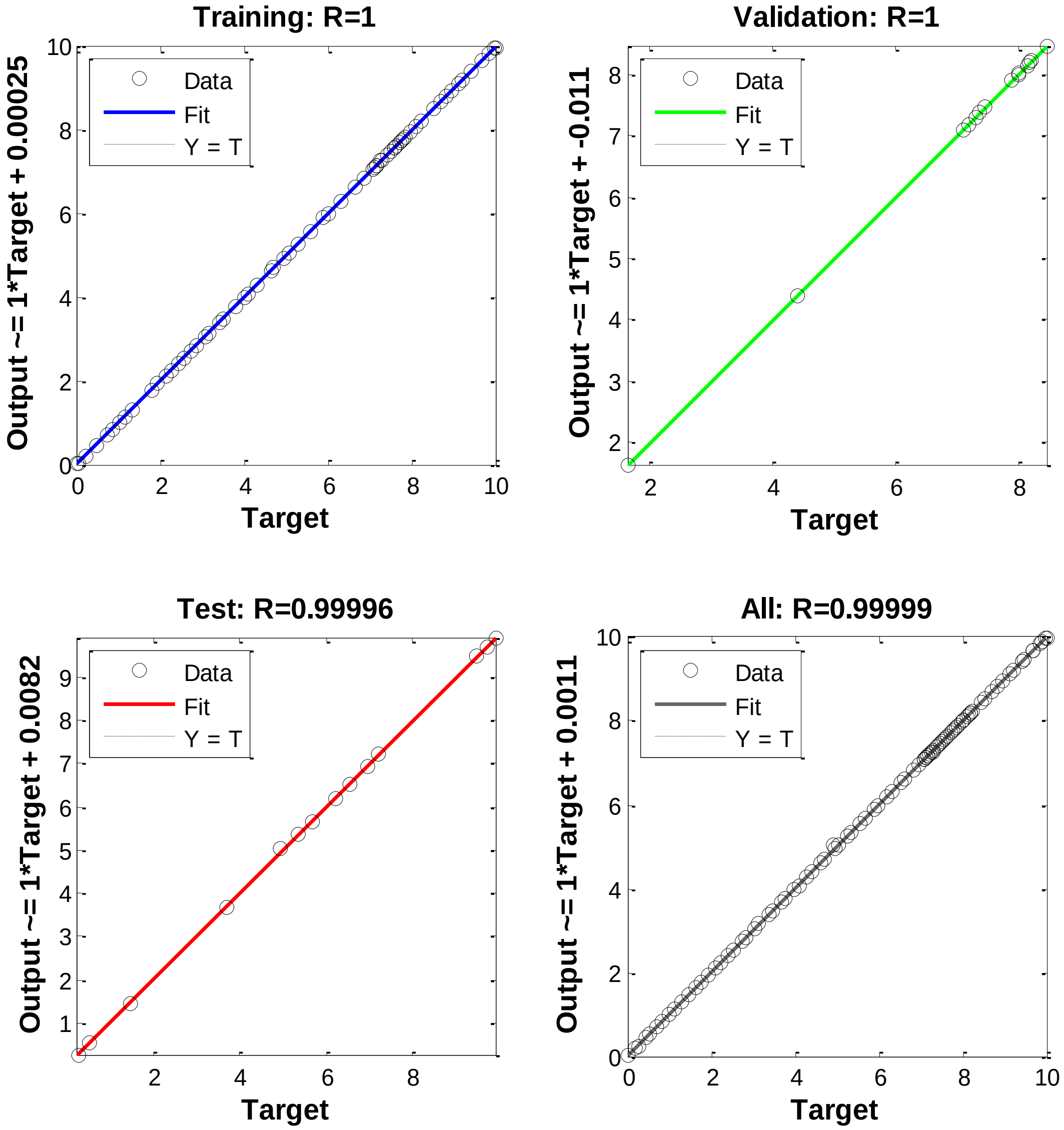


Figure28. Regression curves for Training, Validation and Testing sets

7. Save the network.

After the training has occurred now the network is ready to be used for simulation.

We now use the network in a program where the robot moves towards its goal in an environment containing obstacles. The algorithm is as follows:

1. Input the start and target positions.
2. Input the position of obstacles.
3. Determine the angle in which robot should move.
4. Load the network
5. Run an iteration:
    - Moving the robot towards the goal. Determine the speed of movement.
    - Calculate distance of obstacles from robot.
    - Give safe distance conditions.
    - The robot-obstacle distance is fed to the neural network and the network computes the angle to be taken by the robot.
    - As the robot reaches the target it stops.

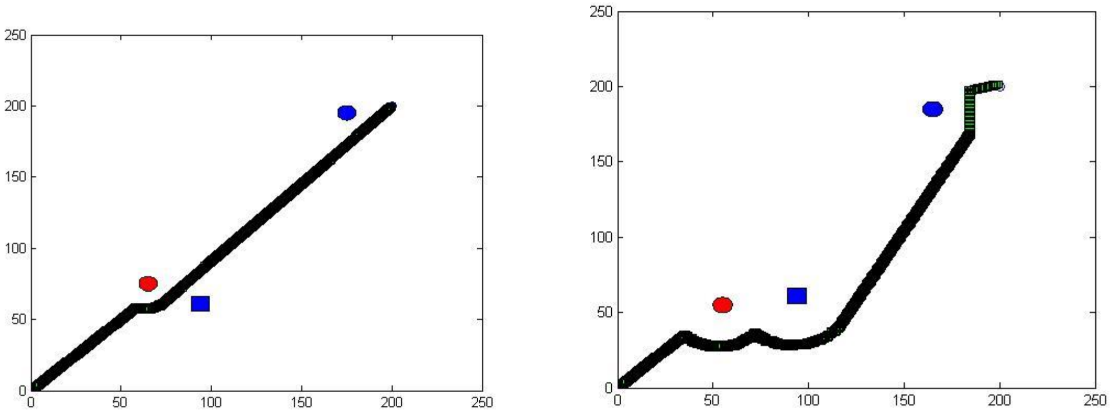


Figure29. Movement of robot in obstacle ridden environment

The following algorithm shows the movement of robot in moving obstacles scenario

1. Give the initial and target coordinates.
2. Give the initial positions of the obstacles.
3. Load the network.
4. Run an iteration
    - Moving the robot towards the goal. Determine the speed of movement.

- Moving the obstacles and set their velocities.
- Calculate distance of obstacles from robot.
- Give safe distance conditions.
- The robot-obstacle distance is fed to the neural network and the network computes the angle to be taken by the robot.
- As the robot reaches the target it stops.

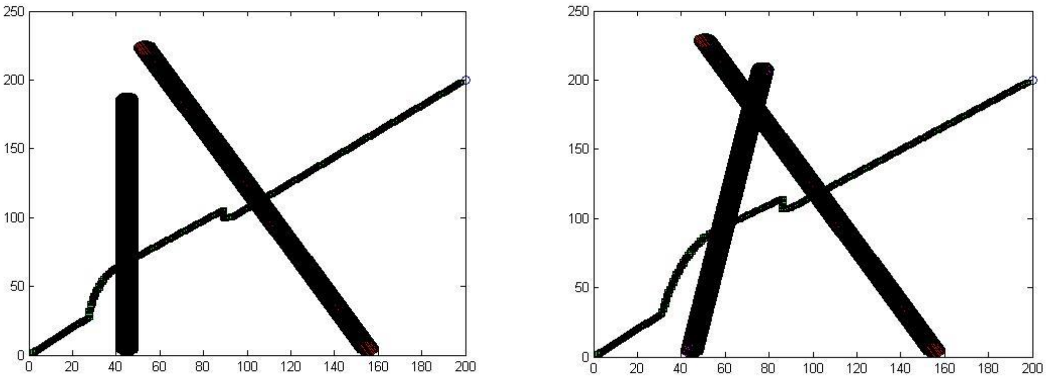


Figure30. Movement of robot in presence of moving obstacles.

The following algorithm shows the movement of robot in multiple obstacles scenario:

1. Give the initial and target coordinates.
2. Give the initial positions of the obstacles.
3. Load the network.
4. Create an array- obdist
5. Run an iteration
    - Calculate the robot-obstacle distance and store in obdist
    - Sort the array obdist in ascending order and store in another array ‘p’. The first 3 values of p give the location of nearest obstacles.
    - Input these obstacles to the network and the network computes the angle to be taken by the robot.
    - As the robot reaches the target it stops.

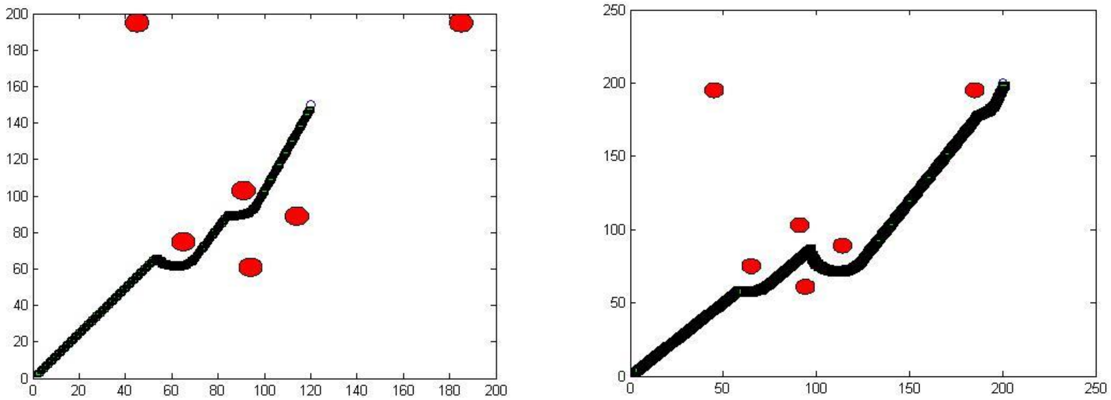


Figure31. Movement of robot in multi-obstacle scenario.

6.2 OFFLINE TRAINING

In online training we were giving Euclidean distances as input to the neural network. In practical situation the sensors detect the obstacles and also the distance between robot & obstacles.

In this case, we need to write a C program in Arduino software. The objective is to turn the robot. The program controls the dc motors by controlling the amount of signal. Depending on the strength of signal, the motor driver controls the speed of the 2 dc motors attached to it. By changing the speeds of different signals we are controlling the steering action of the robot.

Building an Arduino on a Breadboard

Overview

This tutorial shows you how to build an Arduino compatible breadboard with an Atmel Atmega8/168/328 AVR microcontroller and FTDI FT232 breakout board from SparkFun. You could also use the Arduino USB Mini.

Originally created David A. Mellis
Updated from the ITP version by Carlyn Maw
Updated October 23, 2008 by Rory Nugent

Figure32. Arduino workspace

In the following figure, regular time periods are marked by green lines. This period is the inverse of Pulse Width Modulation frequency.For example, each green line at 2 milliseconds implies a pulse width frequency of 500 Hz. A call to analogWrite() is on a scale of 0 – 255. If the call is analogWrite(63) the duty cycle is 25%(on for one-fourth period), and if analogWrite(191) duty cycle is 75%(on for three-fourth of the time) for example. A case of 100% duty cycle means dc motor is operating at full speed while the dc motor is running at half the maximum speed implies 50% duty cycle.

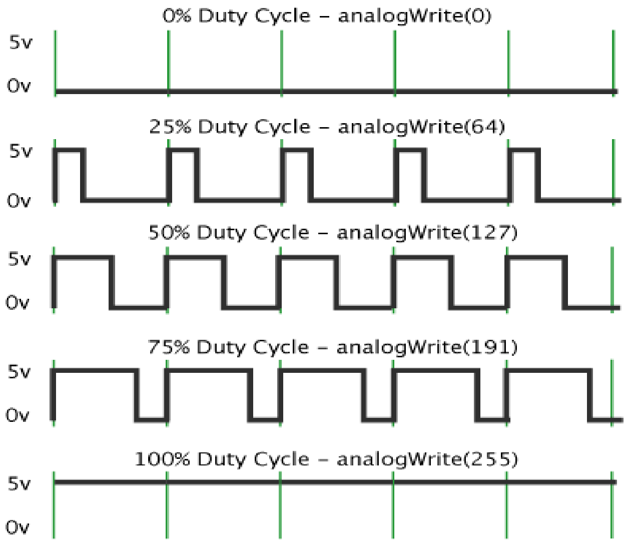

0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v

Figure33. Pulse-width modulation

CHAPTER-7

RESULTS

The various graphics below show the movement of robot towards its goal avoiding the obstacles:

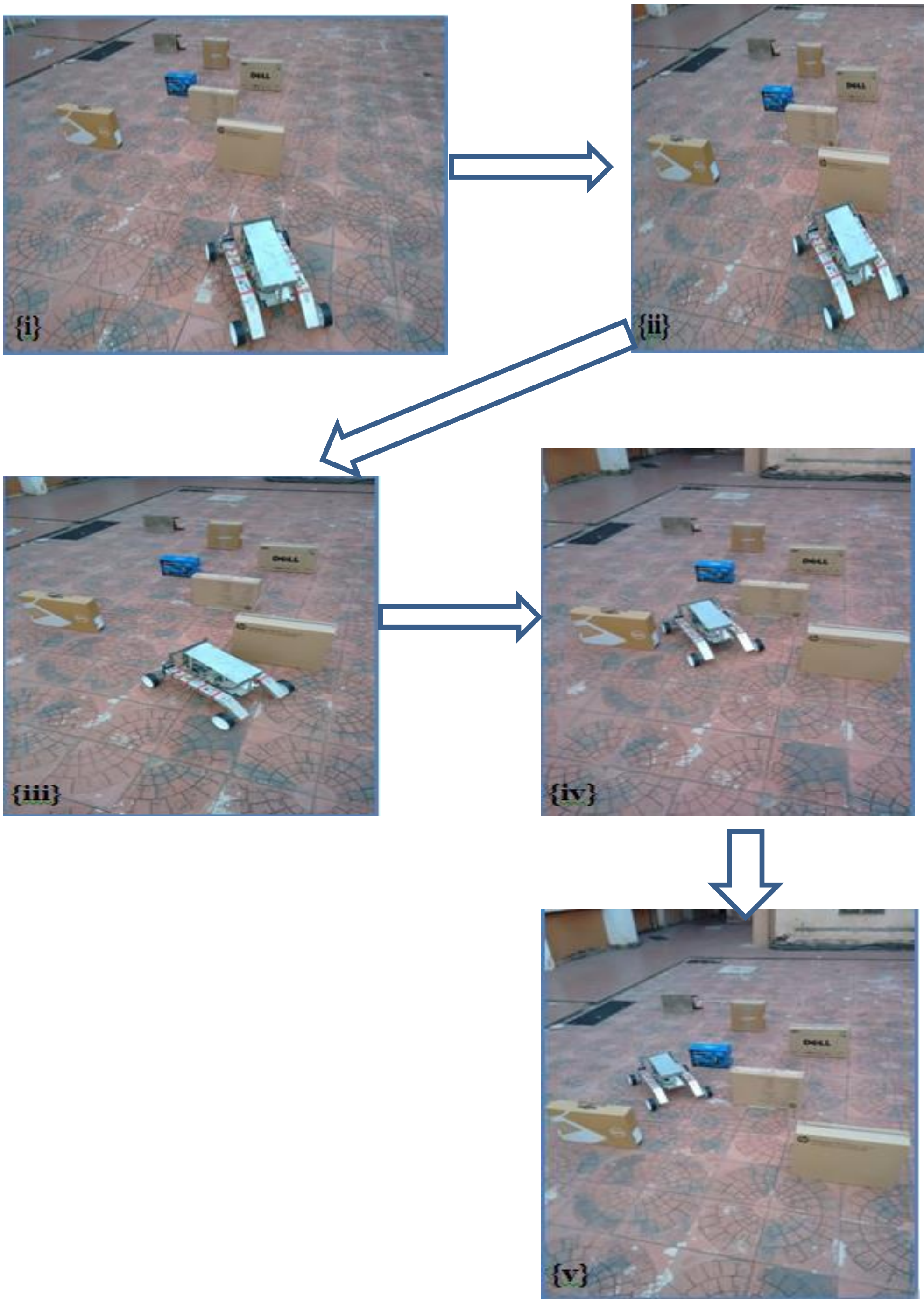


Figure34. Real–world robot movement based on neural networks.

This project uses feed-forward back propagation neural network for the mobile robot's path planning. The functional aspects of the suggested neural network were experimented in some workspaces. Initially experiments were conducted in a surrounding meant for training. It was observed that the robot avoids all obstacles. And it traverses each path from the given initial position to the target position safely. Next experiments were done in the unknown surrounding for the robot. These simulation results have been shown in the previous chapter. The above figure shows the navigation of robot in an unknown environment. Also the simulation experiments showed that the algorithm is effective for obstacles of any shape, number and location. Also the location of goal does not affect the effectiveness of the algorithm.

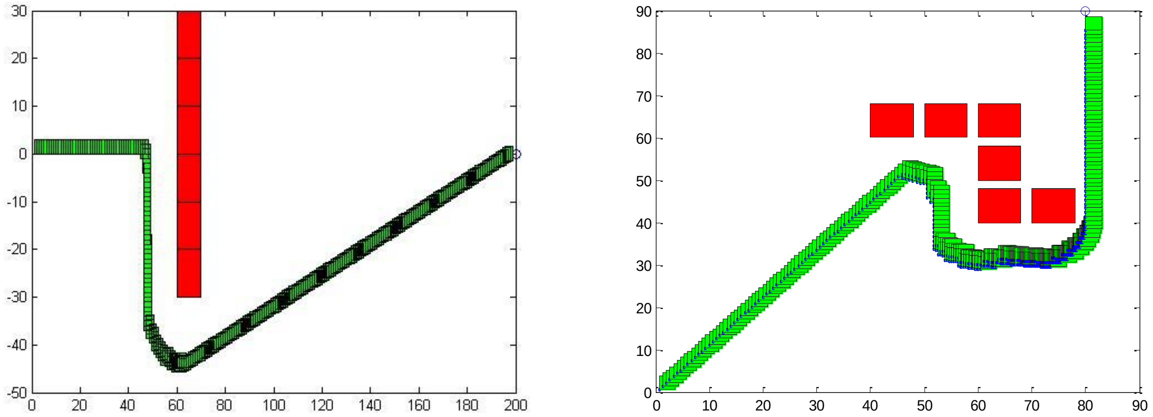


Figure35. Robot movement in varying environments.

This algorithm has been used in static environment. But this algorithm can work well in dynamic environment as well because of continuous inputs from the sensors.

CHAPTER-8

CONCLUSION

The project work conducted presents the findings that we got by using the feed-forward back propagation neural network in proposing a path planning for the mobile robot by utilizing inputs received from sensory data. The simulation examples show that the proposed algorithm is acceptable for solution for optimum path planning. Also the real-time experiment results confirmed the same by comparing online and offline path traversed which came about to be approximately same. Also by applying this algorithm, with simple control rules, robot was able to from straying away to un-optimized paths. However, it fails if the goal is closely surrounded by obstacles on all sides. But, his problem can be fixed by judiciously controlling various parameters.

FUTURE SCOPE OF PROJECT

In this project we have used Feed Forward Back Propagation Algorithm for a static environment. In future this work may be conducted by using other neural networks like radial basis functions, cascaded network, regression network, etc. Also we can use other intelligent techniques like genetic algorithms, fuzzy logic, etc. Another variation could be to navigate the robot in a more complex environment – moving target, moving obstacles, maze complexes, traps. Even in this algorithm one can try to reach the obstacle in spite of being heavily surrounded by obstacles. We can work on multiple-robot to multiple-target problems.

This work is for overland robot. Similar works may be conducted for underwater vehicles as well as for unmanned aircrafts.

REFERENCES

1. Fu K.S., Gonzalez R.C., Lee C.S.G. ROBOTICS – Control, Sensing, Vision, and Intelligence. Singapore, McGraw Hill, 1987.
2. Ge S.S., Lewis F.L. Autonomous Mobile Robots – Sensing, Control, Decision Making and Applications. Boca Raton, Taylor and Francis Group, 2006.
3. Sivanandam S.N., Sumathi S., Deepa S.N. Introduction to Neural Networks using Matlab 6.0.New Delhi, Tata-McGraw Hill, 2006.
4. Nehmzow U. Mobile Robots – A Practical Introduction. London, Springer, 2003.
5. Rajput R.K. Robotics and Industrial Automation. S. Chand & Company Ltd., New Delhi, 2008.
6. Aguirre E., Gonzalez A. Fuzzy behaviors for mobile robot navigation: design, coordination and fusion, International Journal of Approximate Reasoning, 25 (2000): pp. 255-289.
7. Faisal M., Hedjar R., Sulaiman M. A., Al-Mutib K. Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment, International Journal of Advanced Robotic Systems, 10 (2013): pp. 1-7.
8. Buniyamin N., WanNgah W.A.J., Sariff N., Mohamad Z. A Simple Local Path Planning Algorithm for Autonomous Mobile Robots, INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, Issue 2, Volume 5, 2011: pp.151-159.
9. Mohanty P. K., Parhi D. R. A New Intelligent Motion Planning for Mobile Robot Navigation using Multiple Adaptive Neuro-Fuzzy Inference System, Applied Mathematics & Information Sciences An International Journal, 5 (2014): pp. 2527-2535.
10. Janglová D. Neural Networks in Mobile Robot Motion, International Journal of Advanced Robotic Systems, Volume 1 Number 1 (2004): pp. 15-22.
11. Kumar M.P.S., Rajasekaran S. A Neural Network based Path Planning Algorithm for Extinguishing Forest Fires, International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012: pp. 563-568.
12. Zou X., Ge B., Sun P. Improved Genetic Algorithm for Dynamic Path Planning, International Journal of Information and Computer Science, Volume 1, Issue 2, May 2012: pp. 16-20.

13. Bhattacharya S., Talapatra S. Robot Motion Planning Using Neural Networks: A Modified Theory, International Journal of Lateral Computing, Vol.2, No.1, December 2005: pp. 9-13.

14. Mohanty P.K., Parhi D.R. Controlling the Motion of an Autonomous Mobile Robot Using Various Techniques: a Review, Journal of Advance Mechanical Engineering, 1 (2013): pp. 24-39.

15. Dadelahi S., Motlagh M.R.J., Shahri A.R.M. Obstacle Displacement Prediction for Robot Motion Planning and Velocity Changes, International Journal of Information and Electronics Engineering, Vol. 3, No. 3, May 2013: pp. 270-274.

16. Reina A., Gambardella L.M.,Dorigo M., Di Caro G.A., zePPeLIN: Distributed Path Planning Using an Overhead Camera Network, International Journal of Advanced Robotic Systems, (2014): pp. 1-22.

17. Zakaria M.,AL-Shebany M.,Sarhan S. Artificial Neural Network : A Brief Overview, International Journal of Engineering Research and Applications, Vol. 4, Issue 2(Version 1), February 2014: pp.07-12.

18. Pezeshki S., Badalkhani S., Javadi A. Performance Analysis of a Neuro-PID Controller Applied to a Robot Manipulator, International Journal of Advanced Robotic Systems, 9 (2012): pp. 1-10.

19. Cui M., Sun D., Liu W.,Zhao M., Liao X., Adaptive Tracking and Obstacle Avoidance Control for Mobile Robots With Unknown Sliding, International Journal of Advanced Robotic Systems, Vol 9, 171: 2012, pp. 1-14.

20. Priyadharson S.M. A., Kumar T.R., Saravanan M.S., ThilipKumar C.,Dileepan D., NEURAL CASCADED WITH FUZZY SCHEME FOR CONTROL OF A HYDROELECTRIC POWER PLANT, American Journal of Applied Sciences 11 (9), 2014: pp. 1659-1670.

21. Himavathi S., Muthuramalingam A., Venkadesan A., Sedhuraman K. NONLINEAR SYSTEM MODELING USING SINGLE NEURON CASCADED NEURAL NETWORK FOR REAL-TIME APPLICATIONS, ICTACT JOURNAL ON SOFT COMPUTING, APRIL 2012, VOLUME: 02, ISSUE: 03: pp. 309-318.

22. Kumar T.S., Sivanandam S.N. An Improved Approach for Detecting Car in Video using Neural Network Model, Journal of Computer Science 8 (10), 2012: pp. 1759-1768.

23. Jie S., Li W., WeiSong Z., YaQin Z., Malekian R. Numerical Analysis of Modeling Based on Improved Elman Neural Network, The Scientific World Journal, Article ID 271593, (2014): pp.1-12.

24. Celikoglu H.B., Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling, Mathematical and Computer Modelling, 44 (2006): pp.640–658.

25. Vaithiyanathan V., Rajeswari K., Nivethitha N., Shreeranjani Pa., Venkatraman G.B., Ahmed M.I., COMPARING THE IMPACT OF ACCURATE INPUTS ON NEURAL NETWORKS, International Journal of Engineering and Technology, Vol 5 No 3, Jun-Jul 2013: pp. 2054-2058.

26. Hannan S.A., Manza R.R., Ramteke R.J. Generalized Regression Neural Network and Radial Basis Function for Heart Disease Diagnosis, International Journal of Computer Applications, Volume 7– No.13, October 2010: pp. 7-13.

27. Ozyilmaz L., Yildirim T., Koklu K. Comparison of Neural Algorithms for Function Approximation, Pakistan Journal of Applied Sciences 2(3), 2002: pp. 288-294.

28. Ćosić A., Šušić M., Katić D. Advanced Algorithms for Mobile Robot Motion Planning and Tracking in Structured Static Environments Using Particle Swarm Optimization, SERBIAN JOURNAL OF ELECTRICAL ENGINEERING, Vol. 9, No. 1, February 2012: pp. 9-28.

29. Huang Y. Advances in Artificial Neural Networks – Methodological Development and Application, Algorithms 2009, 2, pp. 973-1007.

30. MALLESWARAN M., Vaidehi Dr. V., ANGEL DEBORAH S. Data Fusion Using Different Activation Functions in Artificial Neural Networks for Vehicular Navigation, International Journal of Engineering Science and Technology, Vol. 2(12), 2010: pp. 7676-7690.

31. Senapati M.R., Vijaya I., Dash P.K. Rule Extraction from Radial Basis Functional Neural Networks by Using Particle Swarm Optimization, Journal of Computer Science 3 (8), 2007: pp. 592-599.

32. Baby Anitha, E. and K. Duraiswamy. A HEURISTIC MOVING VEHICLE LOCATION PREDICTION TECHNIQUE VIA OPTIMAL PATHS SELECTION WITH AID OF

GENETIC ALGORITHM AND FEED FORWARD BACK PROPAGATION NEURAL NETWORK, Journal of Computer Science 2012, 8 (12): pp. 2008-2016.

33. MAHMOUD O., ANWAR F., E. SALAMI M.J. LEARNING ALGORITHM EFFECT ON MULTILAYER FEED ORWARD ARTIFICIAL NEURAL NETWORK PERFORMANCE IN IMAGE CODING, Journal of Engineering Science and Technology, Vol. 2, No. 2 (2007): pp. 188 – 199.
34. Suresh A., Latha S.S., Nair P., Radhika N. PREDICTION OF FIGHT OR FLIGHT RESPONSE USING ARTIFICIAL NEURAL NETWORKS, American Journal of Applied Sciences 11 (6), 2014: pp. 912-920.
35. Solanki S., Jethva H.B. Modified Back Propagation Algorithm of Feed Forward Networks, International Journal of Innovative Technology and Exploring Engineering, Volume-2, Issue-6, May 2013: pp 131-134.
36. Sharma A., Chaudhary D.R. Character Recognition Using Neural Network, International Journal of Engineering Trends and Technology, Volume4 Issue4- April 2013 : pp. 662-667.
37. Zhen-Guo Che, Tzu-An Chiang, Zhen-Hua Che. FEED-FORWARD NEURAL NETWORKS TRAINING: A COMPARISON BETWEEN GENETIC ALGORITHM AND BACK-PROPAGATION LEARNING ALGORITHM, International Journal of Innovative Computing, Information and Control, Volume 7, Number 10, October 2011: pp. pp. 5839-5850.