STUDY AND CONTROL OF BIPED ROBOT USING INTELLIGENT CONTROL

End Term Project Report

Submitted By –

SERIAL NO. – 6

Members:

Aayushi Shrivastava 15115002

Under the supervision of –

Prof. Manish M. Joglekar Department of Mechanical Engineering, IIT Roorkee	Prof. M. Felix Orlando Department of Electrical Engineering, IIT Roorkee

In End Term Report Submission

For the course of EEN 400A: B. Tech Project

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY

ROORKEE

2018

PROJECT VERIFICATION

The following Project Report is documented for the Electrical Engineering Course of EEN 400A: B. Tech Project for the students of B. Tech, IV Year.

The report is based on the Topic: ______________________________

____

"

I have proofread and hereby verify that the following Project has been completed and done under my supervision and strict guidance.

I approve that all data, information & documents attached in the Report is under my knowledge and every single member of the group has contributed his share to the making of either of the B.Tech Project / Model / Report.

"

Signature of Supervising Faculty In-charge: _______________________

Name of Faculty In-charge: _______________________

____

Signature of Supervising Faculty In-charge: _______________________

Name of Faculty In-charge: _______________________

____

Index

Acknowledgement

I wish to express my sincere gratitude to **Prof. M. Felix Orlando** and **Prof. Manish M. Joglekar** for providing me the opportunity to work under their guidance. I am also thankful for their continuous guidance and support during the course of project.

Aayushi Shrivastava 15115002

4th Year Electrical Engineering

IIT Roorkee

Abstract

A humanoid robot is a robot that has a human-like shape. Some of the basic benefits of humanoid robots can be summarized as follows;

1. Humanoid robots can work in the environment for humans as it is,

2. Humanoid robots can use tools for humans as it is, and

3. Humanoid robots has a human-like shape which makes it friendlier to work with humans.

The essential difference between humanoids and other kinds of robots (like industrial ones) is that the movement of the robot has to be human-like, using legged locomotion, especially biped gait. The ideal planning for humanoid movements during normal walking should result in minimum energy consumption, as it does in the human body. For this reason, studies on dynamics and control of these kinds of structures has become increasingly important.

The question of walking biped robot stabilization on the surface is of great importance. Maintenance of the robot’s gravity centre over the centre of bearing area for providing a stable position can be chosen as a goal of control. To maintain dynamic balance during the walk, a robot needs information about contact force and its current and desired motion.

Objectives

The main objectives of the project are as follows:

- To calculate kinematics and dynamics parameters of a 4 DOF bipedal robot as a function of time.
- To model a State Space Model of a Bipedal robot for applying the control law.
- To design a required Controller for the robot using Control Law Partitioning.
- To simulate and model the robot along with controller in MATLAB.

Theory

As one of the objective of our project being study of basic kinematics and dynamics of biped robot, we have become familiar with the basic principles of these like zero pole moment (ZMP) and dynamics and inverse pendulum analysis of biped walking. The ZMP (Zero Moment Point) proposed by ***Vukobratovi´cet al***. is a criterion to judge if the contact between the sole and the ground can be kept without solving the corresponding equations of motions. The contact is kept if the ZMP is an internal point on the sole. When the robot does not move, the contact is kept when the projection of the centre of the mass of the robot onto the ground is an internal point of the sole. In the case of a single point of contact of each leg the ZMP lies at the same point for the complete motion and thus it is easier to model and control such a biped.

In “static walking”, the projection of the centre of mass never leaves the support polygon during the walking. In “dynamic walking”, there exist time periods when the projection of the centre of mass leaves the support polygon. Most toy robots perform static walking using large feet. Nevertheless, human feet are too small with respect to the height of centre of mass to perform static walking. Indeed we are performing dynamic walking in our daily life.

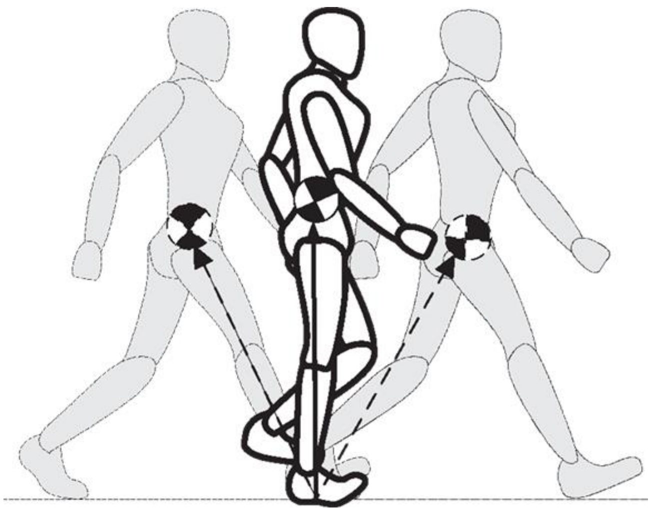

Fig 1: Inverted Pendulum Modelling for Dynamic Walking

For finding the trajectory of COM and walking pattern we model the COM as an inverted pendulum. First, we assume that all the mass of the robot is concentrated at its centre of mass (COM). Second, we assume that the robot has massless legs, whose tips contact the ground at single rotating joints. At last, we only consider the forward/backward and the up/down motions of the robot, neglecting lateral motion.

Methodology

1. <u>Modelling of a Stance Leg</u>

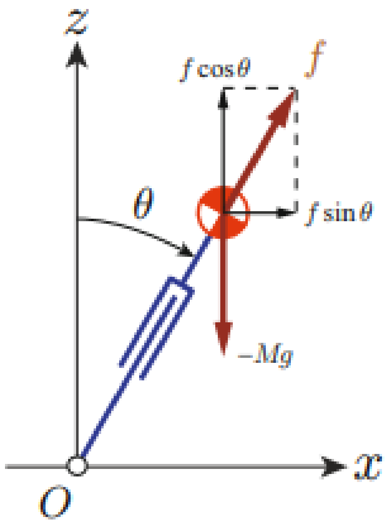

Fig 2: Free body diagram of an inverted Pendulum

By investigating Fig. 2 again, we see the horizontal component of the kick force f remains while the vertical component is cancelled by gravity. The horizontal component accelerates the COM horizontally, thus we have

$$M * \ddot{x} = f * \sin\,\theta$$

By solving this above differential equation for the horizontal dynamics we obtain equation of motion of the COM as follows:

Equation of Motion for Linear Inverted Pendulum

$$x(t) = x(0)*\cosh(t/T_c) + T_c*\dot{x}(0)*\sinh(t/T_c)$$

$$\dot{x}(t) = x(0)*\sinh(t/T_c)/T_c + \dot{x}(0)*\cosh(t/T_c)$$

$$\ddot{x}(t) = g/z*x(t)$$

$$\text{Stride\_time} = T_c *\log((x(0) - T_c * \dot{x}(0))/(x(0)-Tc*\dot{x}(0)))$$

$$T_c = \sqrt{z/g}$$

Minimum Velocity of Walking

For simulation of Biped robots, first we need to find the minimum velocity such that the pendulum has have enough kinetic energy so that it can overcome gravity and COM moves in a horizontal straight line.

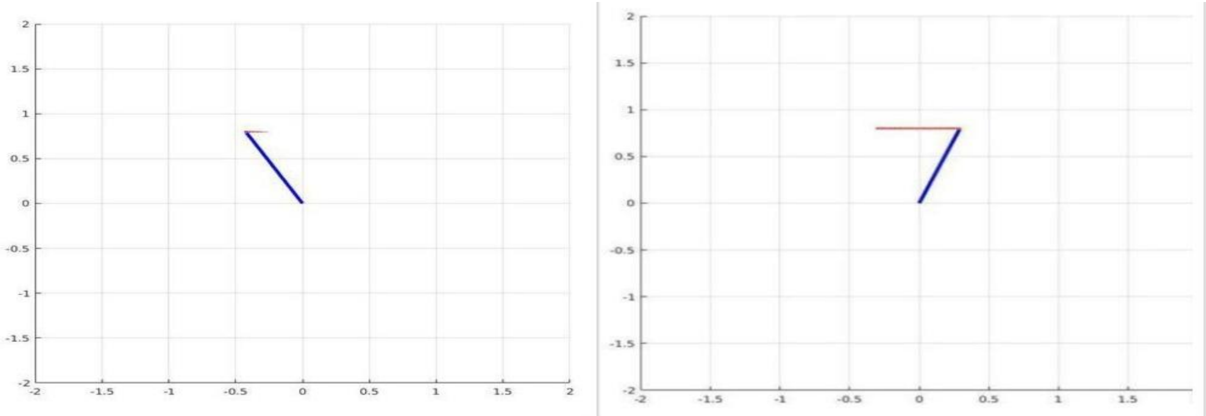

Fig 5- pendulum when (a) Velocity is less than required (b) Velocity is more than required

As the trajectory of COM is found using inverse pendulum analysis and the important parameters like stride length and stride time is calculated from the above equations. For the stance leg the pivot point is the contact point at the ground and trajectory of COM is used to find the joint angles of hip and knee using inverse kinematics.

All the parameters such as Angular velocity and Angular acceleration of each robot joint of stance can be calculated by Jacobian of the stance leg. Following equations were calculated and used in simulation of our project.

Inverse Kinematics of 2 link planar leg

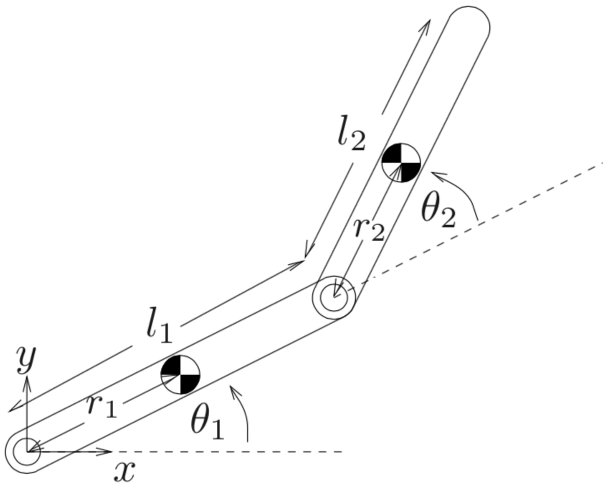

Fig 4- Leg equivalent to 2 link Planar Manipulator

Actuator Kinematics

$$num = (x^2 + y^2 - (l_1)^2 - (l_2)^2);$$
$$den = 2 * l_1 * l_2;$$
$$\theta_2 = acos(num / den);$$
$$sin\theta_1 = ((l_2 + l_1 * cos(\theta_2)) * y - l_1 * sin(\theta_2) * x) / (x^2 + y^2);$$
$$cos\theta_1 = ((l_2 + l_1 * cos(\theta_2)) * x + l_1 * sin(\theta_2) * y) / (x^2 + y^2);$$
$$\theta_1 = atan2(sin\theta_1, cos\theta_1);$$

$$J = \begin{pmatrix} -l_1S_1 - l_2S_{12} & -l_2S_{12} \\ -l_1C_1 - l_2C_{12} & -l_2C_{12} \end{pmatrix}$$

$$Velocity = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}; Acceleration = \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix}$$

$$ActuatorVelocity(\dot{\Omega}) = J^{-1} * Velocity$$

$$ActuatorAcceleration(\ddot{\Omega}) = J^{-1} * (Acceleration - \dot{J} * \dot{\Omega})$$

$$\dot{J} = \begin{pmatrix} -l_1C_1\dot{\theta}_1 - l_2C_{12}(\dot{\theta}_1 + \dot{\theta}_2) & -l_2C_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ l_1S_1\dot{\theta}_1 + l_2S_{12}(\dot{\theta}_1 + \dot{\theta}_2) & -l_2S_{12}(\dot{\theta}_1 + \dot{\theta}_2) \end{pmatrix}$$

$$S_1 = \mathrm{Sin}\,\theta_1 \;\; and \;\; S_{12} = \mathrm{Sin}(\theta_1 + \theta_2) \;\; and \;\; \theta_1 = hip\_angle \;\; and \;\; \theta_2 = knee\_angle$$

2. Modelling of Swing Leg

Modelling of swing leg is done in such a way that the joint Angular Acceleration and Angular Velocity profiles during transition from stance to swing and vice versa should be continuous. So the boundary conditions that must be satisfied during the transition are as follows.

i. Angular Acceleration (at the beginning of the swing phase) =Angular Acceleration (at end of stance phase);
ii. Angular Acceleration (at end of swing phase) = Angular Acceleration (at the beginning of the swing phase);
iii. Angular Velocity (at the beginning of the swing phase) =Angular Velocity (at end of stance phase);
iv. Angular Velocity (at end of swing phase) = Angular Velocity (at the beginning of the swing phase);
v. Angular Position (at the beginning of the swing phase) =Angular Position (at end of stance phase);
vi. Angular Position (at end of swing phase) = Angular Position (at the beginning of the swing phase);

For this purpose we break the trajectory as sum of two quintic equations (5 DOF).

General Equation of Trajectory

$$\theta(t) = a_0 + a_1 * t + a_2 * t^2 + a_3 * t^3 + a_4 * t^4 + a_5 * t^5$$

General solution for this equation is as follows:

The solution for a linear set of equations with six unknowns:

$$a_0 = \theta_0$$

$$a_1 = \dot{\theta}_0$$

$$a_2 = \frac{\ddot{\theta}_0}{2}$$

$$a_3 = \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3}$$

$$a_4 = \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4}$$

$$a_5 = \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5}$$

Generated Trajectory of the non-jerky motion.

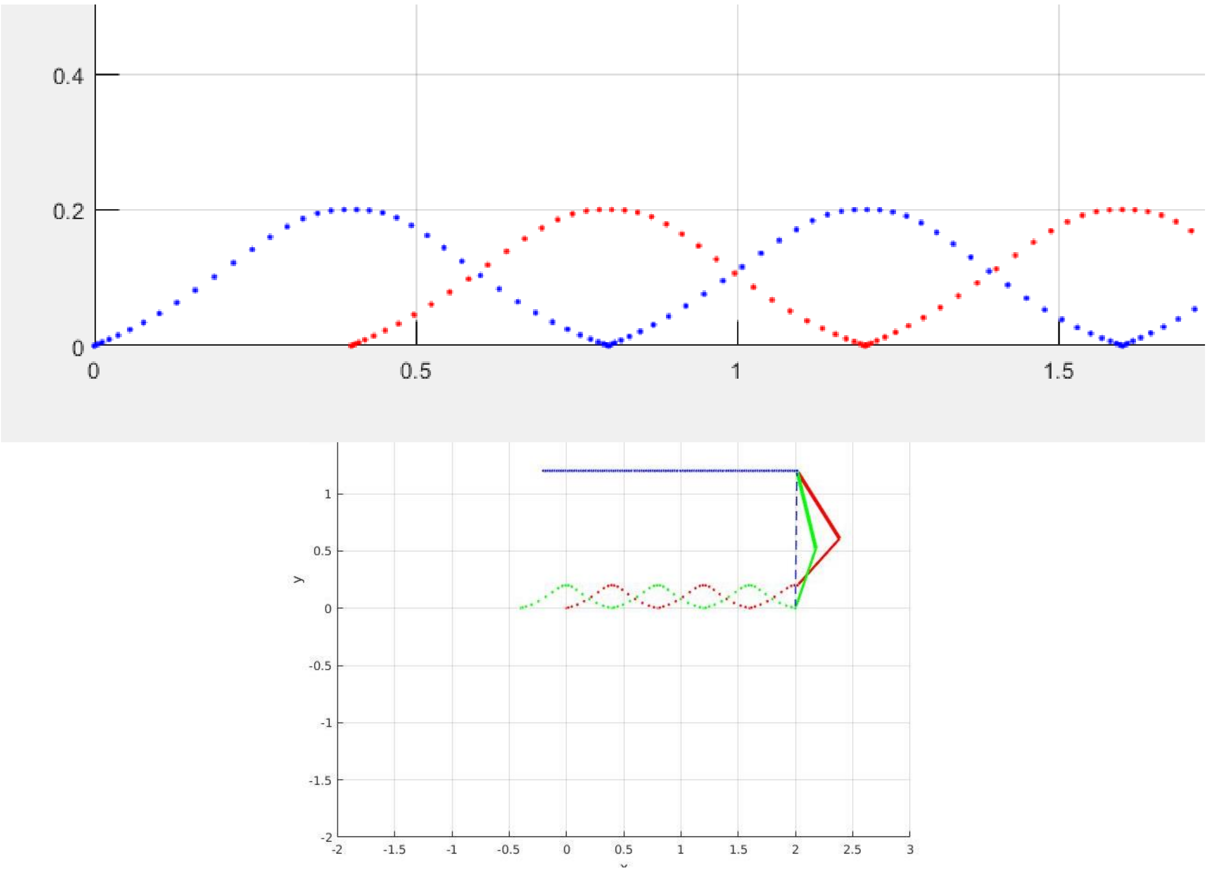

Fig 5- Desired trajectory of foot for non-jerky motion

Below contains the plots for angular acceleration, angular velocity and position for various joints.

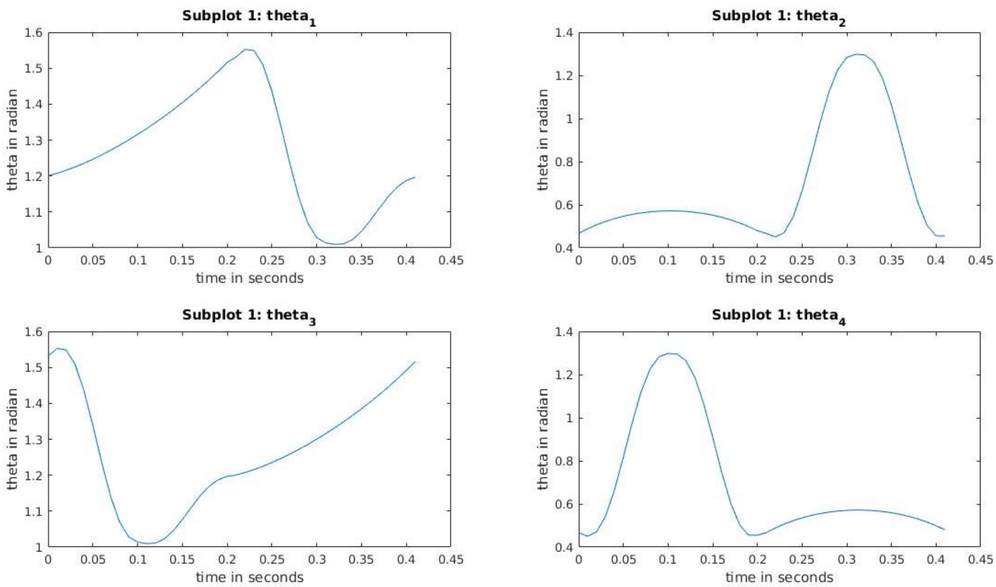

Fig 6- Position for various joints (radians) with respect to time(seconds)

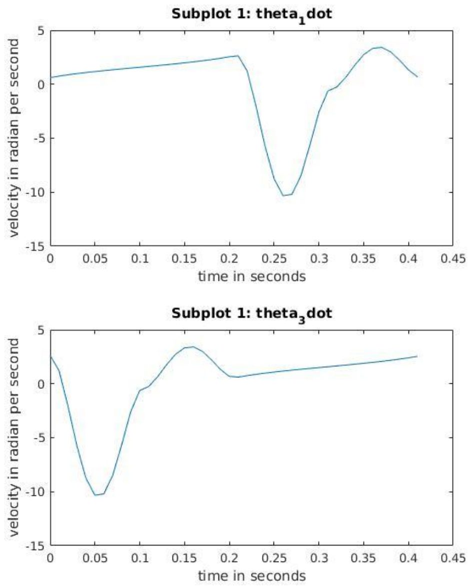

Fig 7- Angular velocity for various joints (radians per second) with respect to time(seconds)

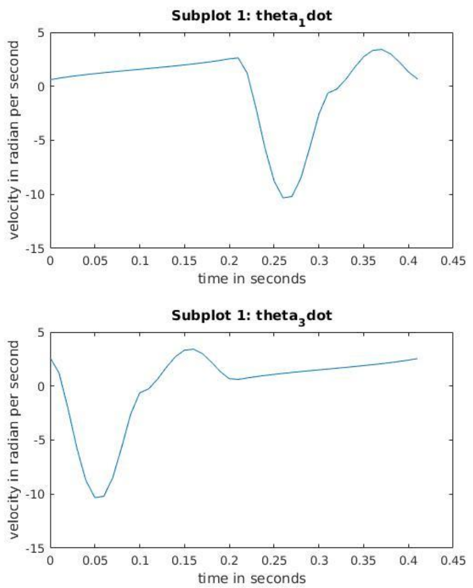

Fig 8- Angular Acceleration for various joints (radians per second^2) with respect to time (seconds)

3. Calculating Torque at different joints for State-Space modelling

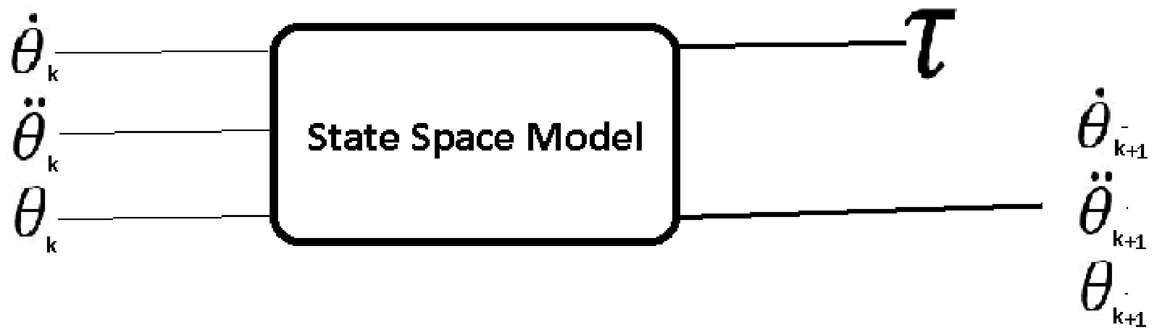

Fig 9- High Level State space model of Robot

For dynamic modelling of the robot we used the Lagrangian approach and calculated all the required joint torques as a function of position, angular velocity and angular acceleration of the joint.

Since to locate a biped in space we need 5 holonomic constraints so joint torques will function of these 5 parameters namely joint angles and displacement. The calculated torque will be required for calculating the required motor variables which will be given by controller for desired biped motion. Also the torque will contain following components namely inertial torque, Coriolis torque and Gravity torque. The calculated torque matrix is as follows.

General Equation of Dynamics

$$L = T - U$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = GeneralisedForce$$

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta})\dot{\theta} + G(\theta) + D(x)$$

Inertial Matrix

$$M(\theta)=\begin{pmatrix} \\ 2*A_2 & B_2 & 0 & 0 \\ B_2 & 2*A_3 & 0 & 0 \\ 0 & 0 & 2*C_2 & D_2 \\ 0 & 0 & D_2 & 2*C_3 \end{pmatrix}$$

$$D(x)=\begin{pmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{pmatrix}$$

$$A_1=\frac{\left(m_1+m_2\right)}{2}$$

$$A_2=\frac{(I_{c1}+I_{c2}+m_1 l_{c1}{}^2+m_2 l_2{}^2+m_2 l_{c2}{}^2)}{2}+m_2 l_1 l_{c2} C_2$$

$$A_3=\frac{(I_{c2}+m_2 l_{c2}{}^2)}{2}$$

$$B_1=-(m_1 l_{c1} S_1+m_2 l_1 S_1+m_2 l_{c2} S_{12})$$

$$B_2=(I_{c2}+m_2 l_{c2}{}^2+m_2 l_{c2}{}^2+m_2 l_1 l_{c2} C_2)$$

$$B_3=-m_2 l_{c2} S_{12}$$

$$C_1=\frac{\left(m_3+m_4\right)}{2}$$

$$C_2=\frac{(I_{c3}+I_{c4}+m_4 l_{c4}{}^2+m_4 l_3{}^2+m_3 l_{c3}{}^2)}{2}+m_4 l_3 l_{c4} C_3$$

$$C_3=\frac{I_{c4}+m_4 l_{c4}{}^2}{2}$$

$$D_1=-(m_3 l_{c3} S_3+m_4 l_3 S_3+m_4 l_{c4} S_{34})$$

$$D_2=(I_{c4}+m_4 l_{c4}{}^2+m_4 l_{c4}{}^2+m_4 l_3 l_{c4} C_4)$$

$$D_3=-m_4 l_{c4} S_{34}$$

$m_i = link\_mass$, $I_{ci} = Link\_moment\_of\_\mathrm{int}eria$, $l = link\_length$ and $l_{ci} = link\_com$

Coriolis Matrix

$$V(\theta,\dot{\theta},\dot{x}) = \begin{pmatrix} \frac{dB_1}{dt} \\ \frac{dB_3}{dt} \\ \frac{dD_2}{dt} \\ \frac{dD_3}{dt} \end{pmatrix} \left(\dot{x}\right) - \begin{pmatrix} \frac{dB_1}{d\theta_1}\dot{x} & \frac{dB_3}{d\theta_1}\dot{x} & 0 & 0 \\ \frac{dA_2}{d\theta_1}\dot{\theta}_1 + \frac{dB_1}{d\theta_1}\dot{x} & \frac{dB_2}{d\theta_2}\dot{\theta}_1 + \frac{dB_3}{d\theta_2}\dot{x} & 0 & 0 \\ 0 & 0 & \frac{dD_1}{d\theta_3}\dot{x} & \frac{dD_3}{d\theta_3}\dot{x} \\ 0 & 0 & \frac{dC_2}{d\theta_4}\dot{\theta}_3 + \frac{dD_1}{d\theta_4}\dot{x} & \frac{dC_2}{d\theta_4}\dot{\theta}_3 + \frac{dD_3}{d\theta_4}\dot{x} \end{pmatrix} \left(\dot{\theta}\right)$$

$$+ \begin{pmatrix} 2*\frac{dA_2}{dt} & \frac{dB_2}{dt} & 0 & 0 \\ \frac{dB_2}{dt} & \frac{dA_3}{dt} & 0 & 0 \\ 0 & 0 & \frac{dC_2}{dt} & \frac{dD_2}{dt} \\ 0 & 0 & \frac{dD_2}{dt} & \frac{dC_3}{dt} \end{pmatrix} \left(\dot{\theta}\right)$$

$$\frac{dA_2}{dt} = -m_2 l_1 l_{c2} S_2 \dot{\theta}_2$$

$$\frac{dB_1}{dt} = -[\dot{\theta}_1(m_1 l_{c1} C_1 + m_2 l_1 C_1 + m_2 l_{c2} C_{12}) + \dot{\theta}_2(m_2 l_{c2} C_{12})]; \frac{dB_2}{dt} = -m_2 l_2 l_{c2} S_2 \dot{\theta}_2; \frac{dB_3}{dt} = -m_2 l_2 C_{12}[\dot{\theta}_1 + \dot{\theta}_2]$$

$$\frac{dC_2}{dt} = -m_4 l_3 l_{c4} S_4 \dot{\theta}_4$$

$$\frac{dD_1}{dt} = -[\dot{\theta}_3(m_3 l_{c3} C_3 + m_4 l_3 C_3 + m_4 l_{c4} C_{34}) + \dot{\theta}_4(m_4 l_{c4} C_{34})]; \frac{dD_2}{dt} = -m_4 l_3 l_{c4} S_4 \dot{\theta}_4; \frac{dD_3}{dt} = -m_4 l_4 C_{34}[\dot{\theta}_3 + \dot{\theta}_4]$$

$$\frac{dB_1}{d\theta_1} = -m_1 l_{c1} C_1 - m_2 l_1 C_1 - m_2 l_{c2} C_{12}; \frac{dB_3}{d\theta_1} = -m_2 l_2 C_{12}$$

$$\frac{dA_2}{d\theta_2} = -m_2 l_1 l_{c2} S_2$$

$$\frac{dB_1}{d\theta_2} = -m_2 l_{c2} C_{12}; \frac{dB_2}{d\theta_2} = -m_2 l_2 l_{c2} S_2; \frac{dB_3}{d\theta_2} = -m_2 l_2 C_{12}$$

$$\frac{dD_1}{d\theta_3} = -(m_3 l_{c3} C_3 + m_4 l_3 C_3 + m_4 l_{c4} C_{34}); \frac{dD_3}{d\theta_3} = -m_4 l_4 C_{34}$$

$$\frac{dC_2}{d\theta_4} = -m_4 l_3 l_{c4} S_4$$

$$\frac{dD_1}{d\theta_4} = -m_4 l_{c4} C_{34}; \frac{dD_2}{d\theta_4} = -m_4 l_3 l_{c4} S_4; \frac{dD_3}{d\theta_4} = -m_4 l_4 C_{34}$$

Gravity Matrix

$$G(\theta) = g * \begin{pmatrix} m_1 l_{c1} C_1 + m_2 l_1 C_1 + m_2 l_{c2} C_{12} \\ m_2 l_{c2} C_{12} \\ m_1 l_{c3} C_3 + m_2 l_3 C_3 + m_2 l_{c4} C_{34} \\ m_2 l_{c4} C_{34} \end{pmatrix}$$

4. <u>Design of the robot</u>

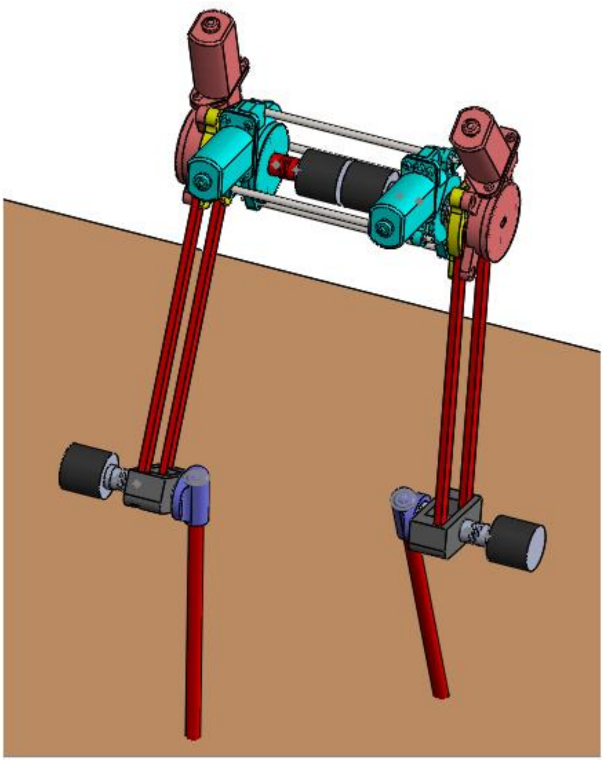

Fig 10- Isometric view of the Biped

Components used

4 Power window motors

4 Optical rotary encoders

1 Arduino Mega

1 12V 2500mAh DC Battery

2 Pressure Sensor

2 cytron motor driver

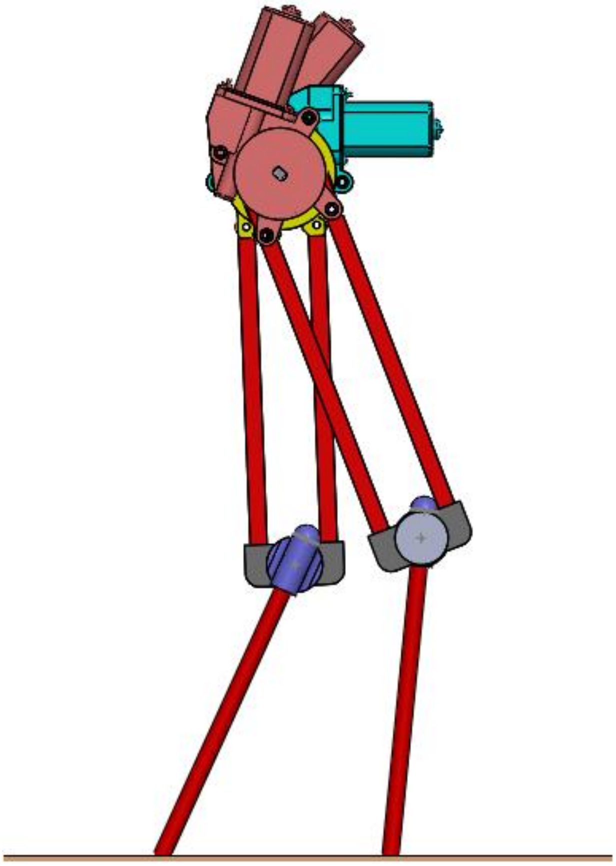

Fig 11- Side view

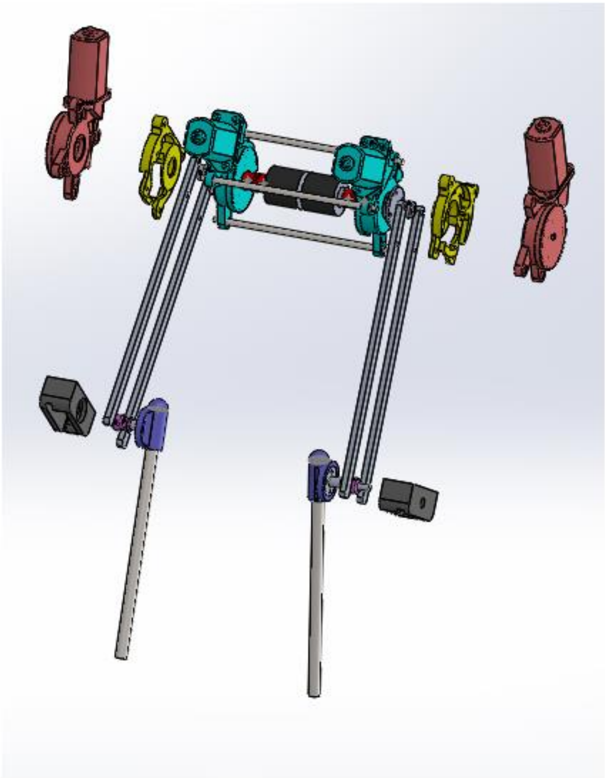

Fig 12- Exploded view showing all components used

5. Control of the Robot

The dynamics of a biped robot are more properly represented by a nonlinear differential equation. Nonetheless, it is often reasonable to make such approximations, and it also is the case that these linear methods are the ones most often used in current industrial practice. We have modelled a biped as a mechanism that is instrumented with sensors at each joint to measure the joint angle and that has an actuator at each joint to apply a torque on the neighbouring (next higher) link.1. Velocity sensors (rotary encoder) are present at the joints.

We wish to cause the manipulator joints to follow prescribed position trajectories, but the actuators are commanded in terms of torque, so we must use some kind of control system to compute appropriate actuator commands that will realize this desired motion. Almost always, these torques are determined by using feedback from the joint sensors to compute the torque required.

The robot accepts a vector of joint torques from the control system. The manipulator's sensors allow the controller to read the vectors of joint positions and joint velocities. Typically, this feedback is used to compute any servo error by finding the difference between the desired and the actual position and that between the desired and the actual velocity:

$$E = \theta_d - \theta$$

$$\dot{E} = \dot{\theta}_d - \dot{\theta}$$

The control system can then compute how much torque to require of the actuators as some function of the servo error. Obviously, the basic idea is to compute actuator torques that would tend to reduce servo errors. The most basic such criterion is that the system remain stable. For our purposes, we have defined a system to be stable if the errors remain "small" when executing various desired trajectories even in the presence of some "moderate" disturbances.

Control Law Partitioning

In this method, we have partitioned the controller into a model-based portion and a servo portion. The result is that the system's parameters (i.e., M (θ), V (θ, θ˙), and G (θ), in this case) appear only in the model-based portion and that the servo portion is independent of these parameters.

The open-loop equation of motion for the system is

$$M(\theta)\ddot{\theta} + V(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \tau$$

We wish to decompose the controller for this system into two parts. In this case, the model-based portion of the control law will make use of supposed knowledge of M(θ), V (θ, θ˙), and G (θ), This portion of the control law is set up such that it reduces the system so that it appears to be a unit mass. The second part of the control law makes use of feedback to modify the behaviour of the system. The model-based portion of the control law has the effect of making the system appear as a unit mass, so the design of the servo portion is very simple—gains are chosen to control a system composed of a single unit mass (i.e., no friction, no stiffness).

The model-based portion of the control appears in a control law of the form:

$$\tau = \alpha\tau' + \beta$$

where u and are functions or constants and are chosen so that, if τ' is taken as the new input to the system, the system appears to be a unit mass. With this structure of the control law, the system equation becomes

$$M(\theta)\ddot{\theta} + V(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \alpha\tau' + \beta$$

Clearly, in order to make the system appear as a unit mass from the τ' input, for this particular system we should choose α and β as follows:

$$\alpha = M(\theta)$$

$$\beta = V(\theta,\dot{\theta})\dot{\theta} + G(\theta)$$

By these assignments we have the system equation

$$\ddot{\theta} = \tau'$$

This is the equation of motion for a unit mass. We design a control law to compute τ' as

$$\tau' = \ddot{\theta}_d + K_v\dot{E} + K_p E$$

Combining this equation with the previous one gives

$$\ddot{E} + K_v\dot{E} + K_p E = 0$$

Under this methodology, the setting of the control gains is simple and is independent of the system parameters; that is,

$$K_v = 2\sqrt{K_p}$$

must hold for critical damping. Figure 13 shows a block diagram of the partitioned controller used to control the biped system.

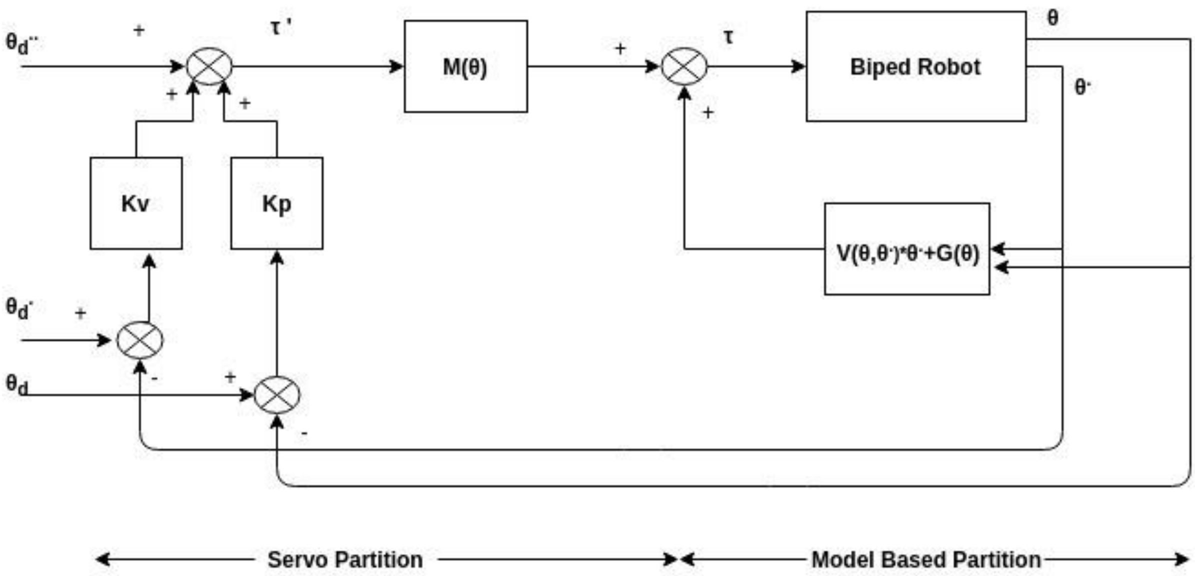

Fig 13- Control Law Partitioning Block Diagram

6. Algorithm for control in MATLAB

The flow chart (figure 14) below shows the algorithm how the simulation of the walking of a biped robot was done in MATLAB software. The dynamics of COM is defined the linear inverted pendulum approach. Then the desired trajectories of stance leg is found using *inverse kinematics* function which gives the hip and knee angles of the stance leg corresponding to the motion of COM. The desired angular velocity and angular acceleration of the hip and knee joint of stance leg is found using the *inverse_jacobian* function.

The trajectory of the swing leg is defined by the stride length, stride time and height upto which the swing leg must lift. Moreover, the curves of joint angles, their velocities and acceleration must be smooth. The desired joint trajectories of swing leg are obtained from the *trajectory_generator* function.

These four joint trajectories are given as input to *inverse_dynamics* function which gives the state space model of the robot dynamics. The state space matrices along with the error vector is given as input to the *controller* function which works of the principle of control law partitioning. The output of the controller is the control torque and joint angles which form the actual trajectory of the robot by the function *forward_kinematics.*

The simulation thus run for the time period and again the origin shifts according to the stride length of the centre of mass of the biped. The error and angular velocities etc. are plotted using *plot* command on MATLAB.

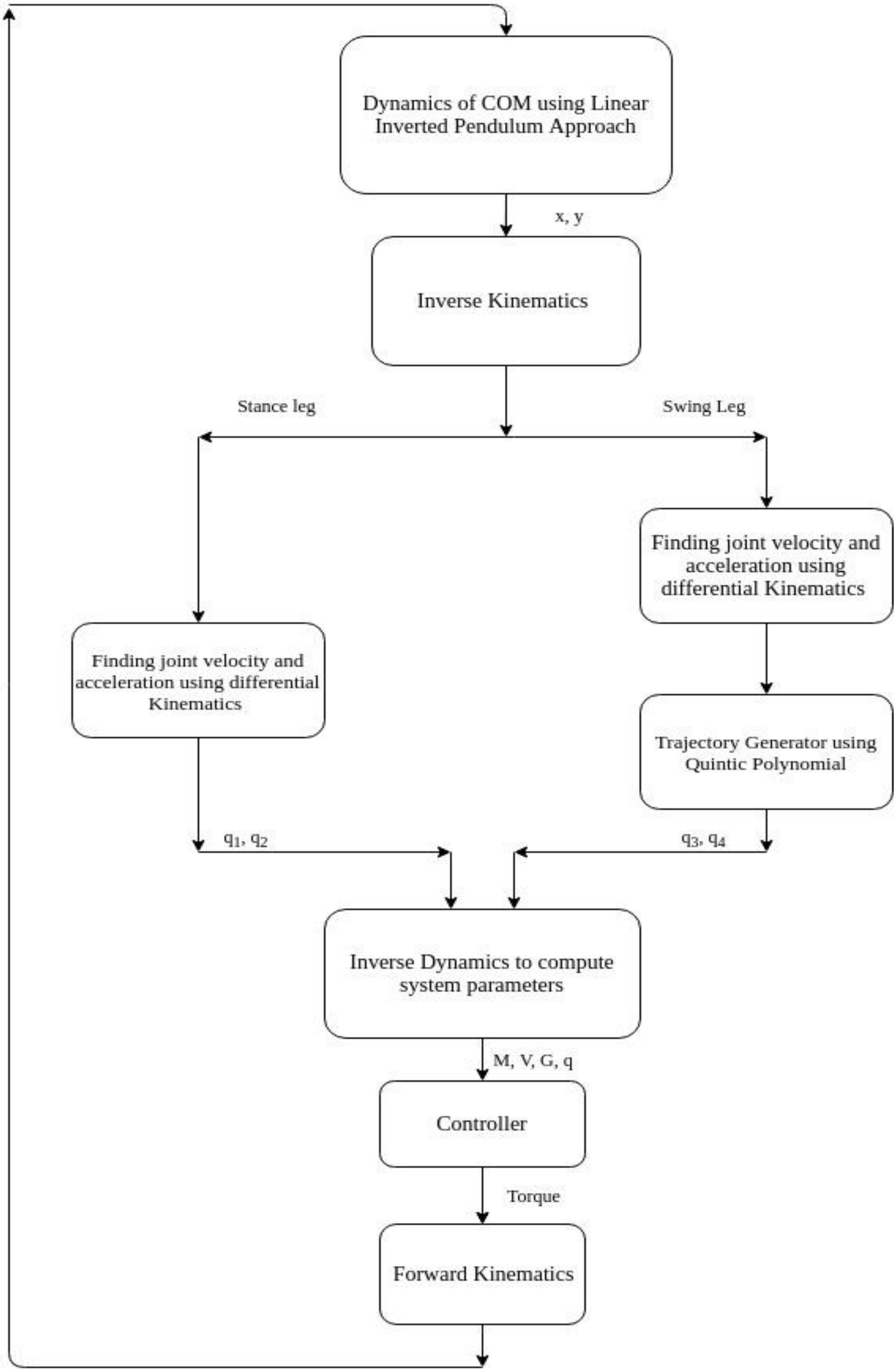

Fig 14- Flow Chart of the Algorithm of Control of Biped

Results

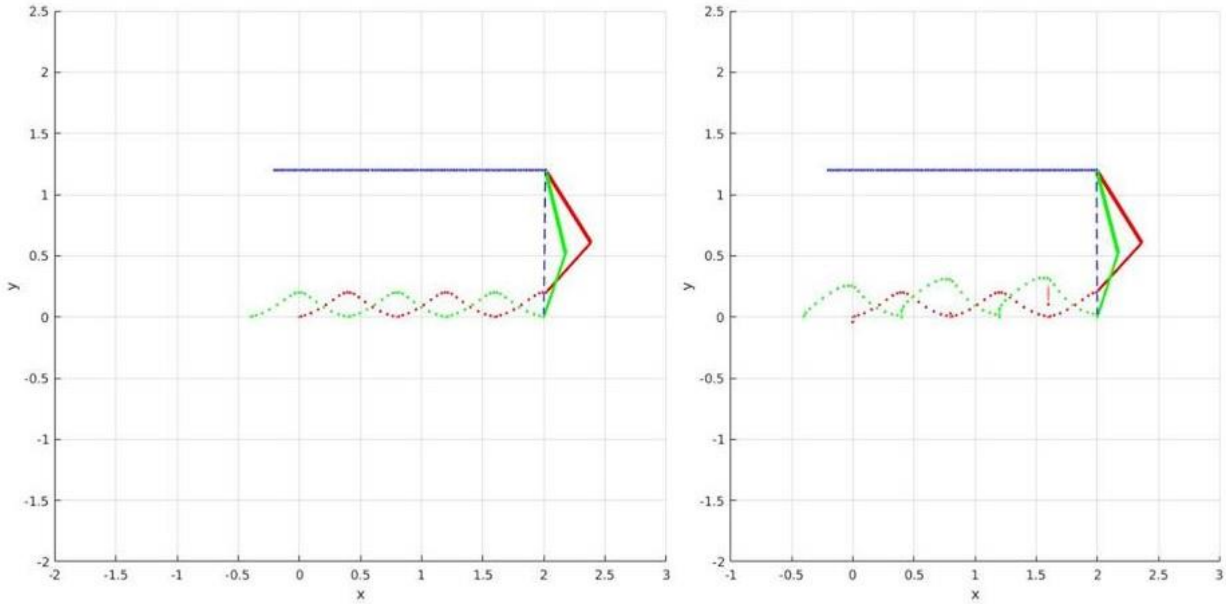

Fig –(a) Desired Trajectory of feet of Biped Robot (b) Actual Trajectory of feet of Biped Robot

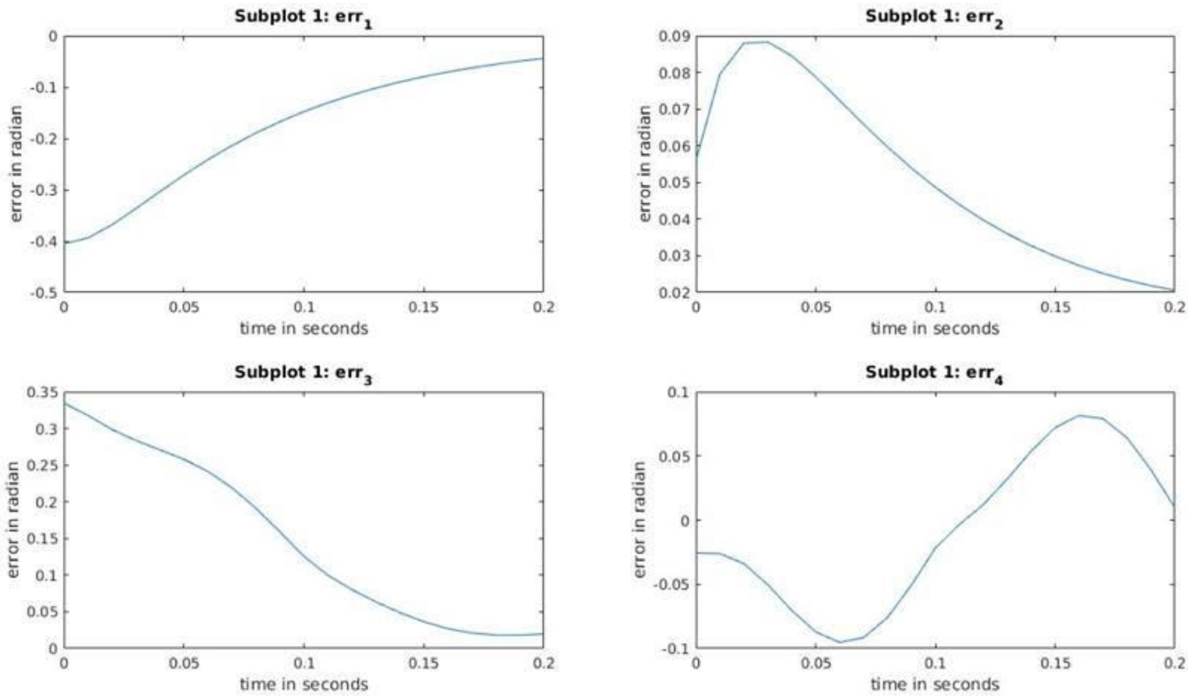

Fig- Error in the joint angles (radians) with respect to time(seconds)

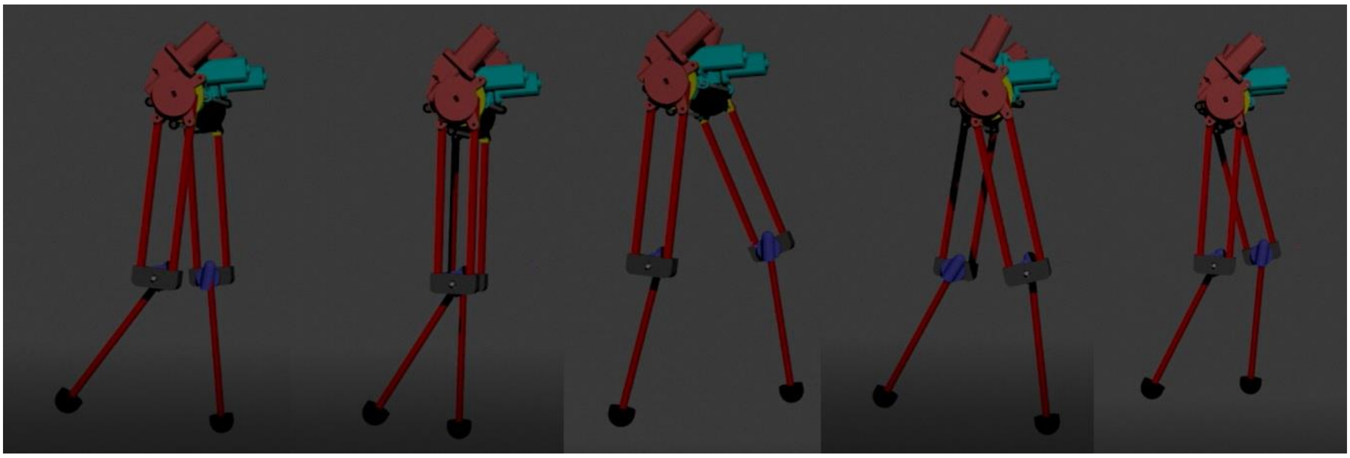

Fig- Various stages in the simulation of the actual model of Biped

Conclusion

We have completed kinematic and dynamic analysis of robot on MATLAB and also modelled a CAD model for actual prototype. We are also done with the designing of the controller and also simulated the controller for stability analysis.

Further Work

Further we will be manufacturing the actual prototype and would try to implement the controller on the actual hardware. We will also try to implement some more advanced controllers for better stability.

We will try further to implement some more algorithms for obstacle avoidance.

References

[1] Shuuji Kajita, Hirohisa Hirukawa, Kensuke Harada, Kazuhito Yokoi, ”Introduction to Humanoid Robotics ”.

[2] Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris, “Feedback Control of Dynamic Bipedal Robot Locomotion”.

[3] John J. Craig, “Introduction to Robotics-Mechanics and Control”.

[4] Hayssan Serhan and Patrick Henaff, “Muscle-Like Compliance in Knee Articulations Improves Biped Robot Walkings”, Recent Advances in Robotic Systems Guanghui Wang, IntechOpen.

[5] Miomir Vukobratovi and Branislav Borovac,“Zero-Moment Point — Thirty five years of its life”, International Journal of Humanoid Robotics, Vol. 1, No. 1, pp. 157–173, 2004.