Self-Learning Robot: A robot that learns to follow motion command using Reinforcement Learning

Dissertation Report

Submitted to
Indian Institute of Technology(ISM), Dhanbad
in fulfilment of the requirements for the award of the degree

of

Master of Technology
in
Computer Science & Engineering

by

Yeshvendra Kumar Singh

Admission No.:16MT001369

Under the guidance
of
Dr. Haider Banka
Associate Professor
(Dept. of Computer Science and Engineering)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Indian Institute of Technology (Indian School of Mines), Dhanbad-826004

MAY 2018

ACKNOWLEDGEMENT

I would like to express my deepest sense of gratitude towards **Dr. Haider Banka**, Associate Professor, Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, under whose guidance I have learnt a great deal. He has always been inspiring and has always catered to my doubts and problems with solutions. I thank you sir for your guidance.

I would also express my gratitude towards the Head of the Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, **Dr. Prasanta K. Jana** and our course coordinator **Dr. Sachin Tripathi,** for their support and for providing a hassle free working environment by allowing me to access the M-Tech Project Lab at any time of the day.

I express my gratitude towards all the faculty members for their encouragement, support and blessings. I would also like to express my gratitude towards my batch-mates for their support and well wishes.

Lastly, I would like to thank my family members for their constant support and inspiration and for having enormous faith in me.

Yeshvendra Kumar Singh

Admission No: 16MT001369
M-Tech (Computer Science and Engineering)
IIT(ISM) Dhanbad

CERTIFICATE

This is to certify that dissertation entitled **"Self-Learning Robot: A robot that learns to follow motion command using Reinforcement Learning"** by **Mr. Yeshvendra Kumar Singh (Admission No. 16MT001369),** a student of M Tech in Computer Science and Engineering, is a record of bonafide work carried out by the candidate under my supervision in the Department of Computer Science and Engineering at Indian Institute of Technology (Indian School of Mines), Dhanbad.

The dissertation submitted is in fulfilment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering from Indian Institute of Technology (Indian School of Mines), Dhanbad.

To the best of my knowledge, the work carried out has not been submitted elsewhere for award of any degree.

Dr. Prasanta K. Jana	**Dr. Haider Banka**
(Professor)	(Associate Professor)
Head of the Department	Guide
Department of CSE	Department of CSE
IIT (ISM), Dhanbad	IIT (ISM), Dhanbad

DECLARATION

I hereby declare that this thesis is an authentic record of research work carried out by me. To the best of my knowledge it contains no material previously published or written by another person or material which has been accepted of any degree or diploma of the university or other institutes of higher learning, my due acknowledgment has been made in the text.

Date:

Yeshvendra Kumar Singh

Admission No: 16MT001369
M-Tech (Computer Science and Engineering)
IIT(ISM) Dhanbad

ABSTRACT

Robotics is a discipline that has large number of problems attached to it, depending on different types of domains, hardware, technology and tasks. Working in different environment proposes different types of challenges like working in a controlled environment as that of industries will have different types of challenges then working in an open environment. For rapid development of robotic systems reusability of codes, robustness and long period of maintainability is very important. The novel approach of providing a general self-learning algorithm which could help robot learn and follow the motion command of instructor without human intervention can help us aid in solving this problem.

In this project we have used reinforcement learning to teach a dumb robot (i.e. a robot which doesn't know how to move forward, left, right and backward) how to move forward, left, right and backward without human intervention. The technique of reinforcement learning helps the robot to interact with the environment with the help sensor attached to it and rewards the robot with a feedback depending upon if the robot is doing the right action for right command or not. So, by the process of trial and error after many episodes of learning the robot ultimately learns how to follow the command of moving forward, left, right and backward.

Table of Content

List of Figures

Chapter 1

Introduction

Application of robots are becoming more and more common each day. Some of the advanced artificial intelligence aided robots such “Sophia” have made it clear that this is what our future holds for us. But when we look at whatever efforts we have put, in order to achieve that level of advancement, then it reflects that, it could be done in much more easier way, if we could give robots the power of self-learning. Here as a part of this project, we will use reinforcement learning to make a robot learn to follow motion commands like move forward, left, right and backward even though the robot doesn’t know how it can be done. By using reinforcement learning the robot will act as an agent and interact with the environment i.e. the surrounding to learn how it can obey the motion command in an efficient manner.

In this project we will use reinforcement learning technique to tackle the complexity of finding the combinations of motor to follow the command i.e. forward, backward, left and right. A robot having ‘k’ degree of freedom have (2k – 1)! possible combination of motions that can be performed. Choosing from among the (2k – 1)! possible actions the right combination for moving forward, backward, left and right is a complex task as ‘k’ increase. Suppose you have a robot which has 6 degree of freedom so the total possible combination for 6 degree of freedom will be (2*6 – 1)! i.e. 11! Which is equal to 39916800 possible combinations. Among these three crore ninety nine lakh sixteen thousand and eight hundred possible combinations only four will be the most efficient ways. So assigning these four most optimal combinations to each command will be like find a needle in the hay stack.

Many a times it has been observed that an ongoing mission has greater chance of being jeopardised because of the malfunctioning of a hardware. However this can be overcome if we give robots participating in the mission, the capability of self-learning with whatever resources the robot have with him. In this project by using reinforcement learning we will build a robot which in case of a hardware malfunction could again self-teach himself to do a particular task without any human interaction. Let us take an example to understand it in a better way. Suppose a robot is sent to mars for exploration of the surface. Somehow a motor of robot malfunctions hampering the job assigned to the robot. In this case if the robot could figure out a way by which it could complete the assigned job then it would save a massive amount of money and time of the researchers.

Self-Learning capability, if given to a robot can be very useful as it would save time of the research and make this monotonous job of figuring out the way to make robot much easier.

1.1 Motivation

Human being are said to be the most intelligent species who has ever lived on earth and this is because we human beings learn by our prior experience. We take appropriate decision by interacting with the environment and also use our history of experiences. Imagine a new born child, when he/she tries to walk for the first time, he/she start trying different ways to walk. Sometimes the new born child falls, sometimes he/she succeed in walking. In order to learn new things humans use trial and error type of technique in which they might fail sometime or they might succeed. But by this experience they eventually figure out a way by which they can achieve there object, which in this case is to walk. Similarly if we give robots the ability to self-learn and the ability to figure out the ways to do a task then we would save a huge amount of money and resources which we would otherwise spend in order to accomplish the task.

Reinforcement learning is a powerful techniques which help agents to figure out the optimal solution to the problems given. In this case the robot can interact with the environment and figure out different way by which it could follow the given command i.e. move forward, turn left, turn right and move backward. So in order to develop such a skilled robot we have used reinforcement learning technique i.e. Q-Learning for the learning process of robot. The robot learns by trial and error technique in which it tries different combination of motions to figure out which one is correct for which motion command. As a feedback a gyro sensor and an accelerometer is attached to the robot so that robot gets to know if the action the robot is performing is correct or not according to the instruction given.

1.2 Related Works

The reinforcement learning was developed by Sutton and Barto [1] which is now being used very often in artificial intelligence related projects. Reinforcement Learning has come so far that now this powerful tool has even proven more powerful by defeating humans in Atari console games and many other similar games. Reinforcement has been successfully applied in different fields like controlling robots, scheduling of elevators, checkers, etc.

Dayan and Hinton [2] suggested to speed up the process of reinforcement learning by sub dividing the tasks into sub tasks which will be governed by managers and sub managers. This enables the idea of simultaneous learning and hence speeding up the process of learning. They demonstrated the proposed system using a simple maze task.

Researcher have also used reinforcement to do complex task in which there are more than one agent interacting with the environment. A similar approach was follow by Littman [3] in which while learning process of one agent he assumed the other agent to be either a friend or a foe. He also stated that Q-Learning provides more convergence as compared to Nash-equilibrium learning rules. Sallans and Hinton [4] in another paper has also used

graphical modelling literature techniques to estimate q - value and apply good actions in order to cater high dimensional state and actions operating in a complex environment.

Martinez and Perez [5] tried to justify the use of bio inspired multi-agent system to control robots. In this paper they have used the peculiarities, organizational and functional structure of brain to propose a robot control system. They have also suggested the use of expected results i.e. expected reward for doing some action so that the system could automatically select the most optimal actions for attaining the objective.

1.3 Machine Learning

Machine Learning is the technique by which computer act and learn like humans. It also enables computer to improve their learning process over time. Machine Learning technique takes input in the form of observations and interactions with real-world environment and then makes a prediction or decision about a given thing.

Machine Learning is the science of building a machine to act without being explicitly programmed. In the past years, machine learning has helped us to develop self-driving cars, speech recognition, effective web search, etc. Machine Learning is so vastly used today that you probably use it many a times a day without knowing it. Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new – but one that has gained fresh momentum.

Machine Learning algorithms can be broadly divided into 3 different types, based on learning styles which are: supervised learning, unsupervised learning and reinforcement learning.

1.3.1 Supervised Learning

Supervised Learning is a type of Machine Learning which is applied on the labelled dataset. This process of learning is called supervised because we already know the class/result of the training dataset. In this type of learning technique we try to find out the dependency of a variable (x) over the output variable (y). The sample equation is as follows:-

$$Y = f(X)$$

In supervised learning we try to find the above sample equation, so that if we have a new sample input data we could accurately predict the sample output. Some of the examples of Supervised Learning algorithms are Linear Regression, Logistic Regression, Decision Tree, etc.

1.3.2 Unsupervised Learning

In unsupervised learning the training dataset in not labelled i.e. it doesn't have already know class/result. In the training process of unsupervised learning we study the structure or distribution of the training dataset to deduce an output. This learning technique is called unsupervised learning because unlike the supervised learning process we don't take help of the pre-established result to supervise the training model. Some of the popular unsupervised learning techniques are k-means clustering, KNN algorithm, etc.

1.3.3 Reinforcement Learning

Reinforcement Learning or in other words RL is a type of machine learning approach which was inspired by behavioural psychology. In this type of learning technique an agent learns to take some action by interacting with the environment so as to maximize the reward generated by that action.

Chapter 2

Reinforcement Learning

Reinforcement Learning is a learning technique in which the agent learns what decisions to make with respect to its environment, this is done by taking some action and getting reward according to the result. Imagine a trainer training a dog in the garden. When the dog obeys the order of the trainer he gets a treat (+1 reward) and if he doesn't obey the order he is scolded upon (-1 reward). This is how the dog understands that obeying the order is a good thing and not obeying the order is a bad thing. Similarly in reinforcement learning agent learns what is good for him and what is bad for him based on the reward it gets by interacting with the environment. It may be a positive reward stating that the action agent is taking is the right action or a negative reward stating that the action agent is taking is the wrong action.

Reinforcement Learning comprises of an agent which interact with the environment through actions to get to a state and gets feedback as a reward. The whole concept is demonstrated in below given image, where R_t is the reward at time t, S_t is the state at time t, A_t is the action at time t and we have one agent and an environment to which the agent interacts.

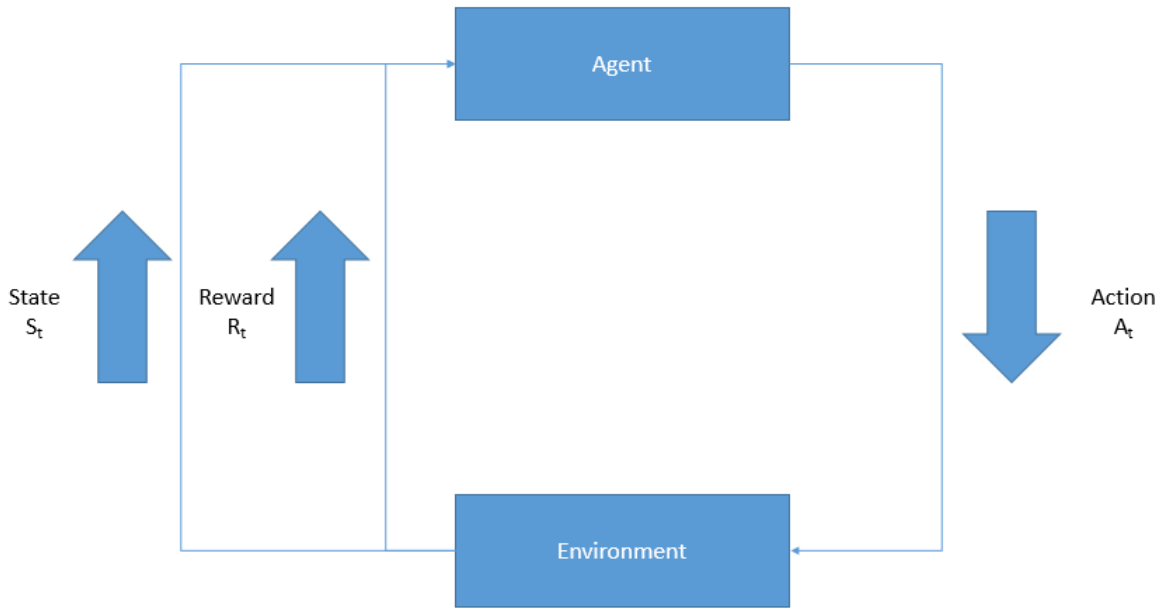


Figure 1: Working of reinforcement learning

By our natural experiences we learn by iteract with the environment. Imagine a small child sitting in a living room on a winter night. The child sees a fireplace and he approaches it. He feels warm, it is a positive effect, he feels good. So he now understands that the fireplace is a good thing (positive reward). But then suddenly the child thinks of touching the fire. This burns his hand (negative reward). This action makes him understand that fire is good when he is at approriate distance form the fire as it produces warmth but when if you touch it you will get burned.

2.1 Difference between Reinforcement Learning and Supervised Learning

The most important difference of supervised learning and reinforcement learning is the reward value which is returned as a feedback when agent interacts with the environment where as in case of supervised learning the model learns by training over a pre-established dataset, in short we don't have training data in reinforcement learning. In reinforcement learning we don't tell the agent which action to take but with the help of trial and error the agent must figure the action out which fetches him the best result. Let us take an example to understand it better. In case of supervised learning if a dog has to fetch a ball you give him the instruction that you have to move 10 steps forward pick up the ball and bring it back to me by again moving 10 steps backward. But when we consider reinforcement learning, you leave the dog around and let him do whatever he wants to do. Whenever the dog fetches you the ball you give him a treat as a reward telling him you have done a good work. In this process you hope eventually he will learn that fetching a ball is a good thing.

2.2 Difference between Reinforcement Learning and Unsupervised Learning

In unsupervised learning we try to find the structure in between the dataset i.e. we categories the dataset depending on the attribute of the dataset and its relationship with each other where as in reinforcement it is not the case as there is no dataset involved. In reinforcement learning an agent interacts with the environment and if he takes an appropriate action according to the state in which the agent is, he is rewarded for that action. This process makes him want to get more and more positive rewards eventually making him learn that this is the appropriate action for this particular state.

2.3 Why maximizing the expected cumulative reward, the goal of an agent?

In reinforcement learning to get the optimal most result we need to maximize the expected cumulative reward. Cumulative reward at time t can be given as

$$G_t = R_{t-1} + R_{t-2} + \_ \_ \_ \_ \_$$

$$= \sum_{k=0}^{t} R_{t+k+1}$$

Where G_t denotes the cumulative reward at time t and R_t also denotes the reward at time t.

2.4 Significance of discount rate in reinforcement learning

In reinforcement learning to capture the significance of future rewards we define a discount rate called gamma. The value of this discount rate is always in between 0 and 1. The greater the value of discount rate the lesser will be the discount i.e. if discount is less, the learning agent cares more about the reward he will be getting in long run. Otherwise, the lesser the discount rate greater the value of discount i.e. if discount is more, the learning agent cares less about the long term reward than the short term reward.

Therefore, the discounted cumulative expected reward is

$$G_t = \sum_{k=0}^{t} \gamma^k R_{t+k+1}$$

$$= R_{t-1} + YR_{t-2} + \gamma^2 R_{t-2} + \_ \_ \_ \_ \_$$

Where, γ denotes the dicount rate and it is always in between 0 and 1, G_t denotes the cumulative reward at time t and R_t also denotes the reward at time t.

2.5 Types of tasks in reinforcement learning

The tasks in reinforcement learning can be broadly divided into two different categories and they are:-

1. Episodic Tasks
   In episodic task, we have a definite starting point and a definite ending point. Hence, the episode comprises of a list of Rewards, Actions, States and Next States. Let's take

an example to understand it better. Imagine a navigation robot in which the robot has to be trained to reach a certain position. So, the robot starts at a given starting position and the task ends when the robot reaches the destination position. This process can be repeated over and over again. This type of task is called as episodic task.

2. Continuous Tasks
   Continuous tasks are those tasks that continue for indefinite period of time. For example, consider an agent which is used to automate the process of stock trading. In this type of task there is no definite starting and ending point. The task continuous until someone decides that it is no more needed. This type of task is called continuous task.

2.6 Types of learning approach in reinforcement learning

There are two basic types of learning approach in reinforcement learning namely Monte Carlo Approach and Temporal Difference Learning.

1. Monte Carlo Approach
   In Monte Carlo Approach at the end of an episode the agent analyse the total cumulative reward to see how efficiently the agent performed. In this type of learning the rewards are received when the agent reaches the goal in other words at the end of game. After this one iteration we start another similar kind of iteration but with the improved knowledge from the previous iterations. This way the agent get better and better in making decision as number of iteration increases. The above statement can be represented as follows:

$$V(S_t) \leftarrow V(S_t) + \alpha\,[G_t - V(S_t)]$$

   Where the left most $V(S_t)$ describes the maximum expected future reward, the $V(S_t)$ on right side denotes the maximum expected future reward of previous iteration, α denotes the learning rate and G_t denotes the discounted cumulative reward.

2. Temporal Difference Learning
   In Temporal Difference Learning the agent does not wait for the end of iteration to analyse the maximum expected future reward where as it updates the estimated maximum expected future reward at every non-terminal state. The above statement can be demonstrated as follows:

$$V(S_t) \leftarrow V(S_t) + \alpha\,[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

Where the left most $V(S_t)$ describes the maximum expected future reward, the $V(S_t)$ on right side denotes the maximum expected future reward of previous iteration, α denotes the learning rate, γ denotes the discount rate, R_{t+1} denotes the observed reward at time t+1 and $V(S_{t+1})$ also denotes estimated maximum future reward at time t+1.

2.7 Approaches to reinforcement learning

There are four basic approaches of reinforcement learning which can be combined further to make more complex approaches. The list of basic approaches are as follows:

1. Value Based Approach

   In value based reinforcement learning we focus on optimizing the value function i.e. $V(S)$. This function denotes the maximum expected future reward that the agent will get in every iteration. So, at the end of the learning process the agent will use this value to determine what decision to take at every step. The agent will always try to choose the bigger value as bigger value means bigger reward and hence better results.

2. Policy Based Approach

   In the learning process it may happen that the policy converged earlier and value function on the other hand took some more time to converge, although we are only interested in the convergence of the policy. Here the concept of policy based approach comes into picture where we directly optimize policy function without even considering the value function. This lets us to achieve the best action for each state.

3. Model Based Approach

   In Model based approach the agent models the environment beforehand i.e. we emulate the behaviour of the environment in which the agent is interacting. After which the model either uses value based approach or policy based approach to get to the optimal policy.

4. Model – Free Approach

   This approach is opposite of what model based approach was i.e. in this approach the agent directly interacts with the environment to observe the state transition and reward function. Simultaneously this approach uses either value function or policy function to get to the optimal policy.

2.8 Q-Learning

Q-Learning is a reinforcement learning technique in which the agent tries to get to the optimal policy by observing the interaction of agent with the environment. In the process of Q-Learning we treat the history of interactions as a sequence of state (s), its action (a), the reward (r) received by taking the action (a) on state (s) and the next state (s'). These history of interaction will be the data from which agent will get to know what to do.

Q-Learning uses temporal difference learning to estimate the value function which is denoted as Q(s, a). So according to the temporal difference learning the formula of value function becomes

$$Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma max_{a'} Q[s',a'] - Q[s,a])$$

Where $Q[s,a]$ denotes cumulative discounted reward at state s and action a, $Q[s',a']$ is the cumulative discounted reward at state s' which is the next state and action a' which is the next action, α is the learning rate, r is the reward and γ is the discount rate. Q-Learning learns the optimal or the best possible policy irrespective of the policy which is being followed by the agent as long as there is no limitation in number of iterations.

2.9 Exploration and Exploitation

In reinforcement learning to learn about the optimal solution of the problem we use the concept known as exploration and exploitation. In exploration we try new things i.e. those thing which we have never tried before in order to achieve the most optimal solution. Whereas in exploitation we try to get the most out of whatever we have experienced in the past in order to achieve the most optimal solution.

Let's take an example to understand it in a better way. Suppose you have large number of slot machines. Now you as the player of the game wants to maximize the reward in a short period of time. One approach to this problem would be to select a single slot machine and keep trying throughout the day. With this approach you may win a prize but a lot of time is wasted in the process. So this approach can be thought of as a pure exploitation of things. The other approach one may take is pull the lever of all the available slot machine and it may happen that you may win a prize. But even this approach will take a large amount of time making this process very cumbersome. This approach is can be imaged as a pure exploration approach.

But both the approaches individually are not optimal. We should find a proper trade off in between these two approach to get the maximum reward and this problem is known as the exploration vs. exploitation dilemma.

2.10 Applications of reinforcement learning

Practical applications of reinforcement learning are as follows:-

1. Manufacturing Industry

   With the help of reinforcement learning a robot can learn to pick up an object from on container and put it in other. Depending on its success and failure it learns how to do this job with better speed and precision.

2. Inventory Management

   Reinforcement learning can also be used for inventor management to find out the optimal space utilization.

3. Delivery Industry

   Reinforcement learning is also used in delivering the packages to customers in the most optimal way i.e. to increase the efficiency of delivery without increasing the delivery vehicles.

4. Finance Industry

   Trading strategy can also be decided using reinforcement learning. The use of reinforcement learning is turning out to be an effective way for optimizing financial objectives.

5. Advertisement Industry

   Reinforcement learning can also be used to find out if an ad is relevant to a customer or not. Depending on the feedback of the customer (i.e. clicks on the ads) ads are being displayed.

Chapter 3

Hardware Description & Interfacing

The project comprises of a self-learning robot which could teach himself to follow the motion command (i.e. forward, left, right and backward) as instructed by the instructor. In order to complete this project we have used some of the hardware and there description is as follows:-

3.1 Raspberry Pi 3 Model B+

Raspberry pi 3 is a small single-board computer developed by Raspberry Pi foundation. This single-board computer does not include the peripherals. Raspberry pi 3 has a Broadcom SoC with an inbuilt central processor and an on-chip graphics processor. The processing speed of raspberry pi 3 model B+ is 1.4 GHz with an on-board memory of 1 GB (RAM). Memory cards are used to store the operating system and program memory. It also supports lower level output from its GPIO pins which supports I^2C and similar other common protocols.

Figure 2: Raspberry Pi 3 Model b+

Raspberry pi 3 model B+ has a 64 bit quad core ARM Cortex-A53 processor with in built Wi-Fi, Bluetooth and USB booting capability. Raspberry pi uses Raspbian, which is a Debian-based Linux operating system. The other operating systems that are used in Raspberry pi are Ubuntu MATE, RISC OS, Windows 10 IoT Core, etc.

Some of the applications of Raspberry pi are:-

1. Education

   Government of countries in middle east has should a huge amount of interest in using Raspberry pi as a tool which could help the children learn the new technology in a much cheaper and efficient way. Courses like 'Picademy' are also being designed to so that people can be taught how to use this new computer technology.

2. Home Automation

   Raspberry pi is being used in home automation industry to help people build a cost effective way to monitor energy consumption in the house.

3. Industrial Automation

   Raspberry pi allows the use of computer module in a harsh environment such as industries. Industries are now willing to use it as an IoT solution and achieve their goals.

4. Commercial Products

   Some of the commercial products are also using Raspberry pi as its computing unit like Next Thing Co. used raspberry pi to create digital camera, Slice a digital media player also used raspberry pi as it processing unit.

In Raspberry pi 3 Model B+, 40 GPIO (General Purpose Input Output) pins are available, which is used for interfacing different types of sensors and actuators. The GPIO pins can work for both analog and digital type of signals. The given GPIO pins are directly interfaced with the microprocessor i.e. ARM Cortex A53 processor. A GPIO pin can either be used as an input pin or an output pin. Among all forty pins there are two pins which supplies 5 volt current, two other pins supply 3.3 volt current, there are seven ground pins and rest twenty nine pins can be used digital or analog input/output operations.

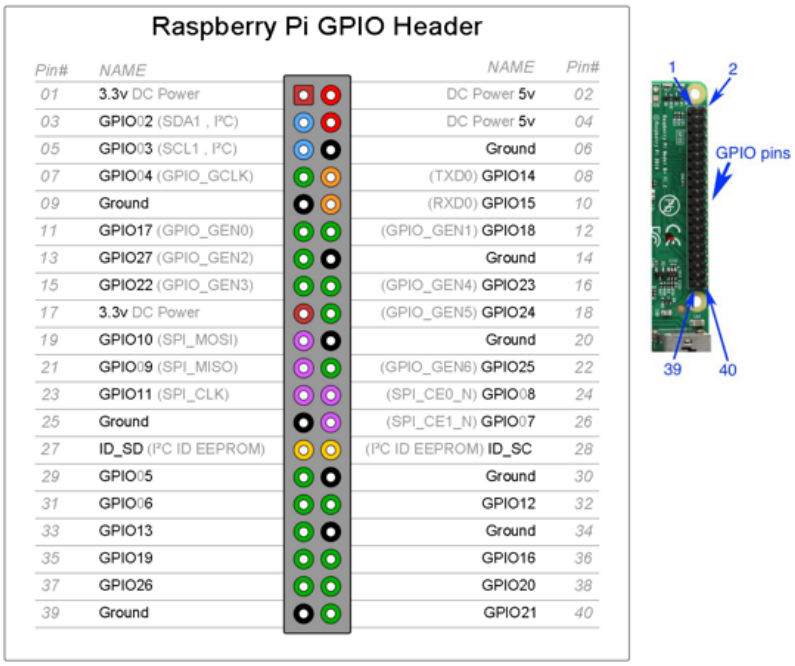


Figure 3: Pin description of Raspberry Pi 3 Model B+

3.2 MPU 6050 Sensor

MPU 6050 is 3 axis gyro sensor and 3 axis accelerometer module which helps in motion tracking. It is a low power, low cost and high performance sensor device.

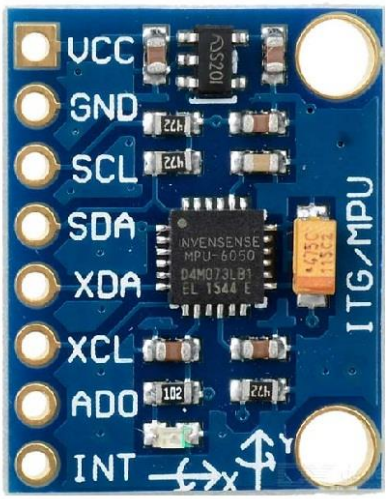


Figure 4: MPU 6050 sensor (3 axis gyro sensor and 3 axis accelerometer)

For accurate tracking of slow and fast motions, the sensor features a programmable gyro of range ±250, ±500, ±1000 and ±2000 degree per second and a programmable accelerometer of range ±2g, ±4g, ±8g and ±16g. Another inbuilt feature if MPU 6050 is the temperature sensor and on-chip oscillator. The sensor is said to be very accurate as it contains 16-bit analog to digital convertor for each channel. This makes it powerful enough to capture x, y and z axis readings at the same time. Even though it has both the gyro sensor and accelerometer, it is considered to be not so expensive.

MPU-6050 module have 8 pins and the description of each pin is as follows:-

1. INT pin is the interrupt digital output pin.
2. AD0 is the I2C slave address pin.
3. XCL is the auxiliary serial clock pin.
4. XDA is the auxiliary serial data pin.
5. SCL is the serial clock pin.
6. SDA is the serial data pin.
7. GND is the ground pin.
8. VCC is the power supply pin.

3.3 Motor Driving IC

L293D IC is commonly known as motor driving IC. This IC allows users to control working speed and the direction of 2 motors simultaneously. This is a 16 legged IC and it allows 5V to 36V of bidirectional driving current.

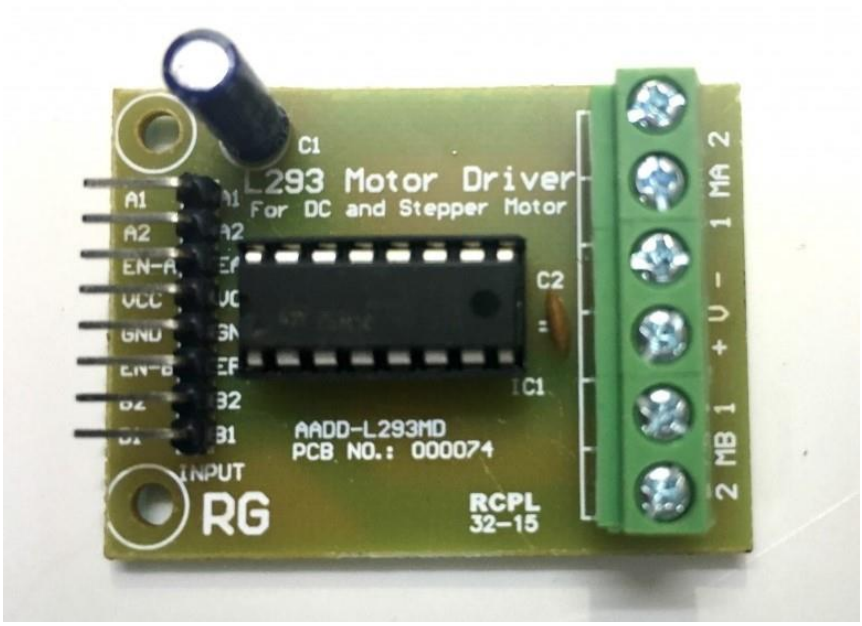
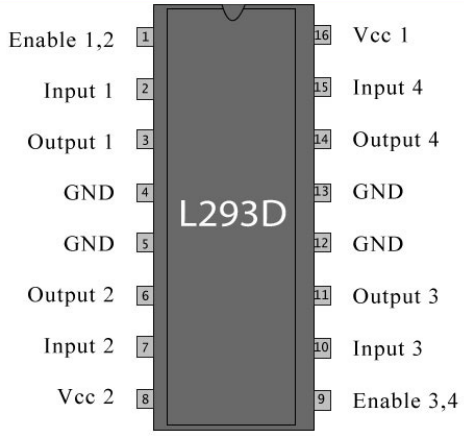


Figure 5: Motor Driving IC (L293D)

This motor driving IC works on the concept of H-bridge. This type of circuit allows the voltage to follow in both the directions. This property of L293D IC makes it suitable for driving the DC motors. Due to its small size it is very often used in robotics implementation for controlling DC motors.



Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	$V_{cc\ 2}$
9	Enable pin for Motor 2; active high	Enable 3,4
10	Input 1 for Motor 1	Input 3
11	Output 1 for Motor 1	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 1	Output 4
15	Input2 for Motor 1	Input 4
16	Supply voltage; 5V (up to 36V)	$V_{cc\ 1}$

Figure 6: Pin description of L293D Motor driving IC

L293D has a total of 16 pins, among which there are 4 input pins and four output pins. It has two enable pins, four ground pins, Vcc pin and the last VSS motor supply pin.

3.4 DC Motor

A DC motor converts the direct current into a mechanical energy. It uses the magnetic field to produce force which in turn can be used for motion of a body.

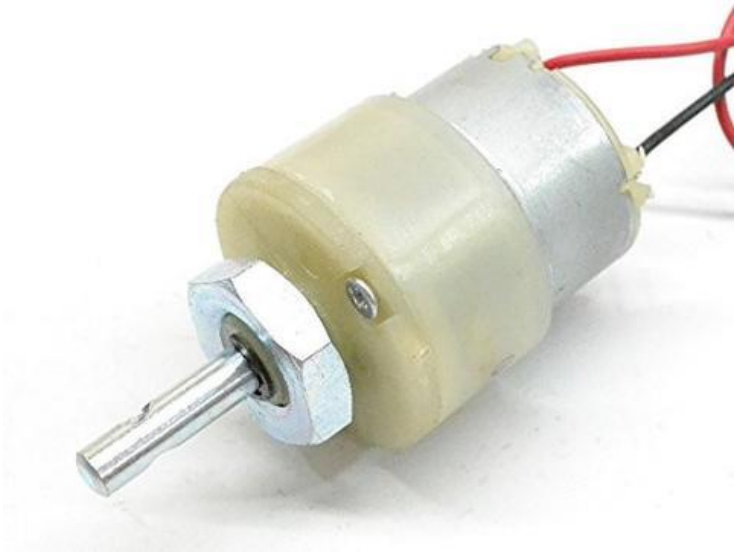


Figure 7: DC Motor

Almost every DC motor either uses the electromechanical or electronic mechanism to periodically change the direction of flow of current in the motor. The DC motor used in this project has specification of 12 volt and 4.5 amp on full load.

3.5 Interfacing with Raspberry Pi

In this project four pins are used for the interfacing of Raspberry Pi with the MPU 6050 gyro sensor and accelerometer. Eight pins are used for the connection with the DC motors and 3 pins are used for the connection with the LED which will denote if the action performed is right or wrong.

Table 1: Pins used for interfacing the sensor, motor and LED with Raspberry Pi

Raspberry Pi Pin Number	Usage
GPIO 26 Pin# 37	Clockwise movement of Back-Left motor
GPIO 19 Pin# 35	Anti-Clockwise movement of Back-Left motor
GPIO 5 Pin# 29	Clockwise movement of Front-Left motor
GPIO 6 Pin# 31	Anti-Clockwise movement of Front-Left motor
GPIO 20 Pin# 38	Clockwise movement of Back-Right motor
GPIO 21 Pin# 40	Anti-Clockwise movement of Back-Right motor
GPIO 12 Pin# 32	Clockwise movement of Front-Right motor
GPIO 16 Pin# 36	Anti-Clockwise movement of Front-Right motor
GPIO 23 Pin# 16	Input for red LED
GPIO 18 Pin# 12	Input for green LED
Ground Pin# 14	Ground for both the LEDs
3.3v DC Power Pin# 1	Vcc for MPU 6050
Ground Pin# 6	Ground for MPU 6050
SDA Pin# 3	SDA for MPU 6050
SCL Pin# 5	SCL for MPU 6050

3.6 Schematic Diagram

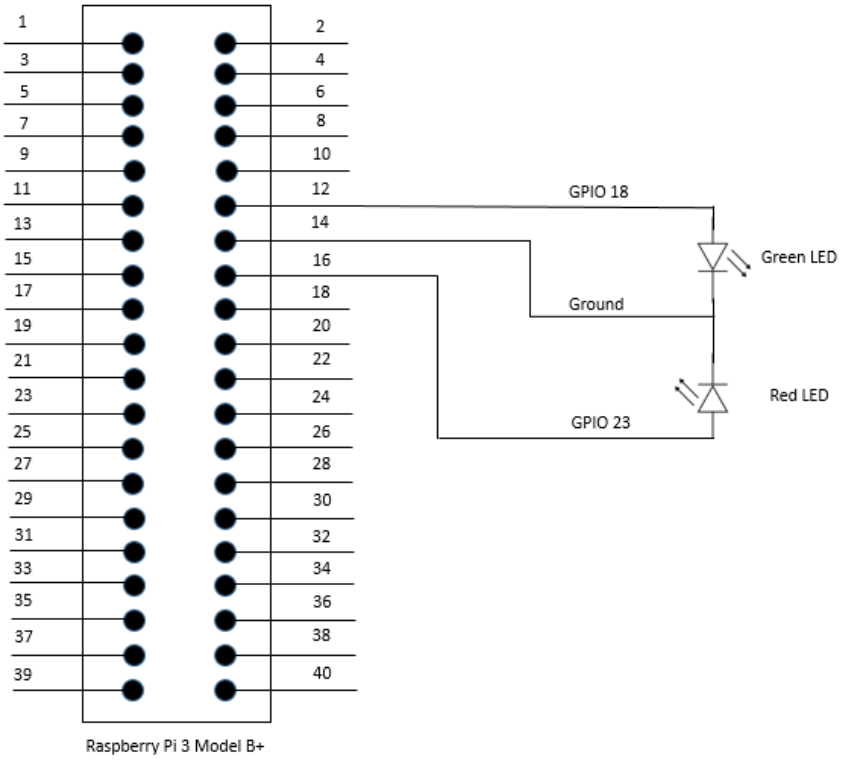


Figure 8: Schematic Diagram of LED and Raspberry Pi Interfacing

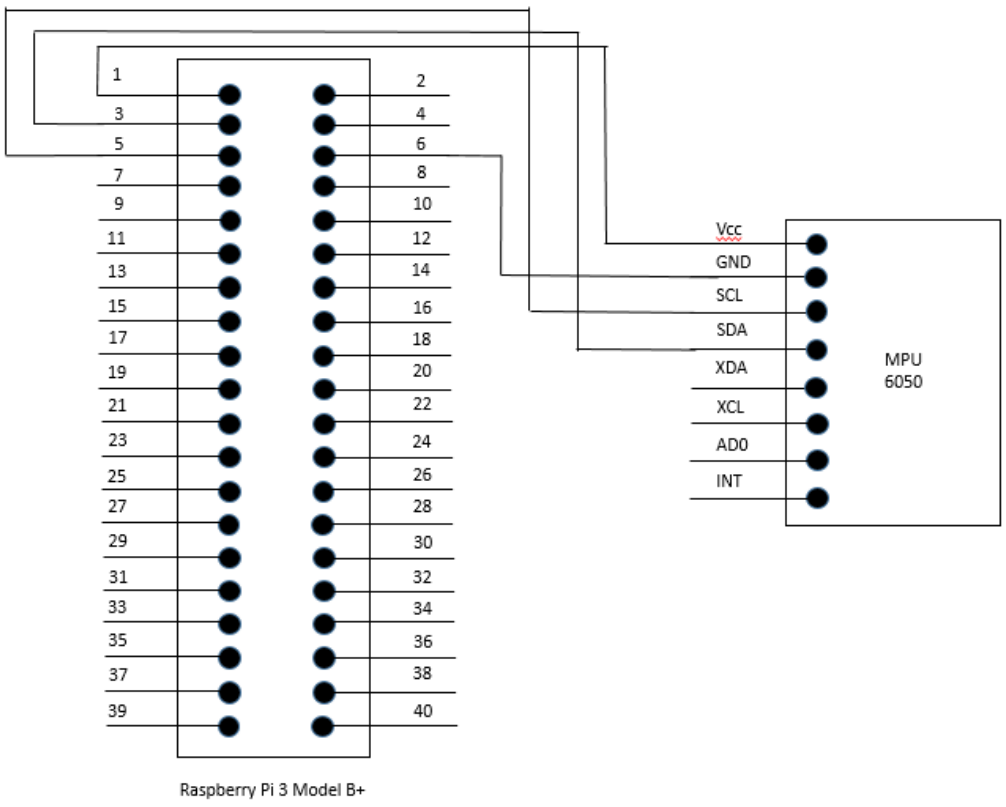


Figure 9: Schematic Diagram of MPU6050 and Raspberry Pi Interfacing

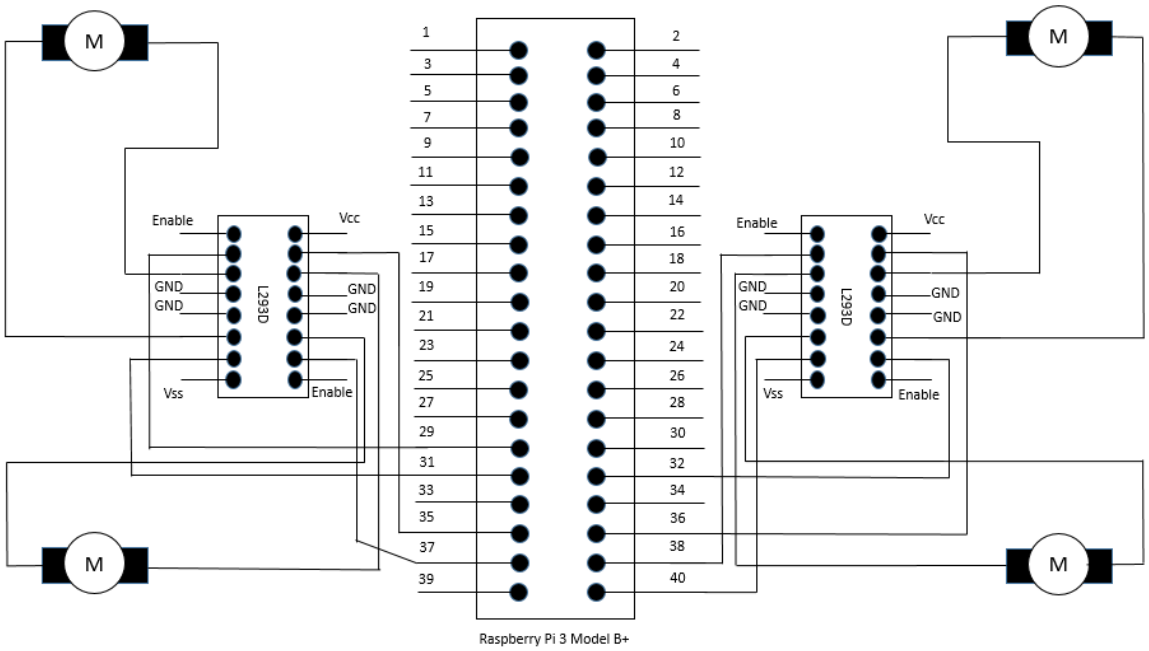


Figure 10: Schematic Diagram of L293D, DC Motor and Raspberry Pi Interfacing

Chapter 4

Implementation Details

In this project, to implement the concept of self-learning robot which could self-learn the motion commands of the instructor, we have assumed a robot which has 4 DC motors attached to it. So according to the formula $(2 * k - 1)!$ which gives us the total number of motions possible in a robot with k degree of freedom, we have a total of 5040 combination of motions that can be performed by this robot. Our object would be to make this robot self-learn the combination of motor movements so that the robot can move forward, left, right and backward using the most optimal combination for those movement.

For the feedback of the motion of robot we have use the MPU-6050 sensor in which the z-axis of gyro sensor is used for observing the left and right motion of robot whereas the y-axis of accelerometer is used for observing the forward and backward motion of the robot.

Two LEDs have also been attached to indicate if the action performed by the robot is correct according to the state or not. The green LED when switched on signifies that the action for current state is the right choice whereas the red LED when switched on signifies that the action for current state is not a right choice.

L293D is used as a motor driving IC to control the four 12 volt DC motor using the raspberry pi. As raspberry pi works on 5 volt power supply which can't be used to directly power the motor. So, in order to power the 12 volt motor, H-bridge is used.

As a learning process Q-Learning is used to train the robot. The pseudo code of the applied approach is as follows:-

1. Input the motor configuration (i.e. output pins of motors).
2. S (Set of states) is initialized with the states i.e. Move Forward, Turn Left, Turn right, Move Backward and Stop.
3. A (Set of actions) is initialized with all possible actions that can be performed.
4. γ (Discount rate) is initialized to 0.1.
5. α (Learning rate) is initialized to 0.1.
6. $Q[S, A]$ (Q-Table) is initialized with zeros.
7. Repeat <number of episodes> times:
    a. Initialize start state of the agent by randomly choosing the one of the states from S.
    b. Assign current state with the start state.
    c. Repeat indefinitely:
        i. Assign current action with any randomly chosen action from A.
        ii. Apply the chosen action on the current state.

iii. Assign reward R according to the action performed in the current state.
iv. Calculate $Q[s,a] \leftarrow Q[s,a] + \alpha(R + \gamma * max_a Q[s',a] - Q[s,a])$
v. Assign current state with next state
vi. If goal is reached (i.e. if the action corresponds to the state):
    - Break the parent loop

8. Find the action which fetches the maximum q-value with respect to each state i.e. (move forward, move backward, turn left, turn right or stop).
9. Store the corresponding actions to be used by the robot for motion.

The above give steps demonstrates the process of Q-Learning in which Gyro Sensor and accelerometer is used as a feedback to identify the correctness of the motion. The pseudo code of the reward function applied in Q-Learning algorithm is as follows:-

1. Perform the action which is currently assigned.
2. If accelerometer value in y-axis is above the positive threshold and the current state is forward:
    a. Switch ON the green LED
    b. Return +25
3. If accelerometer value in y-axis is below the negative threshold and the current state is backward:
    a. Switch ON the green LED
    b. Return +25
4. If gyro sensor value in z-axis is above the positive threshold and the current state is right:
    a. Switch ON the green LED
    b. Return +25
5. If gyro sensor value in z-axis is below the negative threshold and the current state is left:
    a. Switch ON the green LED
    b. Return +25
6. If none of the above condition is true switch ON the red LED and return -25 denoting incorrect action for the state.

The above demonstrated steps were repeated 1000 times in order to train the robot to follow the motion command of instructor using the process of self-learning.

4.1 Flow Charts

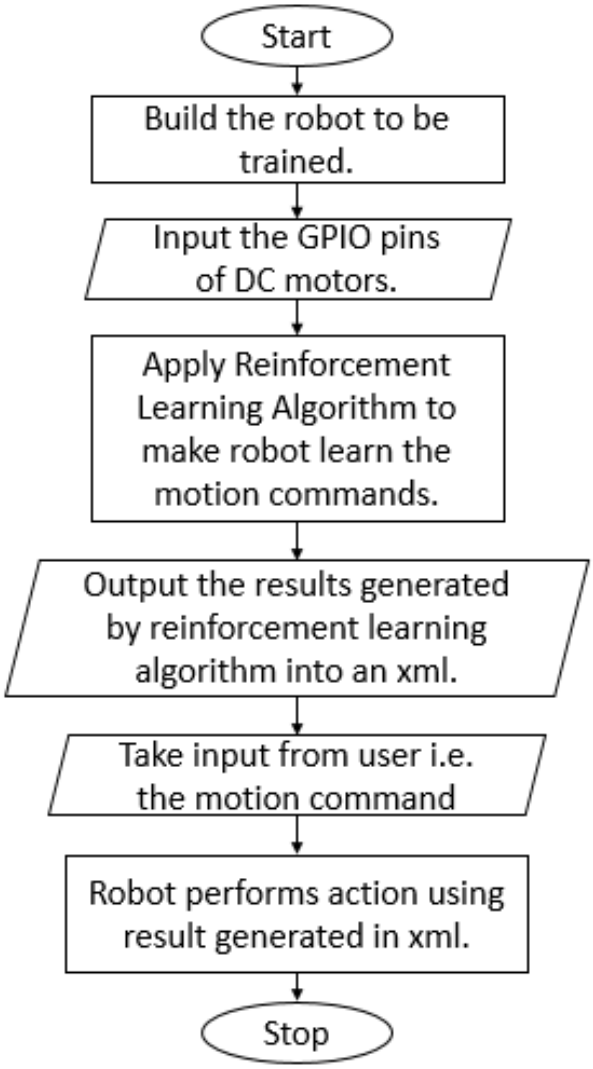


Figure 11: Overall flow chart of the project

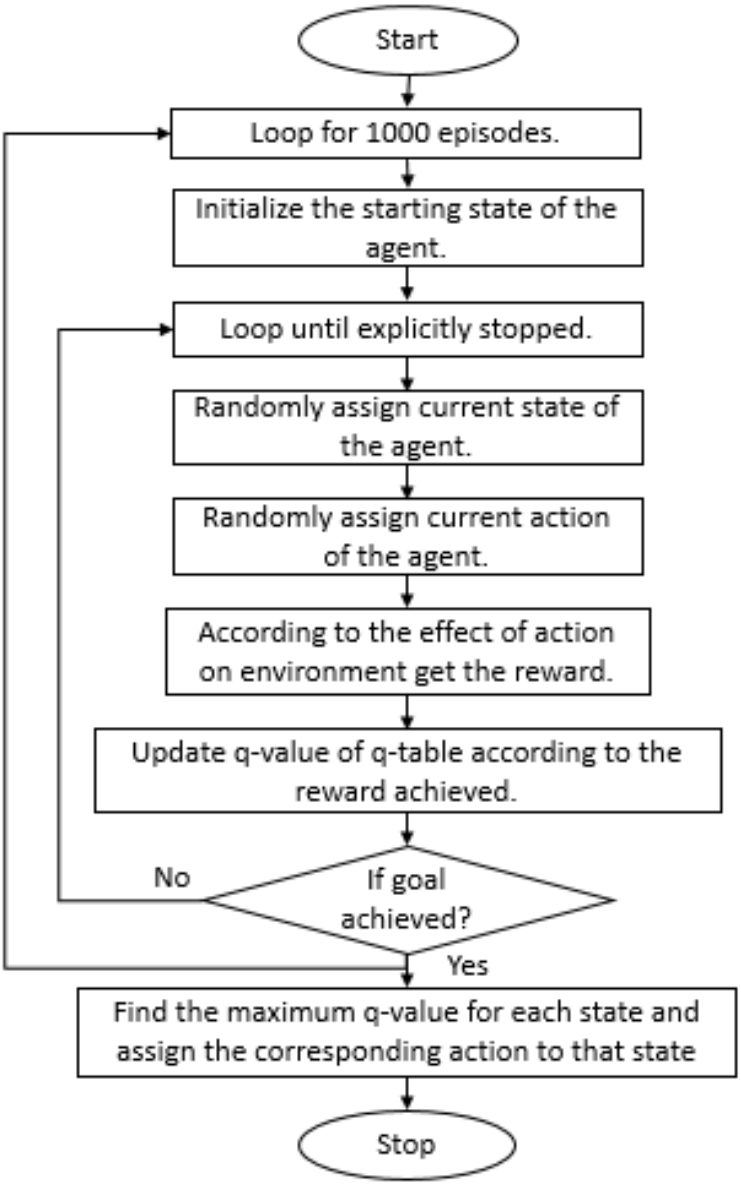


Figure 12: Flow chart of Q-Learning Algorithm

Chapter 5

Results

The implemented reinforcement learning algorithm was tested on the real world robot and the results achieved were satisfactory. In the below give graph it can be seen that approximately after 600 episodes of training the robot, the expected reward value becomes stable. This means that there were no change in the expected reward value and hence no change in the mapping of action with their respective states.

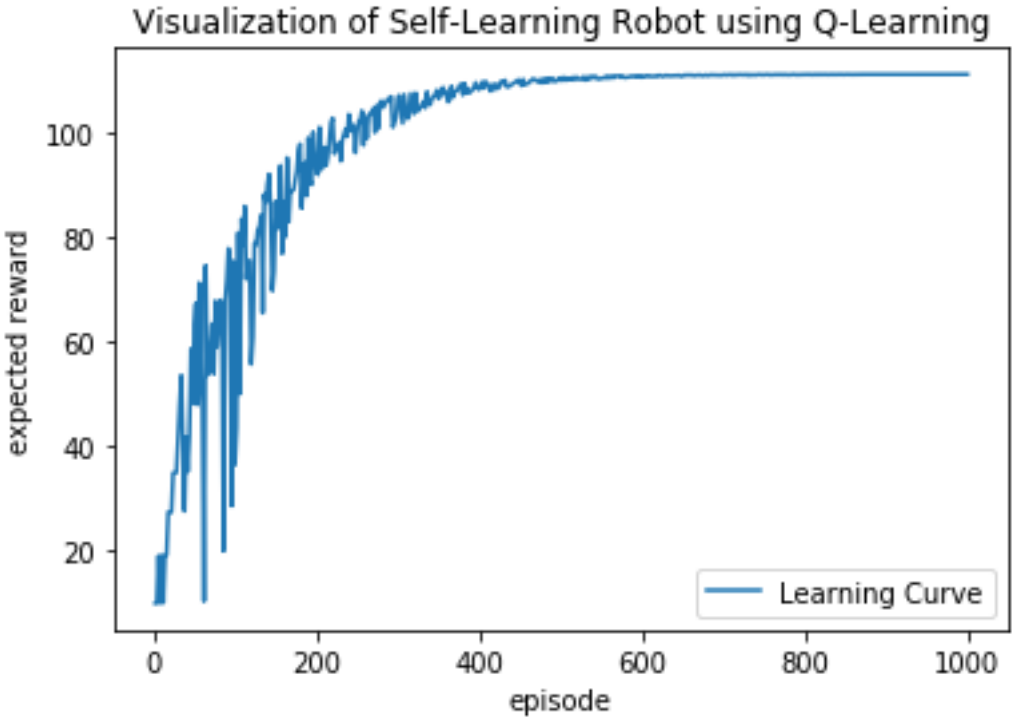


Figure 13: Visualization of Self-Learning robot using Q-Learning algorithm

The output of the Q-Learning algorithm is represented in below given fig 14. where each state has an action mapped against itself. The action comprise of four digit number where the first digit states the direction of front right motor, second digit states the direction of front left motor, third digit states the direction of back right motor and the forth digit states the direction of back left motor. The digit 0 signifies the ideal state of motor, digit 1 signifies the clockwise motion of motor and digit 2 signifies the anti-clockwise motion of the motor. The respective q-value is also shown in the image which is the highest q-value for that particular state.

```
Front : 1212 and its q-value is 111.074606507
Left : 1111 and its q-value is 111.092074959
Right : 2222 and its q-value is 111.102616426
Back : 2121 and its q-value is 111.103867041
```

Figure 14: Result of Q-Learning Algorithm

The output of the Q-Learning algorithm is even was dumped into an xml file so that it can be used later. This resultant xml file is used to control the robot according to the given command by the instructor.

```
<motorDrivingConfiguration>
        <Forward>
                <Pin>12</Pin>
                <Pin>6</Pin>
                <Pin>20</Pin>
                <Pin>19</Pin>
        </Forward>
        <Left>
                <Pin>12</Pin>
                <Pin>5</Pin>
                <Pin>20</Pin>
                <Pin>26</Pin>
        </Left>
        <Right>
                <Pin>16</Pin>
                <Pin>6</Pin>
                <Pin>21</Pin>
                <Pin>19</Pin>
        </Right>
        <Backward>
                <Pin>16</Pin>
                <Pin>5</Pin>
                <Pin>21</Pin>
                <Pin>26</Pin>
        </Backward>
</motorDrivingConfiguration>
```

Figure 15: Resultant XML file of Q-Learning algorithm

The resultant XML file contains the pin number which has to be enabled in order to do the parent node specific motion. For example to move the robot forward the pin number 12,6,20 and 19 should be enabled, to move backward the pin number 16,5,21 and 26 should be enabled, to turn left the pin number 12,5,20 and 26 should be enabled and to move right the pin number 16,6,21 and 19 should be enabled.

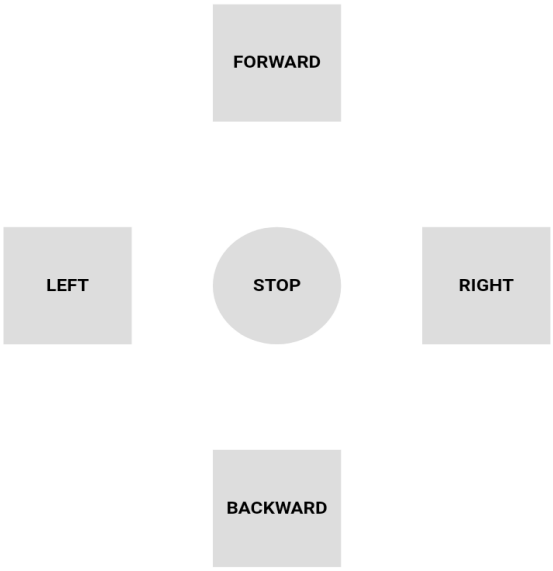


Figure 16: Remote of the robot

The above given image is of the remote that will be used by the instructor to give the commands to the robot such as forward movement, left turn, right turn and backward movement. This remote is based on the webpage, which can be accessed using any mobile phone which has web browser in it. This remote uses the same Q-Learning resultant XML which is given above in order to process the command.

Chapter 6

Conclusion

In this project, we have designed a robot which could self-learn to follow the motion commands given by the instructor. This was done using the concept of reinforcement learning. Reinforcement learning helps this robot to learn the motion commands using the trial and error strategy. The robot interacts with the environment and according to the action it takes the robot is awarded appropriately. The only drawback of this approach is that it takes huge amount of time in order to self-learn the motion commands.

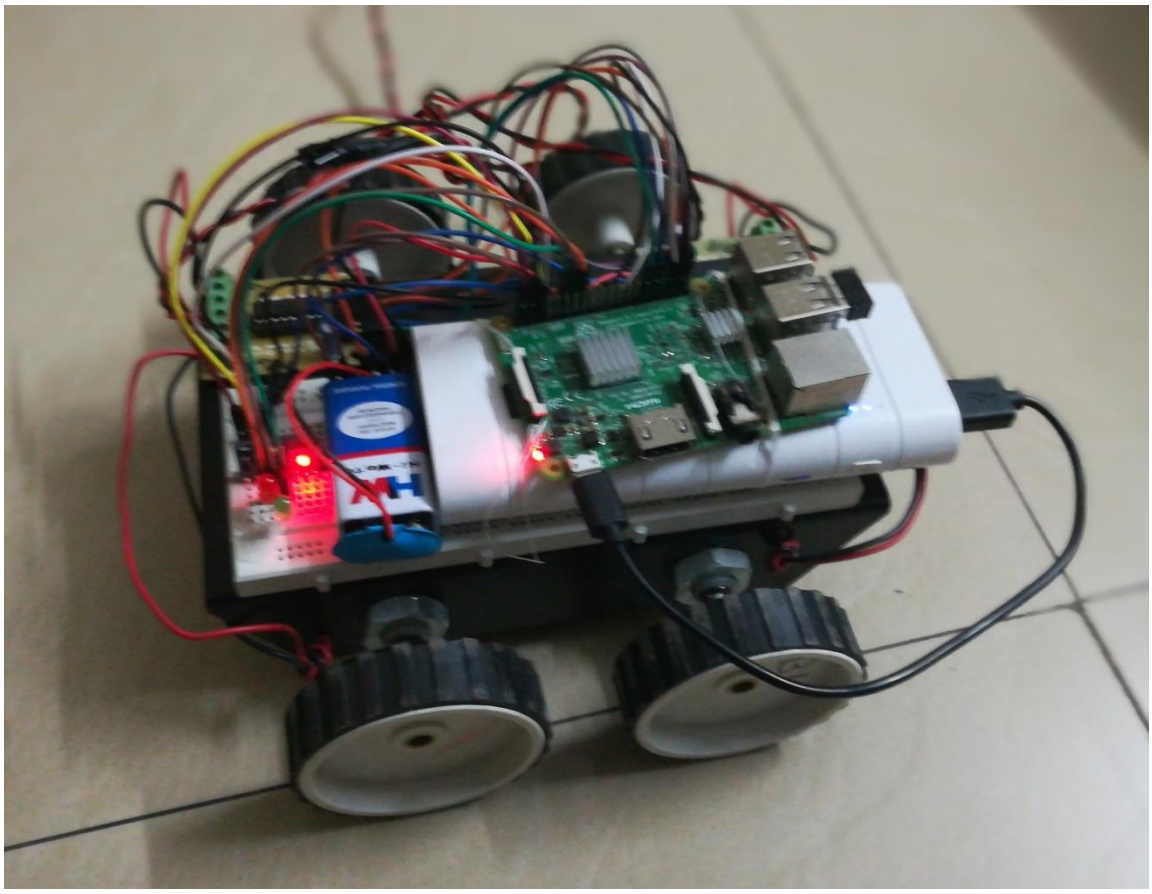


Figure 17: Self Learning Robot

The above image represents the self-learning robot which was built as a part of this project. Here we have used Raspberry Pi 3 Model B+ as a microprocessor to implement Q-Learning technique. We have implement this algorithm using Python as a main language and then the output generated from this algorithm is dumped into an XML file which is further used by the webpage to control the robot.

The Raspberry Pi 3 was powered by 10000 mAh power source and the four 12 volts DC motor used 12 volt external power supply for proper functioning of robot. This robot has the capability to recover from any type of malfunction in motors which could endanger the objective of the robot.

References

[1] Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. Vol. 1. No. 1. Cambridge: MIT press, 1998.

[2] Dayan, Peter, and Geoffrey E. Hinton. "Feudal reinforcement learning." *Advances in neural information processing systems*. 1993.

[3] Littman, Michael L. "Friend-or-foe Q-learning in general-sum games." *ICML*. Vol. 1. 2001.

[4] Sallans, Brian, and Geoffrey E. Hinton. "Reinforcement learning with factored states and actions." *Journal of Machine Learning Research* 5.Aug (2004): 1063-1088.

[5] Berná-Martínez, José Vicente, and Francisco Maciá Pérez. "Robotic control systems based on bioinspired multi-agent systems: application of the principles of neuroscience to robotics." (2011).

[6] Brock, Oliver, and Oussama Khatib. "Executing motion plans for robots with many degrees of freedom in dynamic environments." *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*. Vol. 1. IEEE, 1998.

[7] Patidar, Virendra, and Ritu Tiwari. "Survey of robotic arm and parameters." *Computer Communication and Informatics (ICCCI), 2016 International Conference on*. IEEE, 2016.

[8] Kröse, Ben JA, P. Patrick Van Der Smagt, and Frans CA Groen. "A one-eyed self learning robot manipulator." *Neural networks in robotics*. Springer, Boston, MA, 1993. 19-28.

[9] Sutton, Richard S. "Learning to predict by the methods of temporal differences." *Machine learning* 3.1 (1988): 9-44.