

Cartesian Parallel Manipulator Modeling, Control and Simulation

Ayssam Elkady¹, Galal Elkobrosy², Sarwat Hanna² and Tarek Sobh¹

University of Bridgeport¹,

Alexandria University²,

USA¹,

Egypt²

1. Introduction

Parallel manipulators are robotic devices that differ from the more traditional serial robotic manipulators by their kinematic structure. Parallel manipulators are composed of multiple closed kinematic loops. Typically, these kinematic loops are formed by two or more kinematic chains that connect a moving platform to a base, where one joint in the chain is actuated and the other joints are passive. This kinematic structure allows parallel manipulators to be driven by actuators positioned on or near the base of the manipulator. In contrast, serial manipulators do not have closed kinematic loops and are usually actuated at each joint along the serial linkage. Accordingly, the actuators that are located at each joint along the serial linkage can account for a significant portion of the loading experienced by the manipulator, whereas the links of a parallel manipulator generally need not carry the load of the actuators. This allows the parallel manipulator links to be made lighter than the links of an analogous serial manipulator. The most noticeable interesting features of parallel mechanisms being:

- High payload capacity.
- High throughput movements (high accelerations).
- High mechanical rigidity.
- Low moving mass.
- Simple mechanical construction.
- Actuators can be located on the base.

However, the most noticeable disadvantages being:

- They have smaller workspaces than serial manipulators of similar size.
- Singularities within working volume.
- High coupling between the moving kinematic chains.

1.1 Prior work

Among different types of parallel manipulators, the Gough-Stewart platform has attracted most attention because it has six degrees of freedom (DOF). It was originally designed by (Stewart, 1965). Generally, this manipulator has six limbs. Each one is connected to both the base and the moving platform by spherical joints located at each end of the limb.

Actuation of the platform is typically accomplished by changing the lengths of the limbs. Although these six-limbed manipulators offer good rigidity, simple inverse kinematics, and high payload capacity, their forward kinematics are difficult to solve, position and orientation of the moving platform are coupled and precise spherical joints are difficult to manufacture at low cost.

To overcome the above shortcomings, parallel manipulators with fewer than six degrees of freedom have been investigated. For examples, (Ceccarelli, 1997) proposed a 3-DOF parallel manipulator (called CaPaMan) in which each limb is made up of a planar parallelogram, a prismatic joint, and a ball joint. But these manipulators have coupled motion between the position and orientation of the end-effector. The 3-RRR (Revolute Revolute Revolute) spherical manipulator was studied in detail by (Gosselin & Angeles, 1989).

Several spatial parallel manipulators with a rotational moving platform, called rotational parallel manipulators (RPMs), were proposed (Di Gregorio, 2001), (Karouia & Herve, 2000) and (Vischer & Clavel, 2000). (Clavel, 1988) at the Swiss Federal Institute of Technology designed a 3-DOF parallel manipulator that does not suffer from the first two of the listed disadvantages of the Stewart manipulator. Closed-form solutions for both the inverse and forward kinematics were developed for the DELTA robot (Gosselin & Angeles, 1989). The DELTA robot has only translational degrees of freedom. Additionally, the position and orientation of the moving platform are uncoupled in the DELTA design. However, the DELTA robot construction does employ spherical joints. (Tsai, 1996) presented the design of a spatial 3-UPU (Universal Prismatic Universal) manipulator and pointed out the conditions that lead to pure translational motion and its kinematics was studied further by (Di-Gregorio & Parenti-Castelli, 1998). (Tsai, 1996) and (Tsai et al., 1996) designed a 3-DOF TPM (Translational Parallel Manipulator) that employs only revolute joints and planar parallelograms. (Tsai & Joshi, 2002) analyzed the kinematics of four TPMs for use in hybrid kinematic machines. (Carricato & Parenti-Castelli, 2001) developed a family of 3-DOF TPMs. (Fang & Tsai, 2002) presented a systematic methodology for structure synthesis 3-DOF TPMs using the theory of reciprocal screws (Kim & Tsai, 2002).

Han Sung Kim and Lung-Wen Tsai (Kim & Tsai, 2002) presented a parallel manipulator called CPM (figure 1) that employs only revolute and prismatic joints to achieve translational motion of the moving platform. They described its kinematic architecture and discussed two actuation methods. For the rotary actuation method, the inverse kinematics provides two solutions per limb, and the forward kinematics leads to an eighth-degree polynomial. Also, the rotary actuation method results in many singular points within the workspace. On the other hand, for the linear actuation method, there exists a one-to-one correspondence between the input and output displacements of the manipulator. Also, they discussed the effect of misalignment of the linear actuators on the motion of the moving platform. They suggested a method to maximize the stiffness to minimize the deflection at the joints caused by the bending moment because each limb structure is exposed to a bending moment induced by the external force exerted on the end-effector.

2. Manipulator description and kinematics

2.1 Manipulator structure

The Cartesian Parallel Manipulator, shown in figure 1, consists of a moving platform that is connected to a fixed base by three limbs. Each limb is made up of one prismatic and three revolute joints and all joint axes are parallel to one another.

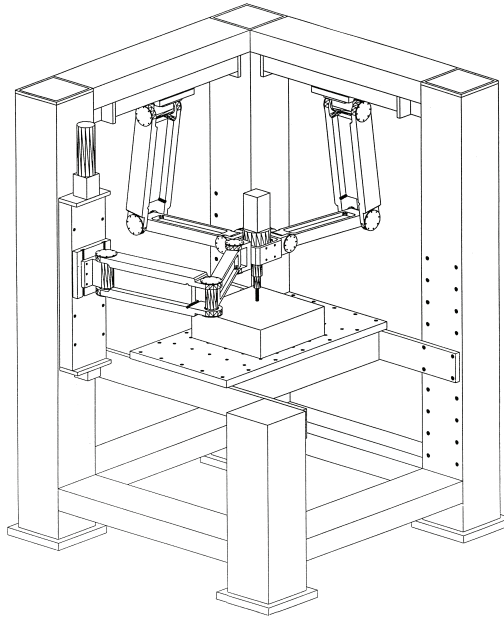


Figure 1: Assembly drawing of the prototype parallel manipulator.

2.2 Kinematic structure

The kinematic structure of the CPM is shown in figure 2 where a moving platform is connected to a fixed base by three PRRR (Prismatic Revolute Revolute Revolute) limbs. The origin of the fixed coordinate frame is located at point O and a reference frame XYZ is attached to the fixed base at this point. The moving platform is symbolically represented by a square whose length side is $2L$ defined by B_1 , B_2 , and B_3 and the fixed base is defined by three guide rods passing through A_1 , A_2 , and A_3 , respectively. The three revolute joint axes in each limb are located at points A_i , M_i , and B_i , respectively, and are parallel to the ground-connected prismatic joint axis. Furthermore, the three prismatic joint axes, passing through point A_i , for $i = 1, 2$, and 3 , are parallel to the X , Y , and Z axes, respectively. Specifically, the first prismatic joint axis lies on the X -axis; the second prismatic joint axis lies on the Y axis; and the third prismatic joint axis is parallel to the Z axis. Point P represents the center of the moving platform. The link lengths are L_1 , and L_2 . The starting point of a prismatic joint is defined by d_{0i} and the sliding distance is defined by $d_i - d_{0i}$ for $i = 1, 2$, and 3 .

2.3 Kinematics constraints

For this analysis, the position of the end-effector is considered known, and is given by the position vector $P = [x, y, z]$ which defines the location of P at the center of the moving platform in the XYZ coordinate frame. The inverse kinematics analysis produces a set of two joint angles for each limb (θ_{i1} and θ_{i2} for the i^{th} limb) that define the possible postures for each limb for the given position of the moving platform.

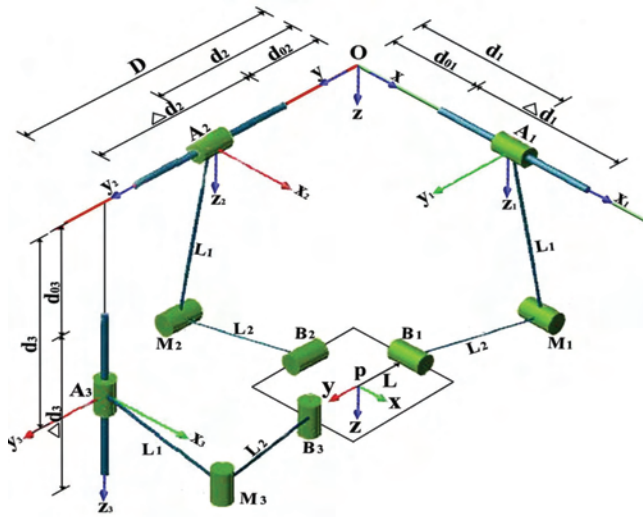


Figure 2: Spatial 3-PRRR parallel manipulator.

2.3.1 The first limb

A schematic diagram of the first limb of the CPM is sketched in figure 3, and then the relationships for the first limb are written for the position P[x, y, z] in the coordinate frame XYZ.

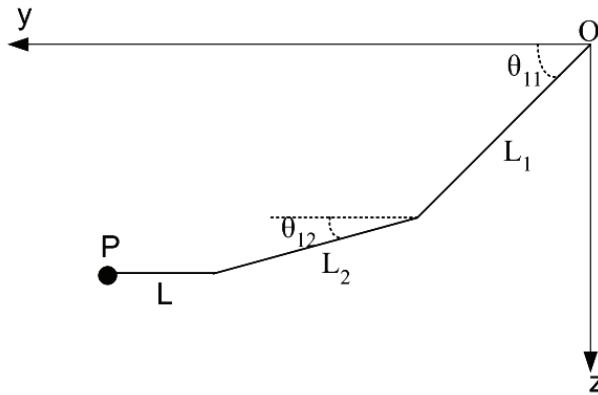


Figure 3: Description of the joint angles and link lengths for the first limb.

$$y = L_1 \cos \theta_{11} + L_2 \cos \theta_{12} + L \tag{1}$$

$$z = L_1 \sin \theta_{11} + L_2 \sin \theta_{12} \tag{2}$$

$$L_2^2 = (y - L_1 \cos \theta_{11} - L)^2 + (z - L_1 \sin \theta_{11})^2 \tag{3}$$

2.3.2 The second limb

A schematic diagram of the second limb of the CPM is sketched in figure 4, and then the relationships for the second limb are written for the position P[x, y, z] in the coordinate frame XYZ.

$$z = L_1 \cos \theta_{21} + L_2 \cos \theta_{22} \tag{4}$$

$$x = L_1 \sin \theta_{21} + L_2 \sin \theta_{22} + L \tag{5}$$

$$L_2^2 = (z - L_1 \cos \theta_{21})^2 + (x - L_1 \sin \theta_{21} - L)^2 \tag{6}$$

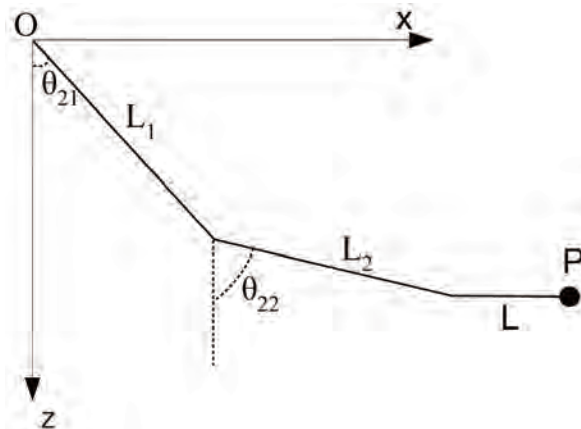


Figure 4: Description of the joint angles and link for the second limb.

2.3.3 The third limb

A schematic diagram of the third limb of the CPM is sketched in figure 5, and then the relationships for the third limb are written for the position P[x, y, z] in the coordinate frame XYZ.

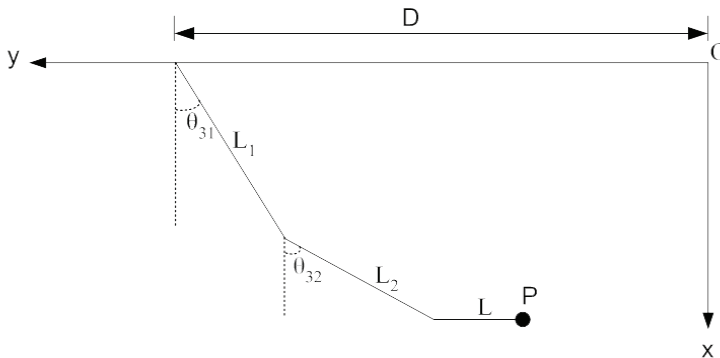


Figure 5: Description of the joint angles and link lengths for the third limb.

$$x = L_1 \cos \theta_{31} + L_2 \cos \theta_{32} \quad (7)$$

$$y = D - L - L_1 \sin \theta_{31} - L_2 \sin \theta_{32} \quad (8)$$

$$L_2^2 = (x - L_1 \cos \theta_{31})^2 + (y - D + L_1 \sin \theta_{31} + L)^2 \quad (9)$$

2.4 Linear actuation method

As described by Han Sung Kim and Lung-Wen Tsai (Kim & Tsai, 2002), for the linear actuation method, a linear actuator drives the prismatic joint in each limb whereas all the other joints are passive. This method has the advantage of having all actuators installed on the fixed base. The forward and inverse kinematic analyses are trivial since there exists a one-to-one correspondence between the end-effector position and the input joint displacements. Referring to figure 2, each limb constrains point P to lie on a plane which passes through point A_i and is perpendicular to the axis of the linear actuator. Consequently, the location of P is determined by the intersection of three planes. A simple kinematic relation can be written as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \quad (10)$$

3. Analysis of manipulator dynamics

An understanding of the manipulator dynamics is important from several different perspectives. First, it is necessary to properly size the actuators and other manipulator components. Without a model of the manipulator dynamics, it becomes difficult to predict the actuator force requirements and in turn equally difficult to properly select the actuators. Second, a dynamics model is useful for developing a control scheme. With an understanding of the manipulator dynamics, it is possible to design a controller with better performance characteristics than would typically be found using heuristic methods after the manipulator has been constructed. Moreover, some control schemes such as the computed torque controller rely directly on the dynamics model to predict the desired actuator force to be used in a feedforward manner. Third, a dynamical model can be used for computer simulation of a robotic system. By examining the behavior of the model under various operating conditions, it is possible to predict how a robotic system will behave when it is built. Various manufacturing automation tasks can be examined without the need of a real system. Several approaches have been used to characterize the dynamics of parallel manipulators. The most common approaches are based upon application of the Newton-Euler formulations, and Lagrange's equations of motion (Tsai, 1999). The traditional Newton-Euler formulation requires the equations of motion to be written once for each body of a manipulator, which inevitably leads to a large number of equations and results in poor computational efficiency. The Lagrangian formulation eliminates all of the unwanted reaction forces and moments at the outset. It is more efficient than the Newton-Euler formulation. However, because of the numerous constraints imposed by the closed loops of a manipulator, deriving explicit equations of motion in terms of a set of independent generalized coordinates becomes a prohibitive task. To simplify the problem, additional coordinates along with a set of Lagrangian multipliers are often introduced (Tsai, 1999).

3.1 Lagrange based dynamic analysis

It can be assumed that the first rod of each limb is a uniform and its mass is m_1 . The mass of second rod of each limb is evenly divided between and concentrated at joints M_i and B_i . This assumption can be made without significantly compromising the accuracy of the model since the concentrated mass model of the connecting rods does capture some of the dynamics of the rods. Also, the damping at the actuator is disregarded since the Lagrangian model does not readily accommodate viscous damping as is assumed for the actuators.

The Lagrangian equations are written in terms of a set of redundant coordinates. Therefore, the formulation requires a set of constraint equations derived from the kinematics of a mechanism. These constraint equations and their derivatives must be adjoined to the equations of motion to produce a number of equations that is equal to the number of unknowns. In general, the Lagrange multiplier approach involves solving the following system of equations (Tsai, 1999):

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j + \sum_{i=1}^k (\lambda_i \frac{\partial f_i}{\partial q_j}) \quad (11)$$

For $j=1$ to n , where

j : is the generalized coordinate index,

n : is the number of generalized coordinates,

i : is the constraint index,

q_j : is the j^{th} generalized coordinate,

k : is the number of constraint functions,

L : is the Lagrange function, where $L = T - V$,

T : is the total kinetic energy of the manipulator,

V : is the total potential energy of the manipulator,

f_i : is a constraint equation,

Q_j : is a generalized external force, and

λ_i : is the Lagrange multiplier.

Theoretically, the dynamic analysis can be accomplished by using just three generalized coordinates since this is a 3 DOF manipulator. However, this would lead to a cumbersome expression for the Lagrange function, due to the complex kinematics of the manipulator. So we choose three redundant coordinates which are θ_{11} , θ_{21} and θ_{31} beside the generalized coordinates x , y , and z . Thus we have θ_{11} , θ_{21} , θ_{31} , x , y , and z as the generalized coordinates. Equation 11 represents a system of six equations in six variables, where the six variables are λ_i for $i = 1, 2$, and 3 , and the three actuator forces, Q_j for $j = 4, 5$, and 6 . The external generalized forces, Q_j for $j=1, 2$, and 3 , are zero since the revolute joints are passive. This formulation requires three constraint equations, f_i for $i = 1, 2$, and 3 , that are written in terms of the generalized coordinates.

3.2 Derivation of the manipulator's dynamics

3.2.1 The kinetic and potential energy of the first limb

Referring to figure 6, the velocities of A_1 (the prismatic joint of the first limb), A_2 and A_3 are \dot{x} , \dot{y} and \dot{z} . The angular velocity of the rod $A_1 M_1$ is $\dot{\theta}_{11}$. We can consider the moment of

inertia of rods $A_1 M_1$, $A_2 M_2$, and $A_3 M_3$ is $I = \frac{m_1}{12} L_1^2$. m_3 is the mass of each prismatic joint (A_1 , A_2 , and A_3). So, the total kinetic energy of the first limb, T_1 , is

$$T_1 = [m_1 + m_2 + m_3] \frac{\dot{x}^2}{2} + \frac{m_2}{4} (\dot{y}^2 + \dot{z}^2) + \left(\frac{m_1}{6} + \frac{m_2}{4}\right) L_1^2 \dot{\theta}_{11}^2 \tag{12}$$

The total potential energy of the first limb, V_1 , is

$$V_1 = -\frac{m_1 + m_2}{2} g L_1 \sin \theta_{11} - \frac{m_2}{2} g z \tag{13}$$

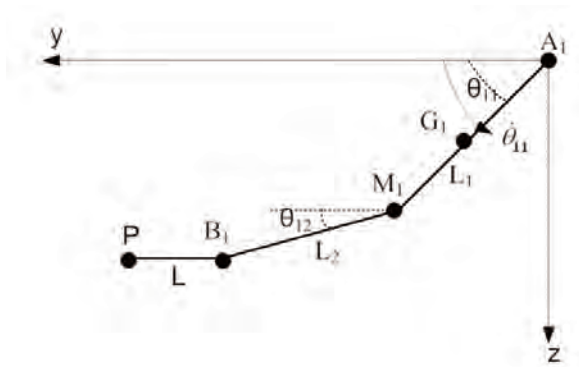


Figure 6: Schematic diagram of the first limb for the dynamic analysis.

3.2.2 The kinetic and potential energy of the second limb

Referring to figure 7, if the angular velocity of the rod $A_2 M_2$ is $\dot{\theta}_{21}$, the total kinetic energy of the second limb, T_2 is

$$T_2 = [m_1 + m_2 + m_3] \frac{\dot{y}^2}{2} + \frac{m_2}{4} (\dot{x}^2 + \dot{z}^2) + \left(\frac{m_1}{6} + \frac{m_2}{4}\right) L_1^2 \dot{\theta}_{21}^2 \tag{14}$$

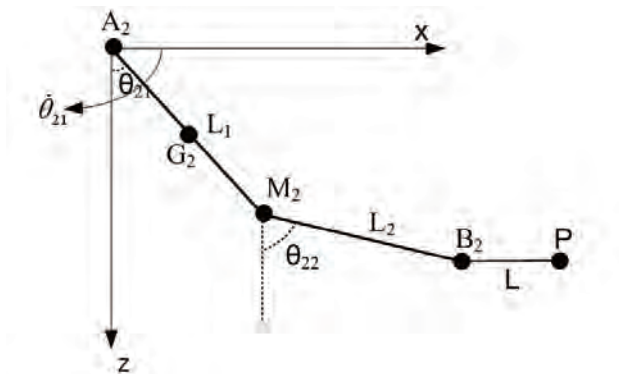


Figure 7: Schematic diagram of the second limb for the dynamic analysis.

The total potential energy of the second limb, V_2 , is given by

$$V_2 = -\frac{m_1 + m_2}{2} g L_1 \cos \theta_{21} - \frac{m_2}{2} g z \quad (15)$$

3.2.3 The kinetic and potential energy of the third limb

Referring to figure 8, the total kinetic energy of the second limb, T_3 is

$$T_3 = [m_1 + m_2 + m_3] \frac{\dot{z}^2}{2} + \frac{m_2}{4} (\dot{x}^2 + \dot{y}^2) + \left(\frac{m_1}{6} + \frac{m_2}{4}\right) L_1^2 \dot{\theta}_{31}^2 \quad (16)$$

The total potential energy of the third limb, V_3 , is

$$V_3 = -(m_1 + m_2 + m_3) g z \quad (17)$$

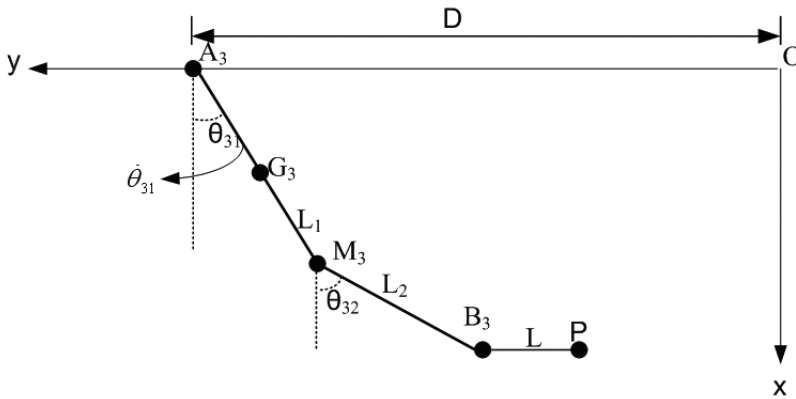


Figure 8: Schematic diagram of the third limb for the dynamic analysis.

3.2.4 Derivation of the Lagrange equation

From equations 12, 14, and 16, the total kinetic energy of the manipulator T is given by:

$$T = \frac{1}{2} [m_1 + 2m_2 + m_3 + m_4] (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \left(\frac{m_1}{6} + \frac{m_2}{4}\right) L_1^2 (\dot{\theta}_{11}^2 + \dot{\theta}_{21}^2 + \dot{\theta}_{31}^2) \quad (18)$$

where m_4 is the mass of the tool. From equations 13, 15, and 17, the total potential energy V of the manipulator is calculated relative to the plane of the stationary platform of the manipulator, and is found to be:

$$V = -\frac{m_1 + m_2}{2} g L_1 (\sin \theta_{11} + \cos \theta_{21}) - (m_1 + 2m_2 + m_3 + m_4) g z \quad (19)$$

The Lagrange function is defined as the difference between the total kinetic energy, T , and the total potential energy V : $L = T - V$

$$L = A(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + B(\dot{\theta}_{11}^2 + \dot{\theta}_{21}^2 + \dot{\theta}_{31}^2) + C(\sin \theta_{11} + \cos \theta_{21}) + Ez \quad (20)$$

Where:

$$A = \frac{1}{2}[m_1 + 2m_2 + m_3 + m_4], \quad B = \left(\frac{m_1}{6} + \frac{m_2}{4}\right)L_1^2, \quad C = \frac{m_1 + m_2}{2}gL_1 \quad \text{and} \quad E = (m_1 + 2m_2 + m_3 + m_4)g$$

3.2.5 The constraint equations

Differentiation of equation 3 with respect to time yields

$$0 = \frac{(y - L_1 \cos \theta_{11} - L)}{((y - L) \sin \theta_{11} - z \cos \theta_{11})L_1} \dot{y} + \frac{(z - L_1 \sin \theta_{11})}{((y - L) \sin \theta_{11} - z \cos \theta_{11})L_1} \dot{z} + \dot{\theta}_{11} \quad (27)$$

Differentiation of equation 6 with respect to time yields

$$0 = \frac{(z - L_1 \cos \theta_{21})}{L_1(z \sin \theta_{21} + (L - x) \cos \theta_{21})} \dot{z} + \frac{(x - L_1 \sin \theta_{21} - L)}{L_1(z \sin \theta_{21} + (L - x) \cos \theta_{21})} \dot{x} + \dot{\theta}_{21} \quad (28)$$

Differentiation of equation 9 with respect to time yields

$$0 = \frac{(x - L_1 \cos \theta_{31})}{(x \sin \theta_{31} + \cos \theta_{31}(y - D + L))L_1} \dot{x} + \frac{(y - D + L_1 \sin \theta_{31} + L)}{(x \sin \theta_{31} + \cos \theta_{31}(y - D + L))L_1} \dot{y} + \dot{\theta}_{31} \quad (29)$$

The equations 27, 28, and 29 are rearranged so as to produce:

$$\begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{21} \\ \dot{\theta}_{31} \end{bmatrix} = \Gamma \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (30)$$

Where

$$\Gamma = - \begin{bmatrix} 0 & \frac{(y - L_1 \cos \theta_{11} - L)}{((y - L) \sin \theta_{11} - z \cos \theta_{11})L_1} & \frac{(z - L_1 \sin \theta_{11})}{((y - L) \sin \theta_{11} - z \cos \theta_{11})L_1} \\ \frac{(x - L_1 \sin \theta_{21} - L)}{L_1(z \sin \theta_{21} + (L - x) \cos \theta_{21})} & 0 & \frac{(z - L_1 \cos \theta_{21})}{L_1(z \sin \theta_{21} + (L - x) \cos \theta_{21})} \\ \frac{(x - L_1 \cos \theta_{31})}{(x \sin \theta_{31} + \cos \theta_{31}(y - D + L))L_1} & \frac{(y - D + L_1 \sin \theta_{31} + L)}{(x \sin \theta_{31} + \cos \theta_{31}(y - D + L))L_1} & 0 \end{bmatrix}$$

3.2.6 Taking the derivatives of the Lagrange function with respect to θ_{11}

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{11}} \right) = 2B \ddot{\theta}_{11}, \quad \frac{\partial L}{\partial \theta_{11}} = C \cos \theta_{11}$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{11}} \right) - \frac{\partial L}{\partial \theta_{11}} = \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial \theta_{11}}) + Q_1$$

(Q_1, Q_2 and $Q_3 = 0$ since the revolute joints are passive)

$$2B \ddot{\theta}_{11} - C \cos \theta_{11} = \lambda_1 \quad (31)$$

3.2.7 Taking the derivatives of the Lagrange function with respect to θ_{21}

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{21}}\right) &= 2B\ddot{\theta}_{21}, \quad \frac{\partial L}{\partial \theta_{21}} = -C \sin \theta_{21} \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{21}}\right) - \frac{\partial L}{\partial \theta_{21}} &= \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial \theta_{21}}) + Q_2 \\ 2B\ddot{\theta}_{21} + C \sin \theta_{21} &= \lambda_2\end{aligned}\quad (32)$$

3.2.8 Taking the derivatives of the Lagrange function with respect to θ_{31}

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{31}}\right) &= 2B\ddot{\theta}_{31}, \quad \frac{\partial L}{\partial \theta_{31}} = 0 \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{31}}\right) - \frac{\partial L}{\partial \theta_{31}} &= \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial \theta_{31}}) + Q_3 \\ 2B\ddot{\theta}_{31} &= \lambda_3\end{aligned}\quad (33)$$

Rearrangement of equations 31, 32, and 33 produces:

$$2B \begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{21} \\ \ddot{\theta}_{31} \end{bmatrix} + C \begin{bmatrix} -\cos \theta_{11} \\ \sin \theta_{21} \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}\quad (34)$$

Differentiation equation 30 with respect to time yields

$$\begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{21} \\ \ddot{\theta}_{31} \end{bmatrix} = \Gamma \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \frac{d\Gamma}{dt} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

Substituting into equation 34 yields

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = 2B\Gamma \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + 2B \frac{d\Gamma}{dt} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + C \begin{bmatrix} -\cos \theta_{11} \\ \sin \theta_{21} \\ 0 \end{bmatrix}\quad (35)$$

3.2.9 Taking the derivatives of the Lagrange function with respect to X

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) &= 2A\ddot{x}, \quad \frac{\partial L}{\partial x} = 0 \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} &= \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial x}) + Q_4\end{aligned}$$

$$2A\ddot{x} = F_x - \Gamma_{11}\lambda_1 - \Gamma_{21}\lambda_2 - \Gamma_{31}\lambda_3 \quad (36)$$

where F_x , F_x and F_x are the forces applied by the actuator for the first, second and third limbs. Γ_{ij} is the (i, j) element of the Γ matrix.

3.2.10 Taking the derivatives of the Lagrange function with respect to Y

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) = 2A\ddot{y}, \quad \frac{\partial L}{\partial y} = 0$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) - \frac{\partial L}{\partial y} = \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial y}) + Q_5$$

$$2A\ddot{y} = F_y - \Gamma_{12}\lambda_1 - \Gamma_{22}\lambda_2 - \Gamma_{32}\lambda_3 \quad (37)$$

3.2.11 Taking the derivatives of the Lagrange function with respect to Z

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{z}}\right) = 2A\ddot{z}, \quad \frac{\partial L}{\partial z} = E$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{z}}\right) - \frac{\partial L}{\partial z} = \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial z}) + Q_6$$

$$2A\ddot{z} - E = F_z - \Gamma_{13}\lambda_1 - \Gamma_{23}\lambda_2 - \Gamma_{33}\lambda_3 \quad (38)$$

Rearrangement of equations 36, 37, and 38 produces:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = 2A \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix} + \Gamma^T \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$$

Substituting into equation 35 yields

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix} + \Gamma^T C \begin{bmatrix} -\cos\theta_{11} \\ \sin\theta_{21} \\ 0 \end{bmatrix} + (2AI + \Gamma^T 2B\Gamma) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \Gamma^T 2B \frac{d\Gamma}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

The dynamic equation of the whole system can be written as

$$F = M(q)\ddot{q} + G(q, \dot{q})\dot{q} + K(q) \quad (39)$$

Where

$$F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}, \quad \ddot{q} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \quad \dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \quad q = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad M(q) = 2AI + \Gamma^T 2B\Gamma, \quad G(q, \dot{q}) = \Gamma^T 2B \frac{d\Gamma}{dt}, \text{ and}$$

$$K(q) = - \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix} + \Gamma^T C \begin{bmatrix} -\cos \theta_{11} \\ \sin \theta_{21} \\ 0 \end{bmatrix}$$

where q is the 3×1 vector of joint displacements, \dot{q} is the 3×1 vector of joint velocities, F is the 3×1 vector of applied force inputs, $M(q)$ is the manipulator inertia matrix, $G(q, \dot{q})$ is the manipulator centripetal and coriolis matrix which is the 3×3 matrix of centripetal and coriolis forces, and $K(q)$ is the vector of gravitational forces which is the 3×1 vector of gravitational forces due to gravity. The Lagrangian dynamics equation, equation 39, possess important properties that facilitates analysis and control system design. Among these are (Lewis et al., 1993): the $M(q)$ is a 3×3 symmetric and positive definite matrix and the matrix $W(q, \dot{q}) = \dot{M}(q) - 2G(q, \dot{q})$ is a skew symmetric matrix.

4. Controller design

4.1 Introduction

The control problem for robot manipulators is the problem of determining the time history of joint inputs required to cause the end-effector to execute a commanded motion. The joint inputs may be joint forces or torques depending on the model used for controller design.

Position control and trajectory tracking are the most common tasks for robot manipulators; given a desired trajectory, the joint force is chosen so that the manipulator follows that trajectory. The control strategy should be robust with respect to initial condition errors, sensor noise, and modeling errors. The primary goal of motion control in joint space is to make the robot joints q track a given time varying desired joint position q_d . Rigorously, we say that the motion control objective in joint space is achieved provided that $\lim_{t \rightarrow \infty} e(t) = 0$

where $e(t) = q_d(t) - q(t)$ denotes the joint position error. Although the dynamics of the manipulator's equation is complicated, it nevertheless is an idealization, and there are a number of dynamic effects that are not included in this equation. For example, friction at the joints is not accounted for in this equation and may be significant for some manipulators. Also, no physical body is completely rigid. A more detailed analysis of robot dynamics would include various sources of flexibility, such as elastic deformation of bearings and gears, deflection of the links under load, and vibrations.

4.2 PID control versus model based control

The PID controller is a single-input/single-output (SISO) controller that produces a control signal that is a sum of three terms. The first term is proportional (P) to the positioning error, the second term is proportional to the integral (I) of the error, and the third is proportional to the derivative (D) of the error. The PID (or PD) type is usually employed in industrial robot manipulators because it is easy to implement and requires little computation time during real time operation. This approach views each actuator of the manipulator independently, and essentially ignores the highly coupled and nonlinear nature of the manipulator. The error between the actual and desired joint position is used as feedback to control the actuator associated with each joint. However, independent joint controllers can not achieve a satisfactory performance due to their inherent low rejection of disturbances and parameter variations. Because of such limitation, model based control algorithms were proposed (Sciavicco et al., 1990) that have the potential to perform better than independent

joint controllers that do not account for manipulator dynamics. However, the difficulty with the model based controller is that it requires a good model of the manipulator inverse dynamics and good estimates of the model parameters, making this controller more complex and difficult to implement than the non-model based controller. The model based control scheme was intensively experimentally tested for example the experimental evaluation of computed torque control was presented in (Griffiths et al., 1989).

4.3 PD control with position and velocity reference

The first PD control law is based on the position and velocity error of each actuator in the joint space. To implement the joint control architecture, the values for the joint position error and the joint rate error of the closed chain system are used to compute the joint control force F (Spong & Vidyasagar, 1989).

$$F = K_p e + K_D \dot{e} \quad (40)$$

Where $e = q_d - q$, which is the vector of position error of the individual actuated joints, and $\dot{e} = \dot{q}_d - \dot{q}$, which is the vector of velocity error of the individual actuated joints. Where \dot{q}_d and q_d are the desired joint velocities and positions, and K_D and K_p are 3×3 diagonal matrices of velocity and position gains. Although this type of controller is suitable for real time control since it has very few computations compared to the complicated nonlinear dynamic equations, there are a few downsides to this controller. It needs high update rate to achieve reasonable accuracy. Using local PD feedback law at each joint independently does not consider the couplings of dynamics between robot links. As a result, this controller can cause the motor to overwork compared to other controllers presented next.

4.4 PD Control with gravity compensation

This is a slightly more sophisticated version of PD control with a gravitational feedforward term. Consider the case when a constant equilibrium posture is assigned for the system as the reference input vector q_d . It is desired to find the structure of the controller which ensures global asymptotic stability of the above posture. The control law F is given by (Spong & Vidyasagar, 1989):

$$F = K_p e + K_D \dot{e} + K(q_d) \quad (41)$$

It has been shown (Spong, 1996) that the system is asymptotically stable but it is only proven with constant reference trajectories. Although with varying desired trajectories, this type of controller cannot guarantee perfect tracking performance. Hence, more dynamic modeling information is needed to incorporate into the controller.

4.5 PD control with full dynamics feedforward terms

This type of controller augments the basic PD controller by compensating for the manipulator dynamics in the feedforward way. It assumes the full knowledge of the robot parameters. The key idea for this type of controller is that if the full dynamics is correct, the resulting force generated by the controller will also be perfect. The controller is given by (Gullayanon, 2005)

$$F = M(q_d)\ddot{q}_d + G(q_d, \dot{q}_d)\dot{q}_d + K(q_d) + K_p e + K_D \dot{e} \quad (42)$$

If the dynamic knowledge of the manipulator is accurate, and the position and velocity error terms are initially zero, the applied force $F = M(q_d)\ddot{q}_d + G(q_d, \dot{q}_d)\dot{q}_d + K(q_d)$ is sufficient to maintain zero tracking error during motion. This controller is very similar to the computed torque controller, which is presented next. The difference between these two controllers is the location of the position and velocity correction terms. This controller is less sensitive to any mass changes in the system. For example, if the robot picks up a heavy load in the middle of its operation, this controller is likely to respond to this change slower compared to computed torque controller. The advantage of this controller is that once the desired trajectory for a given task has been specified, then the feedforward terms relying on the robot dynamics $M(q_d)\ddot{q}_d + G(q_d, \dot{q}_d)\dot{q}_d + K(q_d)$ can be computed offline to reduce the computational burden.

4.6 Computed torque control

This controller is developed for the manipulator to examine if it is possible to improve the performance of the trajectory tracking of the manipulator by utilizing a more complete understanding of the manipulator dynamics in the controller design. This controller employs a computed torque control approach, and it uses a model of the manipulator dynamics to estimate the actuator forces that will result in the desired trajectory. Since this type of controller takes into account the nonlinear and coupled nature of the manipulator, the potential performance of this type of controller should be quite good. The disadvantage of this approach is that it requires a reasonably accurate and computationally efficient model of the inverse dynamics of the manipulator to function as a real time controller. The controller computes the dynamics online, using the sampled joint position and velocity data. The key idea is to find an input vector, F , as a function of the system states, which is capable to realize an input/output relationship of linear type. It is desired to perform not a local linearization but a global linearization of system dynamics obtained by means of a nonlinear state feedback. Using the computed torque approach with a proportional-derivative (PD) outer control loop, the applied actuator forces are calculated at each time step using the following force law as described by Lewis, 1993:

$$F = M(q)[\ddot{q}_d + K_D\dot{e} + K_P e] + G(q, \dot{q})\dot{q} + K(q) \quad (43)$$

where F is the force applied to input links, K_D is the diagonal matrix of the derivative gains, K_P is the diagonal matrix of the proportional gains, and e is the vector of the position errors of the input links, $e = q_d - q$. To show that the computed torque control scheme linearizes the controlled system, the force computed by equation 43 is substituted into equation 39, yielding: $M(q)\ddot{q} = M(q)\ddot{q}_d + M(q)[K_D\dot{e} + K_P e]$. Then multiplying each term by $M^{-1}(q)$, and substituting the relationship, $\ddot{e} = \ddot{q}_d - \ddot{q}$, provides the following linear relationship for the error:

$$\ddot{e} + k_D\dot{e} + k_P e = 0 \quad (44)$$

This relationship can be used to select the gains to give the desired nature of the closed loop error response since the solution of equation 44 provides a second order damped system with a natural frequency of ω_n , and a damping ratio of ζ where:

$$\omega_n = \sqrt{K_p}, \quad \zeta = \frac{K_D}{2\sqrt{K_p}} \quad (45)$$

Since the equation 44 is linear, it is easy to choose K_D and K_p so that the overall system is stable and $e \rightarrow 0$ exponentially as $t \rightarrow \infty$. Usually, if K_D and K_p are positive diagonal matrices, the control law 43 applied to the system 39 results in exponentially trajectory tracking.

It is customary in robot applications to take the damping ration $\zeta = 1$ so that the response is critically damped. This produces the fastest non-oscillatory response. The natural frequency ω_n determines the speed of the response. So, the values for the gain matrices K_D and K_p are determined by setting the gains to maintain the following relationship:

$$K_D = 2\sqrt{K_p} \quad (46)$$

If the error response is critically damped. Hence, the general solution of equation 44 is:

$$e(t) = (c_1 + c_2 t) e^{-\frac{K_p t}{2}} \quad (47)$$

where C_1 and C_2 are constants.

5. Trajectory planning and simulation

5.1 Introduction

The computer simulation is the first step to verify the performance of the controllers because it is an ideal way of comparing performance of various motion controllers. Although computer simulation has much fewer disturbances compared to real experiments, factors such as the integration estimation and sampling rate can cause the controllers to behave differently than the mathematical prediction.

5.2 Tracking accuracy

In this research, the main purpose for developing the motion controllers is to obtain a good trajectory tracking capability. The performance of each control method is evaluated by comparing the tracking accuracy of the end-effector. The tracking accuracy is evaluated by the *Root Square Mean Error* (RSME). The end-effector error is defined as

$$E_{xyz} = \sqrt{(e_x^2 + e_y^2 + e_z^2)} \quad (48)$$

where e_x , e_y and e_z are the position errors in x-, y-, and z-axis given in manipulator's workspace coordinates.

$$RSME = \sqrt{\frac{\sum E_{xyz}^2}{n}} \quad (49)$$

Where n is the number of the samples.

5.3 Trajectory planning

In controlling the manipulator using any types of joint space controllers, any sudden changes in desired joint angle, velocity, or acceleration can result in sudden changes of the

commanded force. This can result in damages of the motors and the manipulator. Here, the manipulator is given a task to move along careful preplanned trajectories without any external disturbances or no interaction with environment. The desired trajectory is simulated using all motion controllers presented in Section 4 and the tracking accuracy *RSME* is obtained to be compared. The simulation is used to find a set of minimum proportional gain K_P and derivative gain K_D that minimized *RSME*. It must be considered that the actuators can not generate forces larger than 120 Newtons. The values of the physical kinematic and dynamic parameters of the CPM are given in table 1 and table 2.

Parameters	$L(m)$	$L_1(m)$	$L_2(m)$	$D(m)$
Values	0.105	0.5	0.373	0.9144

Table 1: Kinematic parameters of the CPM

Parameters	$m_1(kg)$	$m_2(kg)$	$m_3(kg)$	$m_4(kg)$
Values	1.892994	0.695528	0.2	0.3

Table 2: Dynamic parameters of the CPM.

The sample trajectory of the end-effector is chosen to be a circular path with the radius of 0.175 meters and its center is $O(0.425, 0.425, 0.3)$. This path is designed to be completed in 4 seconds when the end-effector reaches the starting point $P1(0.6, 0.425, 0.3)$ again with constant angular velocity $\omega = 0.5\pi$ rad/sec. The end-effector path is shown in figure 9. The desired end-effector position along x-axis is $x = 0.425 + 0.175 \cos(\omega t)$ meters, along y-axis is $y = 0.425 + 0.175 \sin(\omega t)$ meters and along z-axis is $Z=0.3$ meters. The desired force obtained from the actuators to move the end-effector along the desired trajectory is shown in figure 10.

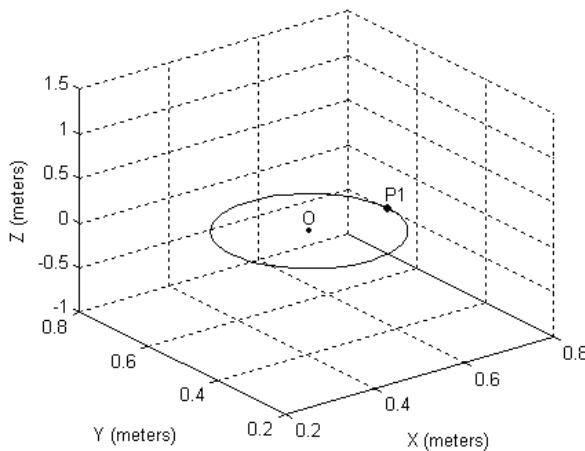


Figure 9: End-effector path for the circular trajectory.

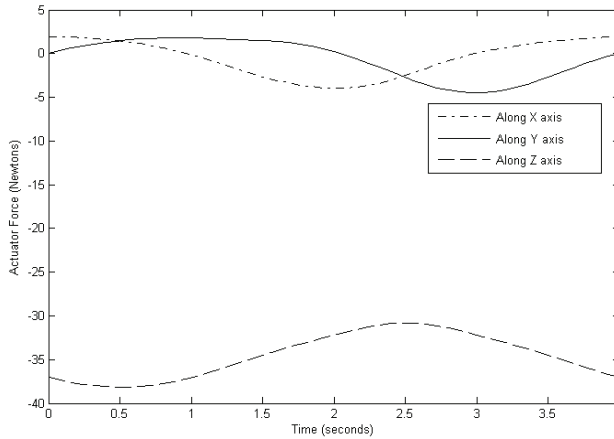


Figure 10: The desired force obtained from the actuators

5.4 Simulation results

To investigate each controller's performance, computer simulation, carried out in Matlab, is used in this thesis. The robot dynamic model, developed earlier, is constructed. The sample trajectory, presented in the previous section, is generated and stored offline. The environmental disturbances are ignored and full knowledge of the manipulator dynamics can be assumed. Hence, the optimal performance of each controller can be obtained and compared. The simulation results are presented in table 3.

5.4.1 PD control with position and velocity reference

It was required that the robot achieved the desired trajectory with a position error less than 3×10^{-3} m after 0.3 seconds.

5.4.2 PD control with gravity compensation

It was required that the robot achieved the desired trajectory with a position error less than 3×10^{-4} m after 0.3 seconds.

5.4.3 PD control with full dynamics Feedforward terms

It was required that the robot achieved the desired trajectory with a position error less than 10^{-5} m after 0.3 seconds.

5.4.4 Computed torque control

The initial conditions of the error and its derivative of our sample trajectory of the End-effector, $e(0) = 0$, and $\dot{e}(0) = \dot{e}_0$, are used to find c_1 and c_2 in equation 47. Then, the solution of this equation is

$$e = \dot{e}_0 t e^{-\frac{K_D t}{2}} \quad (50)$$

Equation 50 suggests that the derivative gain K_D should be a maximum value to achieve the desired critical damping but the actuator force cannot exceed more than 120 Newtons. Using this criterion, the simulation results are presented in table 3. The position and velocity errors of the end-effector obtained from the controllers and the actuator forces required by these controllers are shown in figures 11 to 22.

Controller	K_P	K_D	Position RSME	Velocity RSME
Pd Control with Position and Velocity Reference	12691	436	0.0027	0.0223
Pd Control with Gravity Compensation	8507	436	3.4804×10^{-4}	0.021
Pd Control with Full Dynamics Feedforward	7053	436	3.0256×10^{-4}	0.0182
Computed Torque Control	2550.25	101	2.3469×10^{-4}	0.0161

Table 3: The performance of various controllers

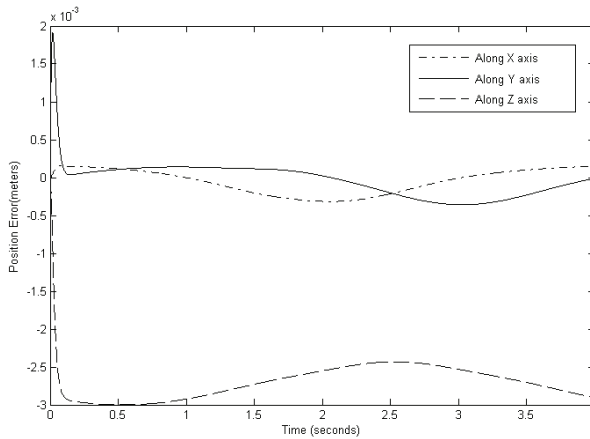


Figure 11 Position error of the end-effector obtained from the Simple PD Controller.

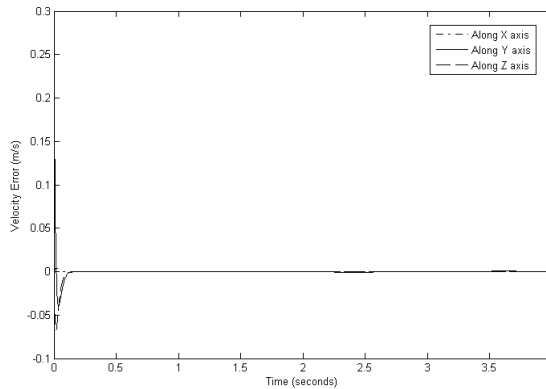


Figure 12: Velocity error of the end-effector obtained from the Simple PD Controller.

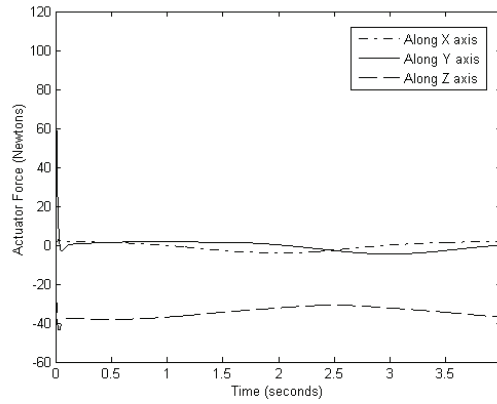


Figure 13: The actuator force required by the Simple PD Controller.

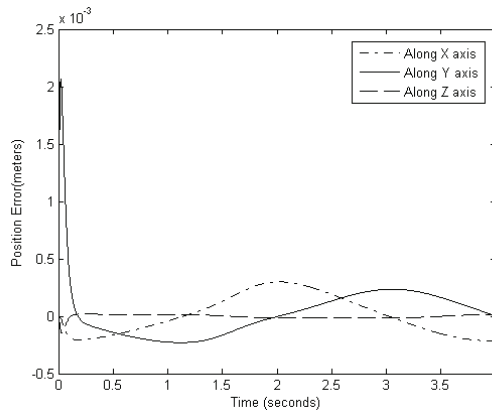


Figure 14: Position error of the end-effector obtained from the second PD Controller.

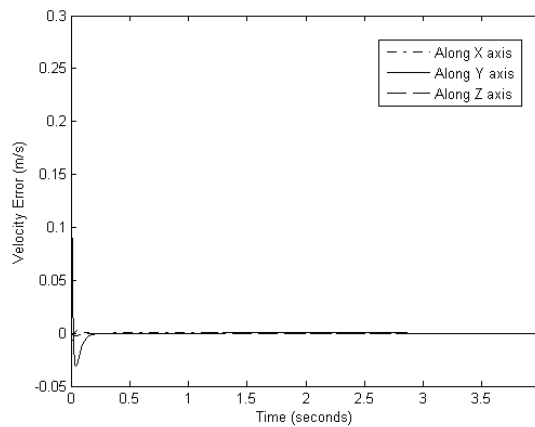


Figure 15: Velocity error of the end-effector obtained from the second PD Controller.

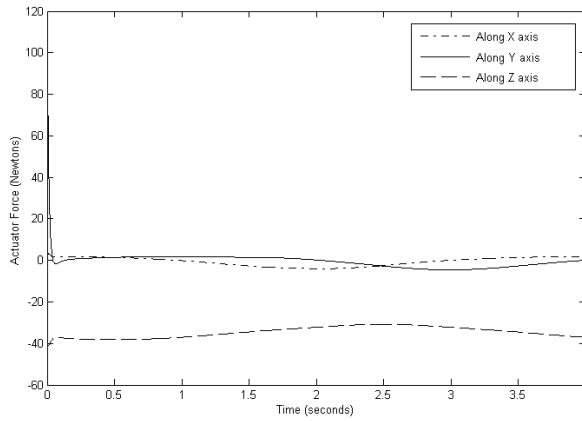


Figure 16: The actuator force required by the second PD Controller.

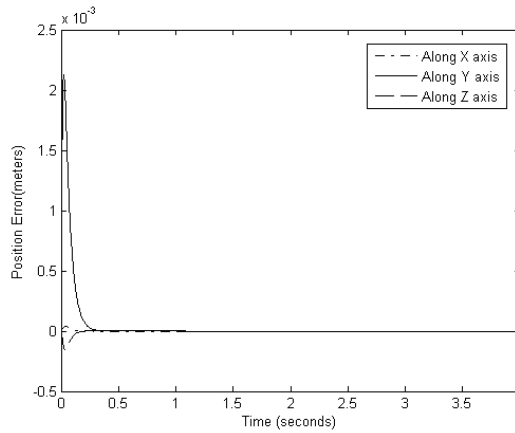


Figure 17: Position error of the end-effector obtained from the third PD Controller.

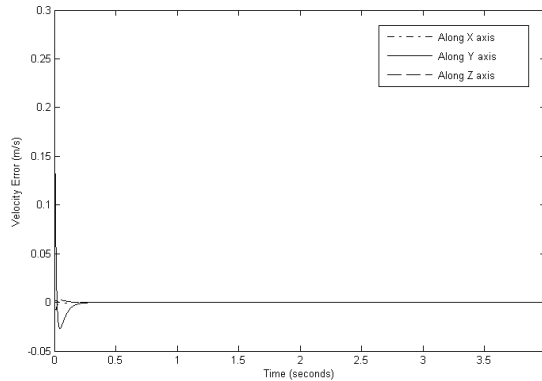


Figure 18: Velocity error of the end-effector obtained from the third PD Controller.

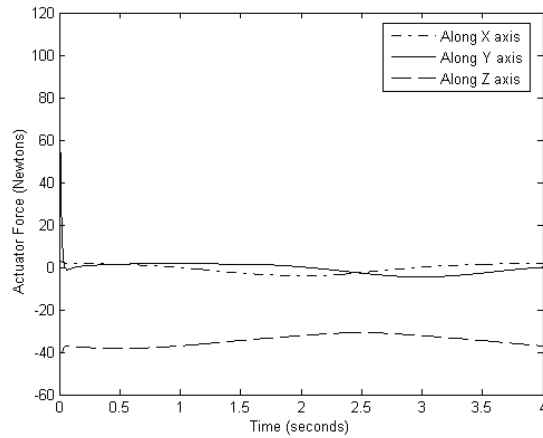


Figure 19: The actuator force required by the third PD Controller.

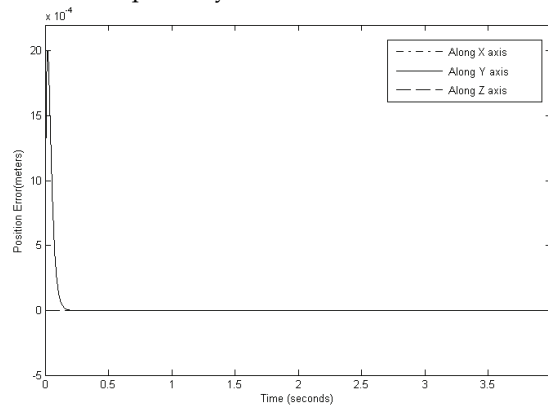


Figure 20: Position error of the end-effector obtained from the computed torque controller.

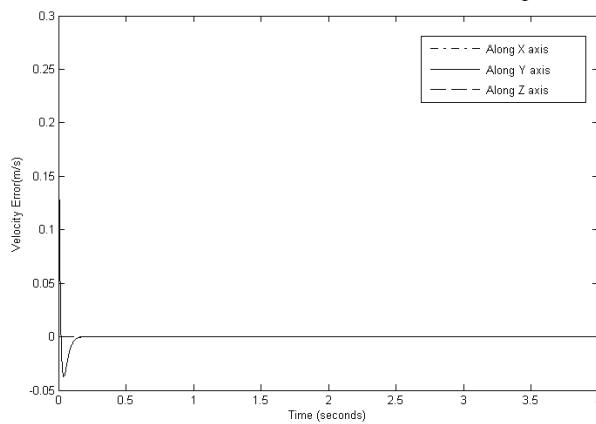


Figure 21: Velocity error of the end-effector obtained from the computed torque controller.

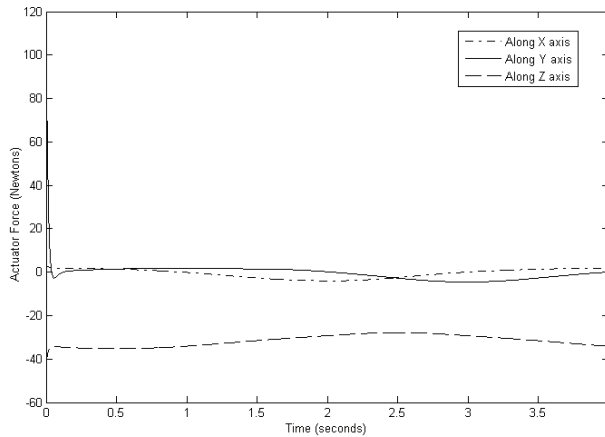


Figure 22: The actuator force required by the computed torque controller.

6. Conclusions

The research presented in this chapter establishes the CPM as a viable robotic device for three degrees of freedom manipulation. The manipulator offers the advantages associated with other parallel manipulators, such as light weight construction; while avoiding some of the traditional disadvantages of parallel manipulators such as the extensive use of spherical joints and coupling of the platform orientation and position. The CPM employs only revolute and prismatic joints to achieve translational motion of the moving platform. The main advantages of this parallel manipulator are that all of the actuators can be attached directly to the base, closed-form solutions are available for the forward and inverse kinematics, and the moving platform maintains the same orientation throughout the entire workspace. From simulations done in this research, performance of various motion controllers are studied and compared. Although the simple PD controller with only position and velocity reference is easy to implement and no knowledge of the system is needed to develop this type of controller, the tracking ability is very poor compared to the rest of the controllers used in this thesis. At the next step, when partial dynamic modeling information is incorporated into the controller, the PD controller with gravity compensation is implemented. The simulation results show a significant improvement in tracking ability from a simple PD controller. Next, the verification is needed to determine if complete mathematical modeling knowledge is needed to give the controller complete advantage in motion control. Hence, the PD controller with full dynamic feedforward terms and computed torque controller are implemented and put to the test. The model based controllers such as computed torque and PD control with full dynamic feedforward terms can generate force commands more intelligently and accurately than simple non-model based controllers. Hence, the need for studying dynamics of robot manipulator as well as having a good understanding of various basic motion controller theories are important in designing and controlling motion of the robot to achieve the highest quality and quantity of work. The simulation results show that the computed torque controller gives the best

performance. This is a result of the computed torques canceling the nonlinear components of the controlled system. From the observations seen in this work, one can see the motivation for engineers to develop more advanced controllers that not only know the dynamic model of the manipulator, but can also detect if the dynamic is changed and can tune itself accordingly (i.e. adaptive control).

7. Future work

1. The effect of some unknown parameters such as the friction and the nonlinear factors introduced by the motors and the gear boxes which may be obtained by experimental measurements and through the identification methods can be studied.
2. The performance of model based control relies on an accurate model of a system. However, identifying the accurate dynamic model of a system is very difficult. Therefore, effective controllers for the versatile application of parallel robots should be developed. Adaptive control has the potential to improve the tracking accuracy because it updates the unknown parameters online. Adaptive control algorithm is too complicated to be utilized in high speed applications. In such applications, robust independent joint control is a prospective method to improve the performance of simple PD control.
3. *Adaptive Neuro Fuzzy Inference System (ANFIS)* controller can be used for each active joint to generate the required control system, then its performance is compared with the conventional controllers. Although many of model based methods have been found and they provide satisfactory solutions, these solutions have been subordinated to the development of the mathematical theories that deal with over idealized problems bearing little relation to practice.

8. Acknowledgment

The authors would like to thank Prof. Han Sung Kim for his valuable suggestions and his kind assistance during this work.

9. References

- Carricato, M., and Parenti-Castelli, V., (2001), "A Family of 3-DOF Translational Parallel Manipulators", Proceedings of the 2001 ASME Design Engineering Technical Conferences, Pittsburgh, PA, DAC-21035.
- Ceccarelli, M., (1997), "A New 3 D.O.F. Spatial Parallel Mechanism", Mechanism and Machine Theory, Vol. 32, No. 8, pp. 895-902.
- Clavel, R., (1988), "Delta, A Fast Robot with Parallel Geometry", Proceedings of the 18th International Symposium on Industrial Robots, pp. 91-100.
- Di Gregorio, R., (2001), "A New Parallel Wrist Using only Revolute Pairs: The 3 RUU Wrist", Robotica, Vol. 19, No. 3, pp. 305-9.
- Di Gregorio, R. and Parenti-Castelli, V., (1998), "A Translational 3-DOF Parallel Manipulator", in Advances in Robot Kinematics, Edited by J. Lenarcic and M. L. Husty, Kluwer Academic Publishers, London, pp. 49-58.

- Fang, Y. and Tsai, L. W., 2002, "Enumeration of 3-DOF Translational Parallel Manipulators Using the Theory of Reciprocal Screws", accepted for publication in ASME Journal of Mechanical Design.
- Gosselin, C. and Angeles, J., 1989, "The Optimum Kinematic Design of a Spherical Three-Degree-of-Freedom Parallel Manipulator", ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 111, No. 2, pp. 202-7.
- Griffiths, J.D., An. C.H., Atkeson, C.G. and Hollerbach, J.M., 1989, "Experimental evaluation of feedback and computed torque control", International Journal of Robotics and Automation, 5(3):368-373, June.
- Gullayanon R., 2005, "Motion Control of 3 Degree-of-Freedom Direct-Drive Robot.", A master thesis presented to the School of Electrical and Computer Engineering, Georgia Institute of Technology.
- Karouia, M., and Herve, J. M., 2000, "A Three-DOF Tripod for Generating Spherical Rotation", in Advances in Robot Kinematics, Edited by J. Lenarcic and V. Parenti-Castelli, Kluwer Academic Publishers, pp. 395-402.
- Kim H.S., and Tsai L.W., 2002, "Design optimization of a Cartesian parallel manipulator", Department of Mechanical Engineering, Bourns College of Engineering, University of California.
- Lewis, F., Abdallah, C. and Dawson, D., 1993, "Control of Robot Manipulators", MacMillan Publishing Company.
- Pierrot, F., Reynaud, C. and Fournier, A., 1990, "Delta: A Simple and Efficient Parallel Robot", Robotica, Vol. 6, pp. 105-109.
- Sciavicco, L., Chiacchio, P. and Siciliano, B., 1990, "The potential of model-based control algorithms for improving industrial robot tracking performance", IEEE International Workshop on Intelligent Motion Control, pp. 831-836, August.
- Spong, M. W., 1996, "Motion Control of Robot Manipulators", University of Illinois at Urbana-Champaign.
- Spong, M.W. and Vidyasagar, M., 1989, "Robot dynamics and control", John Wiley & Sons.
- Stewart, D., 1965, "A Platform with Six Degrees of Freedom", Proceedings Institute of Mechanical Engineering, Vol. 180, pp. 371-386.
- Tsai, L. W., and Joshi, S., 2002, "Kinematic Analysis of 3-DOF Position Mechanism for Use in Hybrid Kinematic Machines", ASME Journal of Mechanical Design, Vol. 124, No. 2, pp. 245-253.
- Tsai, L. W., 1999, "Robot Analysis: the mechanics of serial and parallel manipulators", John Wiley & Sons.
- Tsai, L. W., 1996, "Kinematics of a Three-DOF Platform Manipulator with Three Extensible Limbs", in Advances in Robot Kinematics, Edited by J. Lenarcic and V. Parenti-Castelli, Kluwer Academic Publishers, pp. 401-410.
- Tsai, L. W., Walsh, G. C. and Stamper, R., 1996, "Kinematics of a Novel Three DOF Translational Platform", IEEE International Conference on Robotics and Automation, Minneapolis, MN, pp. 3446-3451.

Vischer, P. and Clavel, R., 2000, "Argos: a Novel 3-DOF Parallel Wrist Mechanism", The International Journal of Robotics Research, Vol. 19, No. 1, pp. 5-11.