Modeling and Simulation of Hexapod Kinematics with Central Pattern Generator

Aju M.T.*, Napolean A.**
* P.G. Scholar, Department of Electrical and Electronics Engineering, V.S.B. Engineering College Karur, India
** Associate Professor, Department of Electrical and Electronics Engineering,
V.S.B. Engineering College Karur, India

Article Info

Article history:

Received Feb 9, 2016
Revised Apr 25, 2016
Accepted May 12, 2016

Keyword:

Artificial Neural Network
Central Pattern Generator
Legged Robot
Limit Cycle Oscillator

ABSTRACT

The revealed secrets of nature always led humans to their aspiring achievements. The fastest animal on land is Cheetah and similar robot has developed by engineers so far to attain a record speed of 20mph among legged robots. But in nature there are some insects those are far ahead of cheetah in speed with a unit of body length per second. Insects are small in their body size with legs usually countable from 4 to 12 or more. With more legs they can have more stability and can adapt to different terrain faster while walking. Six legged robot (hexapod) is generally expect to attain higher speed in terms of body length per second, since the nature has proof for it. Bio-inspired Central Pattern Generator (CPG) is in use for so far in robotic world to mimic the locomotion patterns of insects and other animals. Currently the hybrid controller of CPG and reflex is going on and this paper suggests a new architecture for the system. Neural Network modeled CPG acts as the motor neuron for each joint of the leg. In each instant a neural network models the gait of the robot by learning procedure from the reflex system. This is like the Central Nervous System (CNS) selecting gait of an animal according to the terrain that travels. CNS takes sensory feedback from eyes, force on each leg and body balance from cochlea to adapt the gait for current terrain. This paper in first place tries to simulate the gait patterns for a hexapod.

Corresponding Author:

Aju M.T.,
Departement of Electrical and Electronics Engineering,
V.S.B. Engineering College Karur,
Karudayampalayam P.O., Karur 639 111, Tamilnadu, India.
Email: ajusumithran@gmail.com

1. INTRODUCTION

The revealed secrets of nature always led humans to their aspiring achievements. The fastest animal on land is Cheetah and similar robot has developed by engineers so far to attain a record speed of 20mph among legged robots. But in nature there are some insects those are far ahead of cheetah in speed with a unit of body length per second. Insects are small in their body size with legs usually countable from 4 to 12 or more. With more legs they can have more stability and can adapt to different terrain faster while walking. Six legged robot (hexapod) is generally expect to attain higher speed in terms of body length per second, since the nature has proof for it. Bio-inspired Central Pattern Generator (CPG) is in use for so far in robotic world to mimic the locomotion patterns of insects and other animals. Currently the hybrid controller of CPG and reflex is going on and this paper suggests a new architecture for the system. Neural Network modeled CPG acts as the motor neuron for each joint of the leg. In each instant a neural network models the gait of the robot by learning procedure from the reflex system. This is like the Central Nervous System (CNS) selecting gait of an animal according to the terrain that travels. CNS takes sensory feedback from eyes, force on each leg and body balance from cochlea to adapt the gait for current terrain. This paper in first place tries to simulate the gait patterns for a hexapod.

2. EXISTING SYSTEM

Explaining research chronological, including research design, research procedure (in the form of algorithms, Pseudocode or other), how to test and data acquisition [1]-[3]. The description of the course of research should be supported references, so the explanation can be accepted scientifically [2], [4].

2.1. Legged Robot Configurations

A lot of examples can be found of the 6 legged robot designs; the most referred is the Rhex on the internet and journals. This robot has a very simple but effective design to walk over very rough terrain and can even walk up and down stairs. The problem with the specific design of Rhex is that it has is not capable to turn the robot, it can only go "straight" ahead. In a TV show called “Prototype This!” an attempt is made to make a full scale all terrain robot with 6 legs using the Rhex leg design. They failed to do it in 2 weeks’ time, but show the problems by scaling this concept. Amplifiers, batteries and controllers cannot handle the motion of the 450 Kg vehicle. Another simple and good design discussed here uses only 14 RC-servo motors in total. A similar design as the webx is the so called eggshell robot. This robot also uses simple RC-servo motors and shows some very promising walking behavior. A different design approach where the legs of the robot are placed complete around the robot (circular body). This has the advantage of being more flexible and in moving and achieving the same walking speeds in all directions which is needed for the RoboCup robot. This Hexateuthis can turn around his axis at a high velocity[3].

2.2. Leg Trajectory Planning

The walking algorithm determines the robot’s movement, the load of the robot engine, the dynamics of power consumption, torque of the separate reductors, etc. The way of moving the leg is there for very important. There is a lot of literature the leg trajectory algorithms; they vary from cyclic genetic algorithms to relative simple cycloid functions. The easiest way for a leg trajectory generation is that from the Rhex robot with the "half" wheel design. The leg can spin quickly circular in the transfer phase to get back in the support phase [1].

For the general 3DOF leg a different approach is needed. The pathway of the arms is prescribed in order to a movement. First the leg inverse kinematic model should be designed where the joint angles are set as a function of the leg tip. Assuming that the robot walks in the tripod gait, every leg makes the exact same path, only one group of legs is half a period delayed in time. One group is standing on the ground and pushes the legs backwards, and the others are in lifted-up position and move forward (assuming the robot walks forward) . This basic approach can then later be optimized for faster, stable and flexible walking, which then also influence the efficiency of the power consumption [8].

The leg trajectories are also studied from animals. By linking the biology and technical application . Walking insects and most six-legged robots requires simultaneous control of up to 18 joints. A simple model of a stick insect leg consists of four functional segments: the Coxa, the femur, the tibia and the foot. When the joint of the foot and the foot itself are neglected for simplicity, a stick insect leg can be modeled as a manipulator with three hinge joints resulting in 3DOF, which commonly the same as a hexapod robots leg.

For the optimization of the leg trajectories different techniques are found. The FIR filter approach explained in. A finite impulse response (FIR) filter is a type of a discrete-time filter. With the use of this filter the motion gets smoother and therefore also more efficient. More literature for optimization of the leg trajectory is found with the use of genetic algorithms. A quick and converging algorithm suitable for any-time learning. Cyclic genetic algorithms are developed with four variations, with one variation that produces an optimal tripod gait which is robust enough to adapt to significant changes in the capabilities of the robot model. The optimization can also be achieved by soft computing techniques. Further studies are based on point to point trajectory planning and the and how to make adapt trajectories so that the hexapod follows a certain line. Even when joint failure happens and a joint is locked the robot can still walk. Or adapting the center of gravity position of the robot with respect to the feet placement in order to stabilize the hexapod, and of course optimized algorithms for turning [4].

2.3. Robot Control

In the last years there has been a growing interest in the area of legged robots. Especially in the gait generation mechanism and control of this mechanism. One of the aspects related with the control of legged robots that has received more attention is that of statically stable gaits. The task of such a gait generation mechanism can be defined as appropriate sequence of leg and body movements so that the robot moves in the desired speed and direction to reach his target. The control of the robot is there for directly connected to the leg trajectory planning therefor a lot of literature is found for both. In the literature of legged robots there are many different approaches to implement controllers for wave gait generation. A gait is defined as a sequence of leg motions coordinate with a sequence of body motions for moving the overall body of the robot in the desired direction and orientation from one place to another. Some examples found in the literature are for instance the periodic gaits, free gaits or a combination of both approaches. A gait is called periodic when similar states of the same leg during successive strokes occur at the same interval for all legs. The free gait or also called non-periodic gait is when each leg can move on a free chosen interval and own stroke algorithm.

Periodic gaits are suitable for smooth terrain and they have been studied by several investigators, some which are worth mentioning are: Song and Waldron, Zang and Song. The free gaits on the other hand, are much more effective on rough terrain with obstacles. A lot of research is done on free gaits to find graphical and analytical methods. The difficulty with these methods is the complexity and interactions. Therefor genetic algorithms (GAs) are mostly used for free gait implementation. GA are population-based search and optimization techniques which work on Darwin's principle of natural selection. GA has shown to be a powerful tool for global optimization and has been used as key elements in many learning techniques. Another way then using GA is a fuzzy logic controller (FLC), a potential tool for handing imprecision and uncertainty [4].

The robots leg also has a separate control for the dynamics of foot-ground interaction. Here an algorithm is adopted with foot-force feedback control for the robots locomotion. The division of the leg's pathway can be optimized by FIR filtering so that the leg pathway gets a more efficient trajectory generation. Also using the techniques of soft computing will improve the leg speed, this is done by on-line trajectory generation, see chapter 7. Most of the studies focused on the control at the leg level and leg coordination using neural networks, fuzzy logic, and hybrid force/position control and sub-assumption architecture. The control at the joint level is almost always implemented using a PID scheme. A semi-autonomous six-legged walking machine with hexagonal architecture has been developed for research on gait control is described. The machine weighs 13 kg; each leg has three DOF with closed-loop kinematics. The kinematics is designed to achieve gravitational decoupling, which simplifies considerably the velocity control of the legs. The control architecture is decentralized, each leg being controlled by a separate micro controller[6].

3. PROPOSED SYSTEM

The hexapod robotic system presented in this paper is developed bearing in mind several abilities such as, moving on irregular paths, overcoming obstacles, climbing and going down stairs, without compromising its stability and keeping a low overall weigh.

After several iterations, a geometric model was achieved, as it is shown in the figure 1. The mechanical structure of the hexapod robot consists of one rigid, load carrying mainframe with six legs, similar and symmetrically distributed. Each leg is composed by three links, interconnected by two revolute joints and attached to the main body by means of a third revolute joint. Revolute motors and linear actuators accomplish traction movement and elevation, respectively. The foot of each leg is rigidly attached. The main dimensions of the model are length ≈ 8.35 inches, width ≈ 3.54 inches and height ≈ 5.15 inches. This model is used to generate elementary locomotion behavior. For an N rigid body system the mobility or the number of degrees of freedom (DOF) is given by Gruebler& Kutzbach Equations.

3.1. Mechanical Model

This chapter describes the mathematical model of the hexapod platforms used in this thesis. This includes a full kinematic description as well as a definition of the local frames of the components and their relationship amongst each other.

3.2. Body Model

This chapter describes the mathematical model of the hexapod platforms used in this thesis. This includes a full kinematic description as well as a definition of the local frames of the components and their relationship amongst each other.

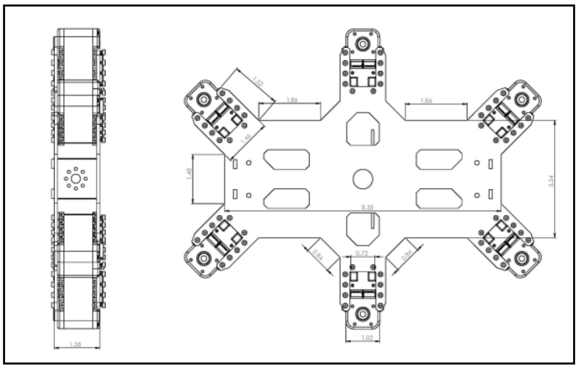

Figure 1. Hexapod Body Plan

The hexapod consists of six legs connected to a rigid body via Coxa joints. The total body width is W_a. W_a is the distance between the middle-left and middle-right Coxa joints. The width W_b describes the distance between the front-left and front-right (or rear-left and rear-right) Coxa joint. The total body length is L_b, defined by the distance between the front-left and rear-left (front-right and rear-right) Coxa joints. The 3-dimensional Cartesian body frame B lies in the center of the body with its x-axis pointing forward, its y-axis pointing to the left and its z-axis pointing upwards (right-hand rule).

3.3. Leg Model

A leg 2 consists of three links the Coxa link L_c , the femur link L_f and the tibia link L_t with an end effector et. L_c is connected to the body via a coxa joint j_c, the femur link is connected to the Coxa link via a femur joint j_f and the tibia link is connected to the femur link via a tibia joint j_t. Each joint has a local reference frame F_j, with the z-axis being the joint's rotation axis and the x and y axis forming a right-hand-system together with the z-axis. The local reference frame of the Coxa joint serves also as the leg frame F_l of the corresponding leg. Since the Coxa joints for the front legs are rotated by 45-degree (front-right leg) and 45-degree (front-right leg), their local frame F_L is rotated relative to B accordingly. The same applies for the rear legs (135-degree for the rear-left and 135-degree for the rear-right leg).

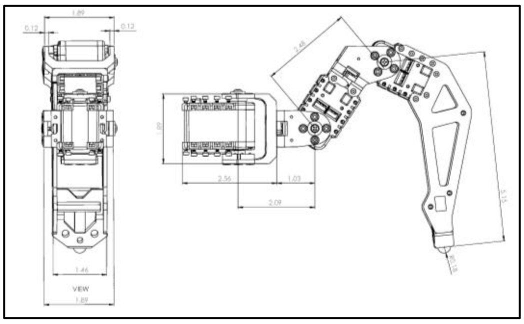

Figure 2. Robot single Leg Configuration

3.4. Modeling of Hexapod in Solidworks

The hexapod is first modeled in Solidworks and then exported to SimMechanics using SimMechanics Link. Some models of different parts of the hexapod were found online from the GrabCAD website. The individual parts of the hexapod are assembled to different solid bodies. To improve simulation time it is important to keep the number of solid bodies low. A CAD model of one leg can be seen in Figure 2 The leg modeled using three solid bodies Coxa, femur and tibia . The main body of the hexapod is modeled as one solid body and is showed in Figure 1. The different parts of the hexapod can be connected via joints in either SimMechanics or in Solidworks. To get more control of how coordinate frames are assigned, the coordinate frames are created in Solidworks. When they have been exported to SimMechanics the frames are connected by joint blocks. Different reference frames used at different joints to connect the different leg parts in SimMechanics using a revolute joint. The six legs are then connected to the main body with the use of revolute joints. This model can be seen in Figure 3.

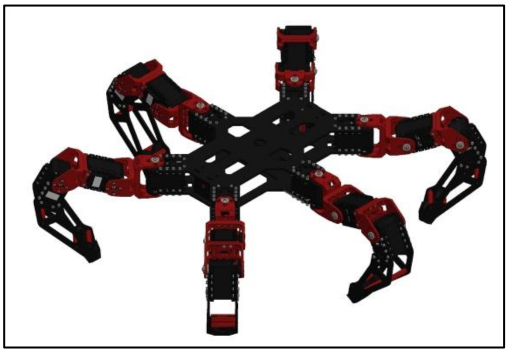

Figure 3. Hexapod Model - Isometric view

The physical modeling in the SimMechanics environment considerably facilitates simulation efforts of complex mechanical systems regardless of their complication by elastic and damping elements and by number of degrees of freedoms. The SimMechanics program scheme having the form of interconnected blocks shows how the physical components with geometric and kinematic relationships of the robot are mutually interconnected. The SimMechanics program enables one to model mechanical systems by bodies and joints, to simulate their motion, to change easily the structure, to optimize system parameters, and to analyze results all within the Simulink environment. This approach does not require cumbersome deriving differential equations of the system and presents an easy and fast way to obtain the dynamic model of the system and saves time and effort.

Mathworks collaboration with Solidworks Corporation extended the engineering analysis capabilities of SimMechanics by allowing seamless integration of Solidworks CAD Assemblies into the SimMechanics simulation and design environment. This means that the Solidworks models can be simulated in the Simulink environment in order to analyze forces and torques in mechanical joints, plot accelerations and displacements of each part of the system, to visualize motion of the CAD assembly, while taking into consideration masses of individual objects. This facility is enabled by installing an appropriate plug-in in SimMechanics which imports the 3D CAD model of the full system with bodies, joints, couplings, and masses from the Solidworks program into the SimMechanics for further work with the model.

3.5. Leg Inverse Kinematics

The goal of the inverse kinematics is to position the foot of the leg at point p1(x1;y1; z1). To achieve this three servo angles need to be calculated. Using the coordinate system and notation presented in Figure 4, the angle for the Coxa servo is calculated by (2) were arctan2 is defined according to (3). When g = 0 the servo is at position 150°. For positive g servo angle is increased, this corresponds to clockwise rotation in Figure 5.

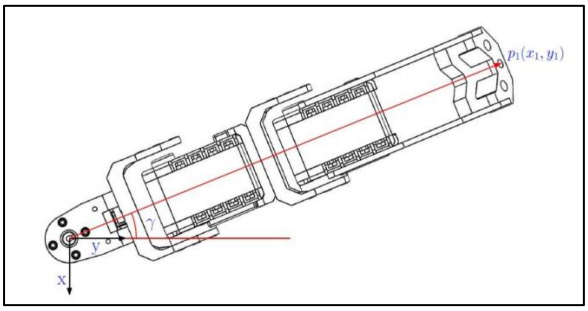

Figure 4. Notation used for calculation the Coxaangle

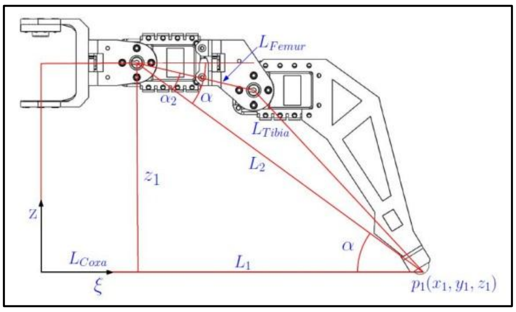

Figure 5. Notation used for calculation the femur

(8) and tibia angles (9)

$$\gamma = -atan2(x_1; y_1) \tag{1}$$

$$\gamma_{coxa} = 150^0 + \gamma \tag{2}$$

$$atan2(y,x) = \begin{cases} arctan\frac{y}{x}, & x > 0 \\ arctan\frac{y}{x} + 180^0, & y \geq 0, x < 0 \\ arctan\frac{y}{x} - 180^0, & y < 0, x < 0 \\ 90^0, & y > 0, x = 0 \\ -90^0, & y < 0, x = 0 \\ undefined, & y = 0, x = 0 \end{cases} \tag{3}$$

To calculate the angles for femur and tibia servo, notation according to Figure 4 is used. The ξ axis points along the vector from origo to p1 in Figure 5.

First L1 and L2 are calculated according to (4) and (5). Using the law of cosines the femur and tibia angle can be calculated according to (8) and (9). In Figure 5 femur and tibia are at an angle of 150°. Increasing γ_{Femur} and γ_{Tibia} makes point p1 move in positive z direction. To calculate offset$_{Femur}$ and offset$_{Tibia}$ the servos is set to 150°. Then coordinates of p1 is found to be $p1 = (0; 209.4, 111.4)$ mm using the CAD model.

$$L_1 = \sqrt{x_1^2 + y_1^2} - L_{Coxa} \tag{4}$$

$$L_2 = \sqrt{L_1^2 + z_1^2} \tag{5}$$

$$\alpha = atan2(z_1, L_1) \tag{6}$$

$$\alpha_2 = arccos\left(\frac{L_{Femur}^2 + L_2^2 - L_{Tibia}^2}{2 \cdot L_{Femur} \cdot L_2}\right) \tag{7}$$

$$\gamma_{Femur} = 150 - (\alpha_2 - \alpha + offset_{Femur}) \tag{8}$$

$$\gamma_{Tibia} = 150 - arccos\left(\frac{L_{Femur}^2 + L_{Tibia}^2 - L_2^2}{2 \cdot L_{Tibia} \cdot L_{Femur}}\right) + offset_{Tibia} \tag{9}$$

3.6. Modeling Of Contact Forces in Simmechanics

To make the SimMechanics model able to interact with the environment, contact force is modeled. Blocks or methods for performing this do not exist in SimMechanics. As a first approach an add-in library for contact forces is used [11]. This library can model contact forces between 2D objects. Contact between two bodies is modeled as a spring-damper system. The most important contact force is considered to be contact between the floor and the hexapod. Because of the limitations of this library, only contact between the six feet and the floor is modeled.

3.7. PID Control

This Most often dynamical systems are controlled by control laws designed according to the methods of classical and modern control. These methods are applicable if a relative precise model of the dynamic system being controlled is available. Planned jointed trajectories constitute the reference for the robot control system. Therefor a inverse dynamic model can be formulated. The trajectory planning is held in the Cartesian space, but the control is performed in the joint space, which requires the integration of the inverse kinematic model in the forward path. The control algorithm considers an external position and velocity feedback and an internal feedback loop with information of the foot ground interaction force.

3.8. Walking using CPG and reflexes

As a model of CPG, we used a neural oscillator proposed by Matsuoka. A neural oscillator (NO) consists of two mutually inhibiting neurons . Each neuron in this model is represented by the nonlinear differential equations. By connecting NO of each leg, the NOs are mutually entrained and oscillate in the same period and with a flexed phase difference. This mutual entrainment between the NOs of legs results in a gait. We used a trot gait, where the diagonal legs are paired and move together, and two legs supporting phase are repeated. Although we realized dynamic walking on at terrain in trot and pace gaits using a CPG alone, sometimes walking became unstable even on at terrain because the supporting legs slipped. This meant that it was difficult to realize stable walking using a CPG alone since CPG could not directly deal with the interaction of legs with a floor. Therefore, for dynamic walking on irregular terrain with a low degree of irregularity, we employed a control system involving a CPG and stretch and flexor reflexes, and obtained the following results from this experiments.

Dynamic walking became much more stable, compared with the use of a CPG alone, because stretch reflex torque assisted in the phase transition from swinging to supporting and in preventing the supporting leg from slipping. Walking on terrain furnished with obstructions to swinging legs was made possible by the flexor reflex.

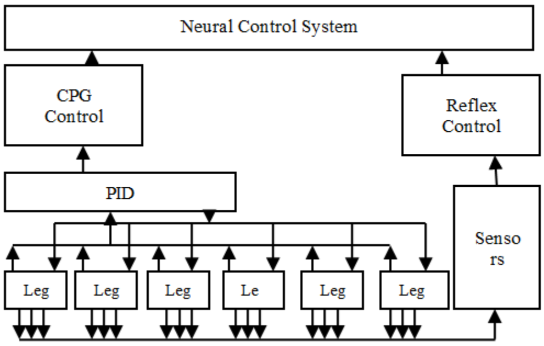

Figure 6. Proposed architecture

3.9. Adaptive control mechanism

When we consider walking as an exchange of supporting legs, the stability of walking is nothing but the reliability of the exchange of supporting legs. Therefore, in the case of walking on irregular terrain, it is essential that A leg not be prevented from moving forward in the former period of the swinging phase, a leg be landed reliably on the ground in the latter period of the swinging leg phase, and The angular velocity of the supporting legs around the contact points at landing moments be kept constant in spite of changes in the height of the ground surface. For (1) and (2) to be satisfied, we have already employed the flexor reflex and the stretch reflex, respectively. In order to avoid collision of a leg with an obstacle as far as possible, recognition of the environment by vision and prior adjustment of swinging leg motion is needed. Condition (3) was commonly used in the control of dynamic walking and running in inverted pendulum model-based control. In order to satisfy condition (3), a larger torque at the hip joints of the supporting legs is required when going up and a smaller torque is required when going down. The adjustment of torque by reflexes based on vestibular and somatic sensation as feedback control is effective to some extent. But, when a change of height in a step is large, prior adjustment of supporting leg torque by vision is needed as feed forward control.

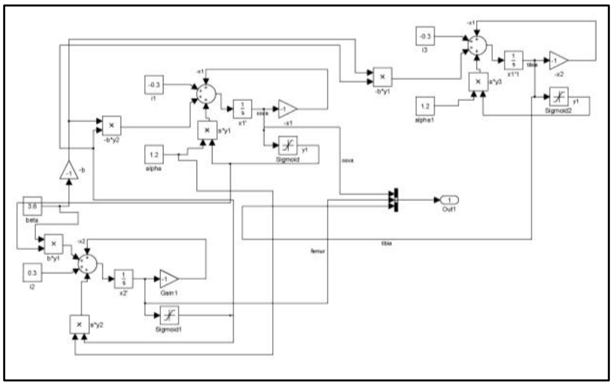

Figure 7. Neural Oscillator

The Figure 7 shows the neural oscillator used in the simulation. The simulation only shows the kinematics of the hexapod. Each leg is connected with one of this neural oscillator through a gain amplifier. This single neural oscillator gives drive signals for all the three servo motors.

4. RESULTS AND OUTPUTS

Solidworks model of the hexapod exported into a Simulink model using the SimMechanics link plugin in SolidWorks. The model is then rearranged and new subsystem levels are introduced to encapsulate similar blocks in SimMechanics. A Central Pattern Generator is then design using control system toolbox in reference to the Matsuoka neural oscillator. This oscillator just drives the legs and gives a kinematic simulation of the legs.

Leg numbering as specified previously defines different gait patterns are then simulated using different triggering sequence of each legs oscillator. In this paper it only shows the kinematic simulation of the hexapod using the neural oscillator. A machine model output of the simulation in Simulink is shown below. Each solid part has its on reference frame and it is marked in them. The centroid of each body and reference frame are the data available from the SolidWorks model so that it is further easy to develop the dynamic model from this project, and it will be dealt in next phase of the project.

Each Leg is driven with three signals for Coxa, Femur and Tibia the Figure 8 shows the signal dynamics for a time period. Figure 9 only shows the signals for the single leg drive signal and the feedback signals are also same because the robot shows only kinematic behavior of the robot.

Neural oscillator is provided for each leg and the triggering time difference of each legs neuron makes the different gait patterns. The Figure 9 shows the difference in oscillations of two different legs in which one contracts and the other extends at an instant. This forms a particular gait and we can make the robot behave as its own with the environment by using the feedback The final result from the MATLAB shows the machine model simulation of the robot. The machine model simulation output window is shown in Figure 10. Simulation shows a particular gait at a time for the neural oscillator parameters. In the next phase the dynamics of the robot will be implemented along from this kinematic model.

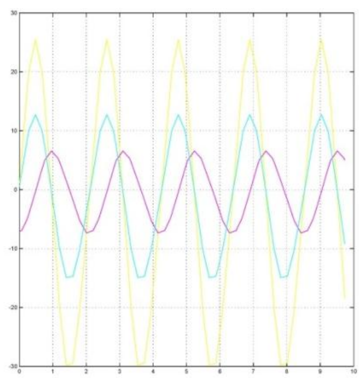

Figure 8. Leg Signals for Coxa(Yellow), Femur(Blue) and Tibia(Pink) servos

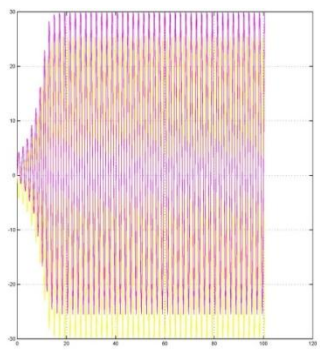

Figure 9 Coxa NO for the extensor and contractor Leg servos

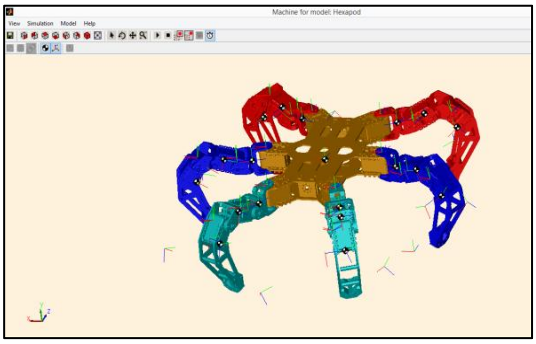

Figure 10. Simulation model of the hexapod

REFERENCES

[1] Filip Stefanovic (2015), "Efferent feedback in a spinal-like controller Reaching with perturbations". Transactions on Neural Systems and Rehabilitation Engineering.

[2] Ferreira, C., Figueiredo, J., & Santos, C. P. (2015). "Quadruped Locomotion Based on Central Pattern Generators and Reflexes." 2015 IEEE International Conference on Autonomous Robot Systems and Competitions.

[3] Mostafa Ajallooeian, Sebastien Gay, Alexandre Tuleu, Alexander Sprowitz and Auke J. Ijspeert (2013). "Modular Control of Limit Cycle Locomotion over Unperceived Rough Terrain". 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) November 3-7, 2013. Tokyo, Japan.

[4] Giles D. Wood and Dallas C. Kennedy. "Simulating Mechanical Systems in Simulink with SimMechanics". The MathWorks, Inc.

[5] Haitao, Y., Wei, G., Jing, D., Mantian, L., & Hegao, C. (2013). "A CPG-based Locomotion Control Architecture for Hexapod Robot". 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) .

[6] Katiegrace Youngsma (2012). "Development of a Model and Simulation Framework for a Modular Robotic Leg".

[7] Leghmizi Said and Boumediene Latifa (2011). "Modeling and Control of Mechanical Systems in Simulink of Matlab. Applications of MATLAB in Science and Engineering", InTech China.

[8] John Schmitt, Raffaele Ghigliazza, Justin Seipel (2009). "Neuromechanical models of legged locomotion:How cockroaches run fast and stably without thinking about it". Neuro-Mechanics, Dynamics and Decision-Making, Technion, Feb 15-16, 2009.

[9] Auke Jan Ijspeert (2008)."Central pattern generators for locomotion control in animals and robots: A review". 21.2008 Neural Networks.

[10] Arthur d. Kuo(2002), "The relative roles of feedforward and feedback in the control of rhythmic movements", Motor Control, 2002, 6, 129-145

[11] Igor Aizenberg (2001), "advances in neural networks", Tempus Project