REINFORCEMENT LEARNING ON A ROBOT

N. Ansari[1], S. Jadhav[2], A. K. Gupta[3], V. Singh[4], S. Wani[5]
[1]Assistant Prof. Department of Electronics and Telecommunication
V.E.S Institute of Technology
[2,3,4,5] Department. of Electronics and Telecommunication
V.E.S Institute of Technology

Abstract

The purpose of this project is to use the concepts of reinforcement learning and digital image processing on robots to teach them to walk. A crawler robot and a quadruped robot are used to apply machine learning algorithms. These robots use their acquired knowledge in training process to make model of themselves and perform specified task.

Index Terms: Reinforcement Learning, Digital Image Processing, Robotics, Quadruped robot, Locomotion.

I. INTRODUCTION

Robots are used practically in every domain, these have unique capabilities, from imitating living organisms to performing assigned task. Industrial automated robots majorly perform repetitive task assigned in a static environment and are unable to achieve robust performance in unpredicted conditions without human intervention [1].These robots are constructed as per their equivalent mathematical model and then used to perform task assigned under manual guidance [2]. It is difficult to mathematically model the robot under every condition, such as change in environment, new task and faulty part. Considerable advancements in method and researches have been done in order to make the robot model its environment autonomously [1]. Without internal models, robotic systems can autonomously synthesize increasingly complex behaviors [3-6] or recover from damage [7] through physical trial and error, but this requires hundreds or thousands of tests on the physical machine and is generally too slow. In order to generate an inference of its own morphology,” the robot need to perform autonomous modelling. A machine is able to indirectly infer to its own morphology through self-directed exploration and then use the resulting self-modelling synthesize new behaviors” [8].

Here we describe a method to make a spider like quadruped robot learn to walk without human intervention. The process involves use of reinforcement learning (RL) with recurrent neural network and computer vision in order to achieve the desired goal. In the initial stage of learning process the robot assumes itself as a black box and performs random actions in order to explore its morphology through self-modelling [8]. In the later stages it starts to use the knowledge it has acquired of its morphology to walk in practical unpredicted environment.

II. PROBLEM STATEMENT

Over the years we have seen robots and machines evolving and also replacing humans in many fields, but whenever it comes to dealing with a new set of problem or to build a machine for new task we need to start from the grass root level, also machine is designed to perform a particular task under a lot of environmental restrictions i.e. the machine will not be able to work or will not provide with satisfactory result in alien conditions, which make all the machines restricted to their domain. These restrictions are majorly due to explicit programming. In order to make our machines and robots robust in nature we need to come up with a solution which satisfies most of our needs. Explicit programming reduces adaptability of the robot, in order to provide maximum adaptability and robustness to the robot, it should be able to learn and adapt from its surrounding. First step towards building an environment adaptive robot is building a robot who has complete knowledge of its own morphological structure and can perform a particular task without being explicitly programmed to do so. We aim to build a quadruped (spider) & crawler robot which will learn to walk in a given region independent of human intervention or control.

III. RELATED WORK

RL is found to be difficult at multiple stages of is application and when dealing with robots this list grows even longer few of the problem are discussed in Reinforcement learning in robotics a survey [9]. 1. Curse of dimensionality, robotic systems have to deal with continuous states and actions, in order to handle this complexity we lower the functionality or using computational abstraction.2. Curse of real world samples, robots interacts with real world so this faces real world issues, which includes expensive robot hardware, wear - tear and maintenance, also the dynamics of robot may change due to external factors ranging from temperature to wear. 3. curse of real world interactions, this is due to strict constraints on interaction between algorithm and robot setup, physical delays in sensing and actuation, communication delays. 4. Curse of model errors, unfortunately building an accurate model and environment is challenging. But small models accumulate error. 5. Curse of goal specification, since goal of task is controlled by reward defining a good and efficient reward function in robot RL is daunting task. inverse RL also known as apprenticeship learning is a good alternative [9].In another research it has been found that a self-model driven algorithm is having high probability of inferring a topologically correct model than by random baseline algorithms [8].

IV. LITERATURE SURVEY

A. Modeling

There are seven different dimensions on which models can differ as discussed [10]: 1.Relevance 2.level 3.Generality 4.Abstraction 5.Structural Accuracy 6.Performance match 7.Medium. Biological modelling should be done in terms of their behavior towards real life task and requirement [10]. Modelling is used to simulate in order to understand the behavior of the animal [10]. Template are used to tackle the complexity of model, template uses redundancies and symmetry of animal to simplify the model [11]. For quadruped robot template is of an inverted pendulum [12].

B. Artificial Neural Network

Artificial neural network (ANN or NN) are nonlinear signal processing networks, which are built by interconnecting multiple artificial neurons, these neurons act as the building block of the NN. NN are inspired by biological neural network. It consist of multiple neurons working simultaneously to perform a particular task. NN learns by example (training). It is parallel distributed processor which stores experiences and make use of it for prediction of new inputs. The artificial neuron consist of multiple parts namely weight, activation function, sigmoidal function, bias and threshold. Learning process of the NN involves modifying weights in the network layers aiming to achieve expected output, learning process is classified of three types: 1. Supervised learning 2. Unsupervised learning3. Reinforcement learning [13-15].

1) Supervised Learning

Process of teaching the NN by providing with sample input and comparing the error with the expected output. Back propagation, pattern associated memory net are few example of supervised learning.

2) Unsupervised Learning

In an NN if input training vectors are known but target output is not known. The net modifies such that similar input vector is assigned same output unit. Clustering algorithm is an example of unsupervised network.

3) Reinforcement Learning (RL)

It’s a type of NN where if input training vectors are known but target output is not given but instead an indication of whether the output answer is right or wrong. Neural network learn the input output mapping through trial and error for maximizing performance index called reward, also known as reinforcement signal [13-15]. RL has 5 major component namely environment, agent, policy, reward & model. Environment is the physical surroundings of the robot or system. Agent receives observations from environment & mathematical simulator and return action to them. There are multiple algorithms for agent like Q-learning, SARSA (State -Action -Reward -State -Action), DQN (Deep Q - Network), DDPG (Deep Deterministic Policy Gradient) & Actor-Critic. Each of them have their own ability and used under different circumstances [16]. Policy is mapping from perceived states of environment to actions to be taken when in those states. Reward signal is sent to the RL agent from environment, it defines the goal in RL. Agent's role is to maximize total reward in long term. It defines good and bad decisions for agent [16]. Model (Mathematical Simulator) of environment replicates the environment and makes inferences about how the environment will behave under similar actions, this also helps in planning and deciding the action for possible future situations.

C. Digital Image Processing

Image Processing or Digital Image Processing (DIP) is use of algorithms to enhance and manipulate digital image to achieve particular goal. Images are of two types 1. Gray scale images, these are 2-Dimensional array of data whose pixel value ranges from 0 - 255 for 8 bit point 2. Colored images, these just like gray scale images but its pixel values are sub-divided into three channels R-G-B (Red-Green-Blue) making it a 3-Dimensional array, each data point in an image is referred as a pixel in DIP , manipulation of these pixel values result in DIP. In practice there are multiple concepts and algorithm to perform image processing, the concept related to the project are image thresholding, masking and finding contours. Image thresholding is a process of converting a given image into binary image, binary image is an image in which all the pixel values are either 0 or 1(i.e. 0 & 255). Thresholding is done in order to achieve a region of interest. For a gray scale image thresholding is given by the equation 1, where δ is the threshold value and $0 \leq \delta \leq 255$.For colored image three threshold are required i.e. one for each channel, it is given by the equation 2.

$$g(x,y)\begin{cases}1, & f(x,y) \geq \delta \\ 0, & f(x,y) < \delta\end{cases}$$

Where,

$g(x,y)$ is resultant binary pixel value.

$f(x,y)$ is original pixel value.

$$g(x,y) = \begin{cases}(R,G,B) \geq (\alpha,\beta,\gamma), then\ f(x,y) = (1,1,1) \\ (R,G,B) < (\alpha,\beta,\gamma), then\ f(x,y) = (0,0,0)\end{cases}$$

Where,

α : Threshold for Red channel

β : Threshold for Green channel

γ : Threshold for Blue channel

Masking is done to get the ROI (Region of Interest) form the image i.e. making the background of image as zero or some other constant value. The equation 2 gives the mathematical formulation of mask. Different methods are available to perform masking, for this application we use binary image & original image and do binary operation on these two images, which is given by the equation

$$h(x,y) = f(x,y) \land g(x,y)$$

Where,

$h(x,y)$: Resultant Image

$f(x,y)$: Original Image

$g(x,y)$: Binary Image

Contours are mathematically defined as function of curve in a plane of two or three variables i.e. $f(x,y,z)$ or $f(x,y)$ [17-18] in DIP domain it is defined as a continuous curve joining continuous points having same pixel intensity value [19]. This enables us to identify the ROI i.e. circle, and find out its properties like area, mean, intensity etc. for further calculation.

D. Digital Video Processing

Videos are sequence of images are presented one after the other with a very small delay which make us human perceive discontinuous images as a continuous moving picture. In video each image is called a frame. Digital video processing involves breaking the video in frames then performing DIP algorithms on each frame and recombining frames into one frame. Widely used in media and robotic automation [20].

E. Velostat

Velostat is a fabric made of a polymeric foil (polyolefins) and is capable of conducting electricity, widely used for protecting electronic gadgets from electrostatic discharge generated from surroundings. Recognizable feature of velostat sheet is its ability to vary its resistance under pressure or flexing, which makes it an inexpensive sensor in market for experiments and use in day to day life, as compared to its alternatives present in market. The sensor works in a sandwich structure between two conductive plates, shown in Fig 1.

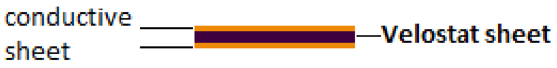

Fig 1 : Sandwich structure for Velostat

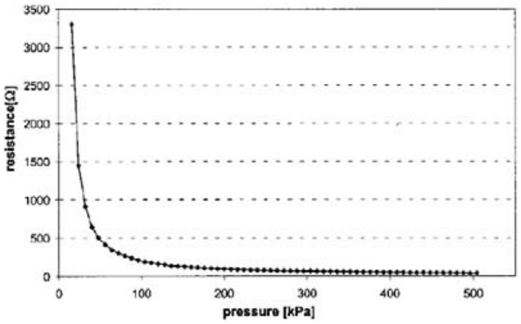

Fig 2 : Resistance curve of velostat[21]

It has several advantages in respect to other commercial realizations: it is of very simple design and made with low-cost materials easily found on the market; it is a resistance variation sensor, therefore it can be supplied with a direct current which allows the utilization for static applications. The measuring range and the calibration curve have been carried out applying known pressures to velostat sheet, its response is nonlinear at initial stages and shows hyperbolic behavior till 500 pka after this the sensor becomes asymptotic in nature and reaches its scaling capability, post reaching the asymptotic nature the there is no more variation in resistance indicating saturation [21], as shown in Fig-2.

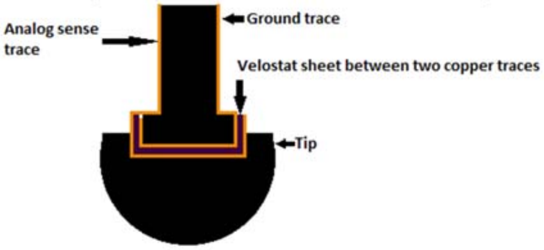

Fig 3: Tip setup for spider

V. DESIGN

A. Crawler

Spring-Loaded Inverted Pendulum (SLIP) concept proposes use of spring like model to replicate behavior of segmented locomotive organisms [11]. As the name suggest crawler robots crawl over the ground to move from one point to another. Our model of crawler is divided into three segments, namely the body, middle and tip segment, these segments are held together by two joints which are revaluating in nature and have vertical axis of revolution.

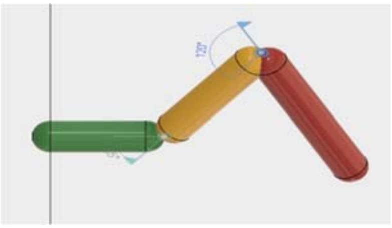

Fig 4: Crawler model

The Body - Middle segment joint is called joint A whereas the joint between Middle - Leg segment is called joint B. The joints A & B provide the robot with 2 degrees of freedom. Even though theoretically these joints have no limits i.e. angular displacement can be anywhere between 0° & 360° but practically limitations has to be applied in order to make a more realistic model and also since the servo motors are capable to rotate between 0° to 180°, so at any given instance the angular displacement of a servo motor θ always remains in the range of [$\delta_{\min}$, δ_{max}] where δ_{min} is the minimum allowable angular displacement and δ_{max} is max allowable angle of displacement. Fig 4 shows our model of crawler, where the green segment is body, yellow segment is middle segment and red segment is tip segment. Position of body is fixed but the joint-A and joint-B are revolute joints. Hardware design is as per the equivalent model discussed. Block diagram for crawler robot is given in Fig 5 where L-A and L-B stand for motors at joint A & B, T represent the tip of crawler which has velostat sensor.

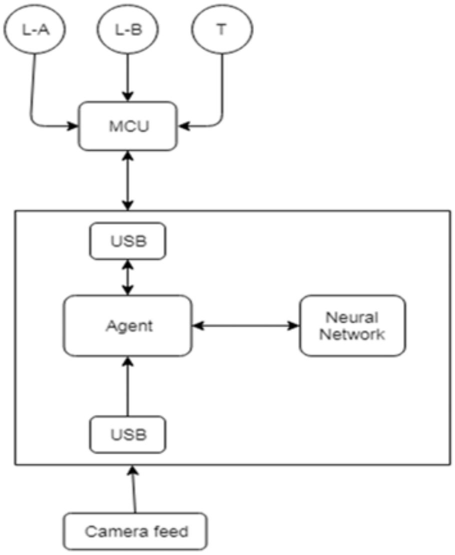

Fig 5: Crawler Block diagram

B. Spider

As mentioned in SLIP concept the locomotive robots are best modelled by SLIP modelling technique. The quadruped robot is a spider like robot. This robot consist of four legs and a body, each leg is subdivided into 3 segments namely lower-segment, middle-segment and tip-segment. Each leg is just like a crawler and has 2 joints i.e. between lower-middle segment and between middle-tip segment also each leg is connected to the main body via another revaluating joint which is body-lower segment joint. The 2 x 4 = 8 leg joints have vertical axis of rotation whereas the 1x4 = 4 body joints has horizontal axis of rotation. The robot has total 12 degrees of freedom associated to it, which are 4 + 8 = 12. Just like the limitation of each joint in crawler the joints of spider robot also face the limiting angular rotation in practical.

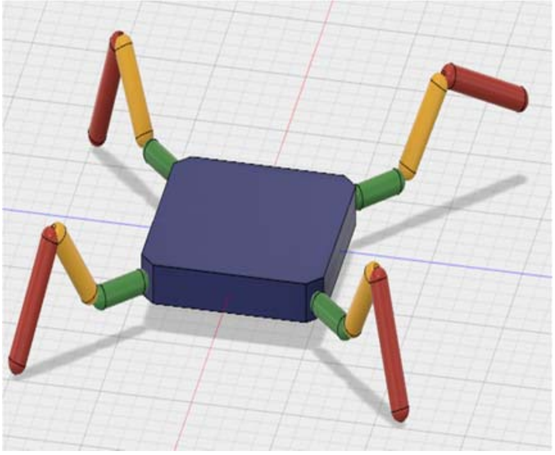

Fig 6: Spider Model

In the above model we can see that the blue is the main body, green is lower-segment, yellow is middle-segment and red is tip-segment. Hardware design is as per the equivalent model discussed above, block diagram for spider robot is given in Fig 7 where L1-A, L1-B and L1-C represent motors at joint A, B & C, T1 represent the tip of robot leg 1 which has velostat sensor and this arrangement is made for each leg. Hardware model of spider is made on a CAD modeling software and then printed using 3D printing technique.

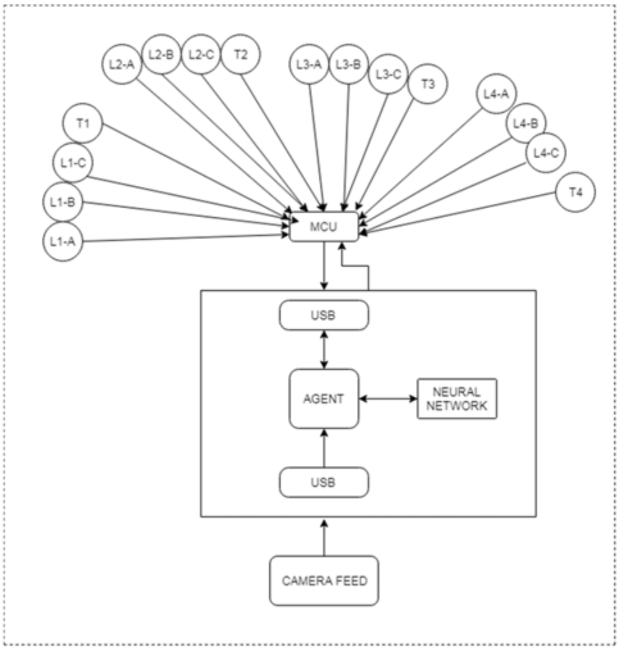

Fig 7: Spider block diagram

C. Digital Image Processing

DIP in our project is used for reward generation in response to action taken by robot in physical environment. Reward received is proportional to the distance travelled by the robot. Position of robot is tracked by a detection of particular template shown in Fig 8.

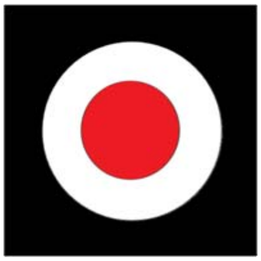

Fig 8: Template

The video frame captured by the camera is processed using openCV python, flow of template detection is given in Fig 9. Resultant image is obtained by application of multiple DIP procedures at different stages, image at each stage is given in the Fig 10-12, in the Fig 10.D we can observe that template detection is not done very accurately since the “red circle” is not detected completely. We are using color thresholding which is dependent on pixel values of image, these pixel values change with surroundings and lighting conditions [21] that’s why we cannot observe a complete circle in image Fig 10.D, for this episode we used upper pixel limits as (130, 120, 250) & lower pixel limit as (0, 0, 130), the image format uses BGR nomenclature.

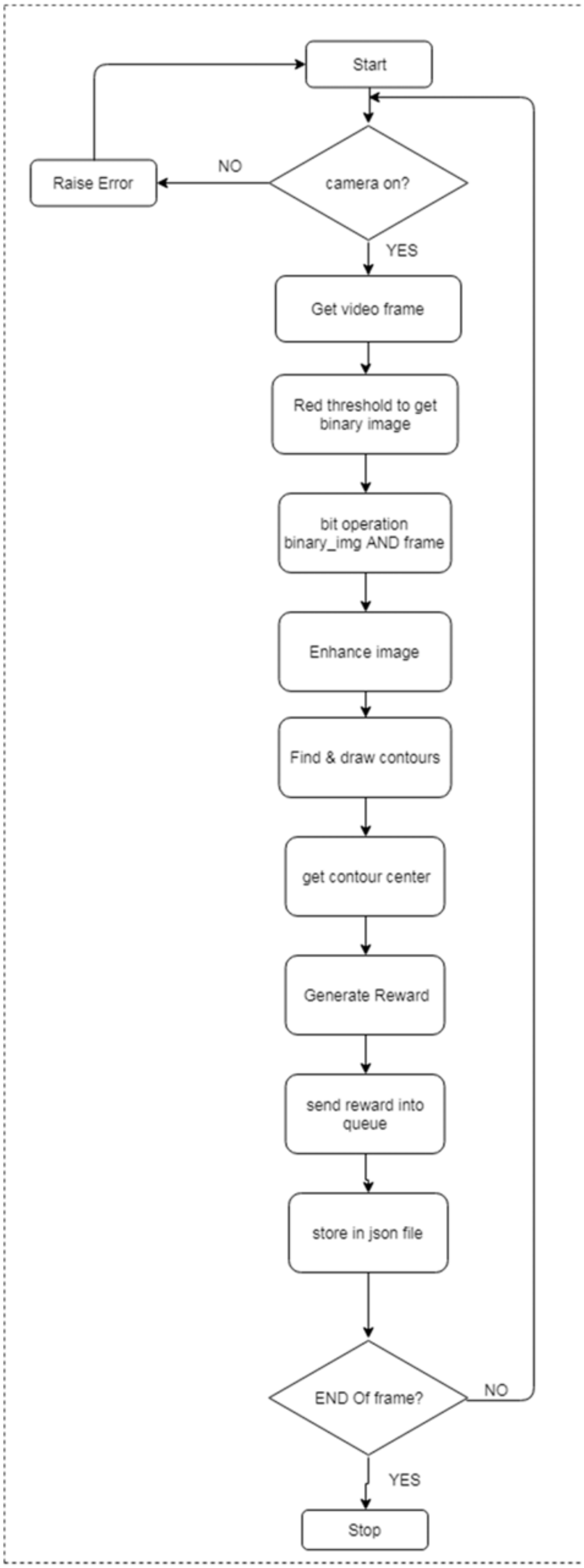

Fig 9:

Even though complete template is not displayed correctly we can observe in out Fig 10.D, that the ROI is drawn in Blue i.e. contour area. In next Fig 11.D we can observe that even though the template is detected accurately some noise in image is present due to surrounding pixels, for this episode we used upper pixel limits as (125, 145, 255) & lower pixel limit as (0, 0, 13) . Now we look at the Fig 12. Here we have perfect template detection and no noise from the surrounding, the threshold value we used in this image is same as in Fig 11. , but provides better results, this is due to the fact that DIP is hugely dependent on its surrounding and illumination.

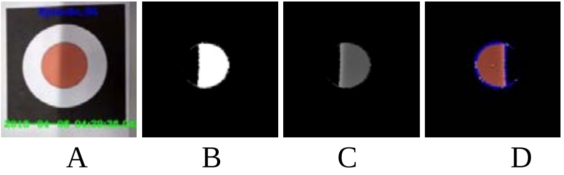

Fig 10: Imperfect Detection of template.

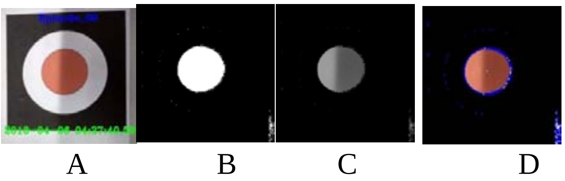

Fig 11: Perfect Template detection with noise

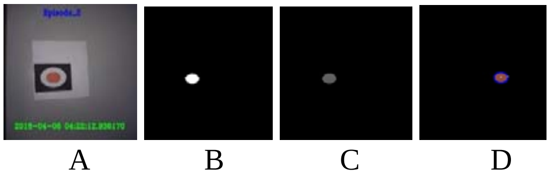

Fig 12: Perfect template detection without noise

VI. RESULT

A. Crawler

Crawler robot has 2 DOF (Degree Of Freedom) and simpler in complexity compared to the spider, the crawler robot successfully learned to walk in environment, through self-evaluation using RL concept. At initial stage the response of environment (ENV) varies from that of input, but as the NN learns the error decreases. Two big circle in the image show position of L-A, L-B motors and smallest circle show the tip T of the robot. Fig 13.A shows initial stage of the simulator when error is very high and accuracy is low, Fig 13.B shows intermediate stage when NN (simulator) SIM almost replicates the ENV.

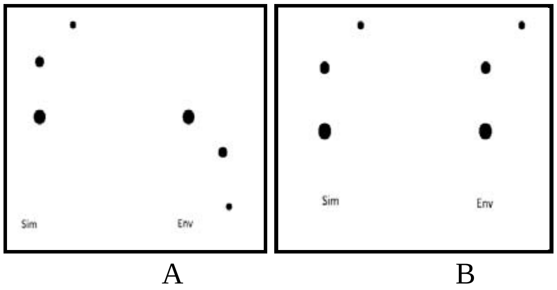

Fig 13: Software simulation of crawler

The error and accuracy values attained by the NN describe the efficiency of network and the system in real time. Our neural network has reduced the error value to ≈ 0. Accuracy value of our neural network is not yet steady and requires more training in order to achieve accuracy of ≥ 0.85. The crawler robot design is capable of walking in an environment without human intervention by using continuous self-modeling and evaluation.

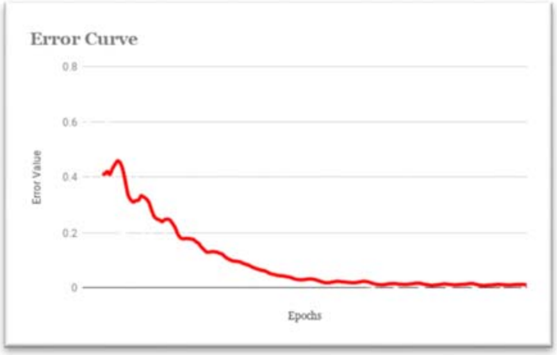

Fig 14: Error curve

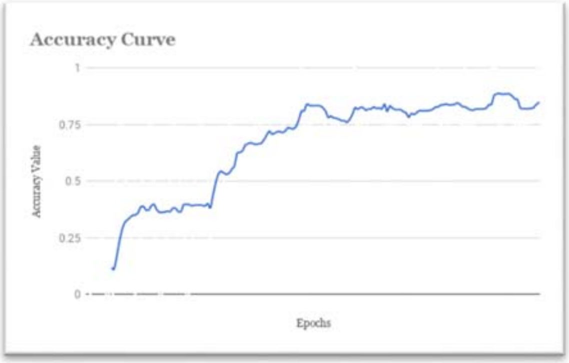

Fig 15: Accuracy Curve

B. Spider

Spider robot is highly complex and involves 12 DOF. The neural network is big and requires long learning cycles to achieve desired goal. Our NN is replicating the actions given to it in ROS-environment (Robotic Operating System) but still need to undergo many epochs to attain a high accuracy, low error state and independent working ability.

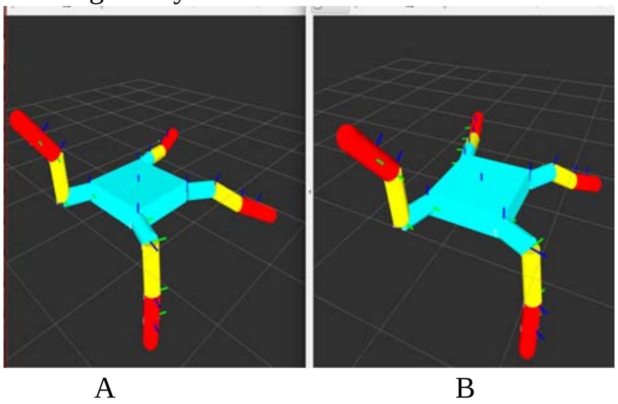

Fig 16: ROS model for A) ENV and B) SIM

VII. CONCLUSION AND FUTURE SCOPE

The use of RL based algorithms on robots can make the teaching process more independent of human intervention and enable robots to perform under unpredicted stages. The Crawler is smaller in complexity in comparison to spider robot, spider robot would require to consider multiple parameters like change in orientation of body and power consumed in single action to achieve ideal modeling of itself.

REFERENCES

[1] Thrun S, Burgard W, Fox D. Probabilistic robotics. MIT press; 2005 Aug 19.

[2] Sciavicco L, Siciliano B. Modelling and control of robot manipulators. Springer Science & Business Media; 2012 Dec 6.

[3] Nolfi S, Floreano D. Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines. MIT press; 2000.

[4] Verschure PF, Voegtlin T, Douglas RJ. Environmentally mediated synergy between perception and behaviour in mobile robots. Nature. 2003 Oct;425(6958):620.

[5] Hornby GS, Takamura S, Yamamoto T, Fujita M. Autonomous evolution of dynamic gaits with two quadruped robots. IEEE Transactions on Robotics. 2005 Jun;21(3):402-10.

[6] Pfeifer R, Scheier C. Understanding intelligence. MIT press; 2001 Jul 27.

[7] (17) Mahdavi SH, Bentley PJ. Innately adaptive robotics through embodied evolution. Autonomous Robots. 2006 Mar 1;20(2):149-63.

[8] Bongard J, Zykov V, Lipson H. Resilient machines through continuous self-modeling. Science. 2006 Nov 17;314(5802):1118-21.

[9] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: A survey. Journal of artificial intelligence research. 1996;4:237-85.

[10] Webb B. Can robots make good models of biological behaviour?. Behavioral and brain sciences. 2001 Dec;24(6):1033-50.

[11] Full RJ, Koditschek DE. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. Journal of experimental biology. 1999 Dec 1;202(23):3325-32.

[12] Chen JJ, Peattie AM, Autumn K, Full RJ. Differential leg function in a sprawled-posture quadrupedal trotter. Journal of Experimental Biology. 2006 Jan 15;209(2):249-59.

[13] Rajashekaran S, Vijayalksmi GA. Neural networks, fuzzy logic and genetic algorithms. Prentice-Hall of India Pvt. Ltd; 2004 Aug 1.

[14] Haykin S. Neural networks: a comprehensive foundation, 1999. Mc Millan, New Jersey. 2010.

[15] Fausett L. Fundamentals of neural networks: architectures, algorithms, and applications. Prentice-Hall, Inc.; 1994 Jul 1.

[16] Sutton RS, Barto AG. Reinforcement learning: An introduction. Cambridge: MIT press; 1998 Mar 1.

[17] Hughes-Hallet D, Gleason AM, McCallum WG, Flath DE, Lock PF, Gordon SP, Lomen DO, Lovelock D, Mumford D, Osgood BG, Pasquale A. Calculus, single and multivariable.

[18] Courant R, Robbins H, Stewart I. What is Mathematics?: an elementary approach to ideas and methods. Oxford University Press, USA; 1996.

[19] Contour Approximation Method [Internet]. OpenCV: Contours :Getting Started. [cited 2018Apr22]. Available from: https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html

[20] Gonzalez RC, Woods RE. Image processing. Digital image processing. 2007;2.

[21] Del Prete Z, Monteleone L, Steindler R. A novel pressure array sensor based on contact resistance variation: Metrological properties. Review of Scientific Instruments. 2001 Feb;72(2):1548-53.