

CHAPTER ONE

INTRODUCTION

1.1. Background of the Study

The vision of manufacturers is to minimize human intervention in handling critical environmental applications with so much advancement and research works carried out across the world. This vision can be absolutely made into reality in the age of automation with the current technology (Shweta and Sanjay, 2012). Development in various fields of engineering have resulted in very sophisticated machines, devices and manufacturing processes. Successful operation of these devices and process require: very short response time, repetitious analytical and mechanical operations, low tolerance to errors, and insensitivity to disturbances that are well beyond human abilities. This is achieved majorly with automatic control systems in the form of robotic manipulators and full mobile robots. Panich (2015) defined robotic arm as a type of mechanical arm, which can usually be programmed with similar functions to a human arm. The reliance on the robots in the industries, military, etc has increased in recent times and most of these robots are engaged in remote and harsh environments to carry out a particular work for a long time. Robotic manipulators have also taken over in the marine industries with the use of Under Water Vehicles such as Remotely Operated Vehicles (ROV) used by most oil industries in their oil exploration works in the ocean and other deep waters. These robots are expected to work accurately without reduction in performance or breaking down in a short while. Therefore, the robots are supposed to be designed robustly to perform satisfactorily despite uncertainties in any environment.

Many research works have been carried out on the study of robotic manipulator development over the years. Building humanoid robots that can do useful things in the real world, not just research labs, is very difficult (Crowe, 2019). However, in recent years the attention of the researchers has been drawn towards improvement of the manipulators especially in performance and robustness due to the high rate of failure of commercial robots as recorded in (Carlson et al, 2004) and the high need for robots that can perform optimally despite disturbances in the system. The improvement in the robotic manipulator development can only be achieved by identifying the problems associated with the failures. Alassar (2010) identified two essential problems in the development of robotic manipulators: the first problem is the mathematical modeling of the manipulator and the actuators and the second problem is the control of the manipulator. Analyses in (Carlson et al, 2004) show that the control system is the most common source of robot failures followed by the mechanical platform which is attributed to the difference between the physical system and its mathematical model i.e. uncertainty. Hence, the modeling and control of manipulators have become a continuous research interest.

The problem of the dynamic model of the robotic manipulator was also identified in Fateh (2008) to be the joint torque control problem. Many advanced control strategies have been proposed to control robotic manipulators by controlling the joint torques. However, the inability of the commercial robots to control joint torques is a well known problem (An and Hollerbach, 1988; Schiller, 1996). The general robot arm dynamic control law proposed in (Spong, 1992; Chiu et al, 2004; Torres et al, 2007) was described by Fateh (2008) as complex and complicated. As a result, there are too many assumptions in the model and the control law cannot be applied to the inputs of the actuators in practice. Since the control law used

in the joint torque control of the robotic manipulator is supposed to be applied at the inputs of the actuator, therefore it becomes necessary to adopt the actuator dynamic control law to control the torque produced at the joints of the manipulator because it can easily be applied at the motor inputs. Actually, for a manipulator driven by DC motors, the currents of the DC motors are directly controlled to implement the torque control law (Miro and White, 2002; Reyes and Kelly, 2001). Most industrial robots are controlled by independent joint control strategy (Spong and Vidyasagar, 1989). In (Spong et al, 2006) independent joint control law was used for robotic manipulator model by considering the actuator dynamics and the arm dynamics. Most of the recent robotic manipulator control designs treat each joint of the manipulator as a simpler linear servomechanism with simple controller like an Independent Joint Control (IJC) (Helal et al, 2015).

Robotic manipulators are highly nonlinear dynamic systems with unmodelled dynamics and other uncertainties (Ren et al, 2007). This problem of uncertainties can be solved by the application of robust control to the plant model. The design of a robust control for the robotic manipulator involves the development of a dynamic model considering every non-negligible dynamics and designing a controller based on robust controller design specifications. Robustness has gained more and more attention (Fallahi et al, 2011) due to its importance in the realization of physical systems that can perform optimally and satisfactorily in real world. The goal of robust design is to retain assurances of system performance in spite of model inaccuracies and changes. A system is robust when the system has acceptable changes in performance due to model changes or inaccuracies (Dorf and Bishop, 2008). Piltan et al (2012) opined that disturbance rejection is used to test robustness.

Robust control for robotic manipulators is a typical control scheme to achieve good tracking performance in the presence of model uncertainties such as an unknown payload and unmodeled friction (Abdallah et al, 1991; Sage et al, 1999). Model uncertainties to be frequently encountered in manipulators working under an unstructured environment or handling variable payloads must be taken into account to solve the tracking problem of robotic manipulators. The estimated uncertainty bounds are often very conservative, leading to degradation of control performance due to unnecessarily high feedback gain selection (Kim et al, 2011). Several robust control strategies for robot manipulators have been introduced (Abdallah et al, 1991; Sage et al, 1999; Dawson et al, 1990; Su et al, 1993).

A Proportional-Integral-Derivative (PID) control scheme can eliminate the steady-state errors, but it can only ensure local asymptotic stability. Moreover, to guarantee the stability, the gain matrices must satisfy complicated inequalities (Arimoto and Miyazaki, 1984). Robust controller design requires both robustness against model uncertainty, as well as good disturbance and noise rejection properties and good performance. Considerable advancements in control system design led to the introduction of H-Infinity (H_∞) synthesis. This approach makes use of weights to achieve desired robustness and performance characteristics loop shape for the controller design. There are many advantages of this method such as high disturbance rejection, high stability and many more (Bansal, 2013). Today controller design based on robust H-Infinity control is the standard procedure for control systems where high demands on control quality exist. Han et al., (2011) applied H-Infinity in robust coordinated motion control of an underwater vehicle-manipulator system. Therefore, H-Infinity (H_∞) controller is applied in this work and compared with PID controller design method.

Many research works have been done on the design of controllers for robust control of robotic manipulators. From the review, the most common research gap in the robust controller design is the problem of robustness design specifications. Dorf and Bishop (2008) stated that a system is robust when it has low sensitivity, it is stable over the range of parameter variation and performance continues to meet the specification in the presence of a set of changes in the system parameters. Hence, robustness is the minimized sensitivity to effects that are not considered in the analysis and design phase for example, disturbances, measurement noise, and unmodeled dynamics.

The motivation for this work came from the interest for local content development and improvement. Most of the locally constructed automated systems especially the robots and other systems that are moved by electric motors were observed to perform poorly or fail completely within a short period of time after initial testing. This is as a result of poor design practice especially in the development of mathematical model or control law, wrong actuator selection and the effects of unresolved uncertainties.

1.2. Statement of Problem

The mathematical model of the manipulator has been one of the major problems in the development of robots. Every robotic manipulator consists of actuators and links; however from the review some of the manipulator dynamic models neglected either the actuator or link dynamics. This method increases the level of error in the system.

The robotic manipulator as a physical system contains uncertainties which can cause the system to function unsatisfactorily in practice or real environment. Therefore, the robust controller which is designed to address the effects of uncertainties should be based on robust design specifications

such as minimized sensitivity, and good disturbance rejection characteristics etc. However, the neglect of these robust controller design specifications leads to non-robust controller. A controller is robust only if it satisfies the robustness and performance characteristics. The characteristics are as follows: stability, disturbance rejection, noise suppression, minimized sensitivity, and minimized reference tracking error.

Actuator selection without proper torque calculation is a common critical problem affecting robotic manipulator design works. Different joints of a robotic manipulator carry different sizes of loads and experiences different torques. Hence, the actuators for the joints must be selected based on the total torques calculated at the respective joints. However, if the manipulator is designed and the joint actuators are selected without proper calculation of the total torques at the joints then the manipulator will not function properly even at no load when the torque generated at a joint goes beyond the torque produced by the actuator.

1.3. Aim and Objectives

The aim of this research is to develop a robust controller for an articulated 3DOF robotic manipulator under uncertainties based on H-Infinity Controller Technique compared with PID controller.

The objectives of the research are:

- a. To develop a torque control model based on independent joint control method.
- b. To develop a model error (or model uncertainty) based on the torque control model.
- c. To determine the total joint torques and select actuators based on the calculated torques.

- d. To determine the parameters of the joint torque model.
- e. To develop 3D model of the manipulator using Autodesk Inventor and generate an equivalent SimMechanics model in matlab.
- f. To develop a simulink model based on the joint dynamic model.
- g. To design a robust controller using H-Infinity synthesis and compare it with PID controller.
- h. To evaluate the robustness and performance characteristics of the controllers

1.4. Significance of the Study

The increased importance of the robot manipulators in the industries has made the system a vital topic of research. Furthermore, the fact that most robot manipulator systems work independently or remotely monitored, made it possible to locate them in unfriendly environments especially sometimes where human beings would find it difficult to work. There has been increased reliance on the capabilities of robotic manipulator in the industries which made it necessary to develop robots that can carry out their duties satisfactorily despite disturbances in the system. Therefore, the robotic manipulator should be designed to achieve its set goals despite the level of faulty conditions or environmental aggressions it may be facing. This hearty capability of the system depends on the joint torque control ability and robustness of the controller. Hence, the need for the development of a robust control for robotic manipulators becomes very significant.

1.5. Scope of the Study

The robotic manipulator comprises of links connected together by joints; the joint torque is produced by the actuator and controlled by the controller. This research work covers the following: modeling of the robotic manipulator dynamics, actuator selection, joint model parameter derivations and robust

controller design. The decentralized control method was introduced to enable the formulation of joint torque control dynamic model that can be applied to the joint actuators. This model involves actuator dynamics and arm dynamics. The 3DOF articulated robotic arm 3D structure was achieved using AutoDesk Inventor professional and converted into SimMechanics model. However, the SimMechanics model of the manipulator does not provide means for joint actuator design. The joint dynamic model was implemented in SIMULINK for the simulation.

CHAPTER TWO

LITERATURE REVIEW

2.1 Related Literature

According to Melchiorri (1995), the dynamic model obtained using the Lagrange-Euler method is simpler, more intuitive and more suitable to understand the effects of changes in the mechanical parameters. The links are considered altogether, and the model is obtained analytically. He however pointed out that the drawback of the model is that it is obtained from the kinetic and potential energies (non intuitive); the model is not computationally efficient. He continued to state that the Newton Euler method is based on a computationally efficient recursive technique that exploits the serial structure of an industrial manipulator. On the other hand, its mathematical model is not expressed in closed form. Although, the two techniques are equivalent (provide the same results). Iqbal and Author (1995) suggested that there should be an improvement in the dynamics model of the robotic manipulator by adding the complete model of the selected drives (actuators) in the model. Melchiorri declared that the actuation system has several effects on the dynamics: if motors are installed on the links, then masses and inertia are changed, and it introduces its own dynamics (electromechanical, pneumatic, hydraulic, etc) that may be non-negligible. It also introduces additional nonlinear effects such as backlash, friction, and elasticity.

Lui (2011) presented a research work which was focused only on the design of a 3 DOF robotic manipulator with a 2 DOF dexterous wrist mechanism. From the review, this manipulator configuration provides precise, dexterous and flexible movement of the manipulator. The dynamic model applied in (Liu and Peng, 2000; Wang et al, 2008, Lochan and Roy, 2015; Lui et al, 2016) is basically the dynamical description of the mechanical arm of the manipulator. Kim and Lee (2011) proposed a robust control model of robotic

manipulators under parametric uncertainty using only robot link dynamic model based on the Lagrange-Euler equation of motion of robot links which comprises of inertia matrix, centripetal and coriolis vector and the gravitational vector of the link. This method of robot dynamic model was used in many research works but recently, it has been criticized in Fateh (2008), due to its limitations in feedback application and drawback in its application to the actuator inputs. The only parameter in this model that can be fed back is the position q , which is not sufficient for robust control where signal disturbances and dynamic perturbations such as mistakes in choosing the coefficients, unmodeled dynamics, friction, wear and won etc. are involved. The actuator current which determines the desired force produced by the joint was not considered in this model. Since the ability of the robotic arm to maintain precise and desired position even in the presence of uncertainty depends on the force produced by the joint and this force depends on the actuator current, therefore, the robust control law should involve current i and position q . Secondly, the friction vector was not included in their dynamic model, however, non-parametric uncertainties cannot be avoided in robotic manipulators. Biradar et al (2012) investigated Lagrange-Euler method and suggested a future work for an improved model that can be implemented in the controller of the manipulator, and optimized for a specific job task.

In zadbakhsh et al., (2007), the Lagrange model was used when considering the equation of motion of robot links (Eqn. 3.36). However, a complete model of the actuator was used for the robust controller design simulation in their work; they did not consider the arm in the model used in the simulation. Secondly, total torques at the joints was not calculated for proper joint actuator selection. Instead they arbitrarily chose big actuators for the first major joints and small actuators for the last minor joints.

Lewis et al (2004) stated that to obtain a complete dynamical description of the arm plus the actuator (which make up the robotic manipulator), it is required to add the actuator dynamics to the arm dynamics. This is because, the torque produced by the robotic arm is controlled by the voltage applied to the motor armature which represents the control input to the system. Therefore, neglecting the actuator dynamics as unmodeled parameters makes the dynamic model of the robotic manipulator incomplete.

Talole et al proposed a mathematical model of a single rigid link manipulator based on the link (or arm) dynamics plus the actuator dynamics. In their model, they considered the link plus actuator inertias as J , and the actuator damping B . In (Dong and Zhu, 2013), the dynamic model of a single rigid link flexible joint robotic manipulator was presented as:

$$\begin{cases} J\ddot{q} + J_m\ddot{\theta} = \tau \\ J\ddot{q} + k(q - \theta) = 0 \end{cases}$$

Where J is link inertia, J_m is actuator inertia, k is the stiffness of the flexible joint, q is the link angle, and θ is the actuator angle. However, in their work, the system damping and friction were not considered. Other parameters were considered as unmodeled dynamics. In this model the independent joint control law approach was adopted. The distinguishing factor in this model of the flexible joint robotic manipulator is the stiffness constant k .

According to Helal et al (2015), actuators model are computed to merge it with the dynamic model of the robot as:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} k_{m1} \\ k_{m2} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} - \begin{bmatrix} B_{m1}n^2 \\ B_{m2}n^2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} - \begin{bmatrix} J_{m1}n^2 \\ J_{m2}n^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{R_1} \\ \frac{1}{R_2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} - \begin{bmatrix} \frac{L_{a1}}{R_1} \\ \frac{L_{a2}}{R_2} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} - \begin{bmatrix} k_{e1}n \\ k_{e2}n \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

where I is motor current, k_m is motor constant, B_m is friction/damping constant including links, J_m is motor inertia, k_e is back-emf constant, R is motor resistance, L_a is motor inductance, n is the gear box ratio, V is the input voltage to the motors. Conversely, the dynamic model they presented did not include the inertia of the robot link. Thus, the consideration of the manipulator link inertia into this model will give a more complete dynamical description of the robotic manipulator. In addressing the problems in robot force control as presented in (Eppinger and Seering, 1992), the actuator model is coupled to the rigid body model of the robotic manipulator. The effective inertia (total moving mass) of the axis and effective viscous damping to the ground were considered in the model.

Agrawal et al (2012) modeled a controller for an articulated robotic arm.

$$\frac{\theta_L(s)}{V_a(s)} = \frac{nK_a}{s(sR_aJ_{eff} + R_af_{eff} + K_aK_b)}$$

Where $\theta_L(s)$ is the link position, $V_a(s)$ is the input voltage into the actuator, f_{eff} is the effective viscous friction coefficient, n is gear reduction ratio.

They identified precise control upon each degree of freedom of a robotic arm as a great challenge in the implementation of industrial works. Hence, their modeling approach was based on the actuator dynamics instead of the Lagrange model of robotic arm dynamics. Fateh (2008) presented a research work titled "On the Voltage-Based Control of Robotic Manipulators", where he used the independent joint control which is based on the joint actuator dynamic model and the torque due to link, to model and control a robotic manipulator. According to him, using this method obtains simplicity, accuracy, speed of calculation and robustness to the manipulator control system. Alassar (2010) carried out a research on the modeling and control of 5DOF robot arm using supervisory control. He adopted the independent joint control (IJC) technique where the joint dynamic model relating the current and the developed torque in the motor shaft were considered instead

of using only the Lagrangian-Euler (inertia matrix, coriolis and gravitational vectors) of robot arm dynamic model.

$$G_{position} = \frac{\theta(s)}{V(s)} = \frac{K_t}{s[J_m l_a s^2 + (L_a B_m + J_m R_a)s + K_t K_b]}$$

The motor dynamic model enabled the achievement of precise results in the performance of each robot joint. The actuator selection should be determined by the calculated torque at each joint. However, in his work, the torque at each joint was not calculated; therefore the motors were not selected appropriately. Secondly, the torques at the joints of the 5DOF robotic arm cannot be the same; hence using the same motor parameters to simulate the 5DOF joints should be faulty during physical implementation of his design. Thirdly, the motor derived parameters such as the torque constant and the damping coefficient etc were not derived based on the torque at the joint and the current specification of the motor. Ovy et al (2011) designed an articulated robot arm for precise positioning using joint dynamic model instead of the Lagrangian-Euler robot dynamic model.

$$\frac{\omega(s)}{V(s)} = \frac{K_t}{(Js + D)(Ls + R) + K_e K_t} = G(s)$$

Where D is the viscous frictional coefficient, $\omega(s)$ is the speed.

The achievement of high precision capability of the robotic arm was attributed to the joint dynamic modeling method. However, in their work, they did not calculate the torques at the joint before choosing the motors for the joint design and this could cause poor system performance or total failure when implemented physically. The controller design for the robotic manipulator in (Lui, 2011) was based on joint actuation (i.e., the joint motors) which was carried out independently.

In the robust control methods, the controller is designed based on the nominal model plus uncertainty. Uncertainty can be in any parameter, such as the load carrying by the end effector (Alashqar, 2007). Many researchers have proposed and developed many methods of achieving a robust

controller. The major objective of the robust controller design is to obtain controller gains that can achieve the desired output trajectory in the presence of significant uncertainties. Ahuja and Tandon (2013) presented a robust PID and Polynomial controllers for DC motor speed control. The uncertainty caused by the parameter changes of motor resistance, motor inductance and load are formulated in their work as multiplicative uncertainty weight, which are used in the objective function in the design.

Kim and Lee (2011) applied dynamic compensation which is similar to lag compensation. According to Salem (2013), lag compensation is a soft approximation of Proportional-Integral (PI) controller whereby large gain values introduce or increase overshoot and transient oscillation. System sensitivity is a vital measure of robustness of a control system however; the results in Kim and Lee (2011) did not show any sensitivity reduction or stability robustness characteristics.

According to Dorf and Bishop (2008), the design of a control system in the presence of significant uncertainty requires the designer to seek a robust system. They developed many robust controllers for different physical systems using closed loop PID controller method. In recent advances, robust control design methodologies can address stability robustness and performance robustness in the presence of uncertainty. The derivative gain function of the controller solves the problem of rate of change in error thereby maintaining stability and performance robustness. Ackermann et al presented the design of robust PID controllers for improved performance of a robustly decoupled car steering control system.

Bansal and Sharma (2013) applied H_∞ controller design method in their work. They stated that H_∞ control synthesis is found to guarantee robustness and

good performance and also provides high disturbance rejection. Baslamish (2007) applied H_∞ controller in Linear Parameter Varying (LPV) Modeling and Robust Control of Yaw and Roll Modes of Road Vehicles. Yadav and Singh (2015) carried out a design on the robust control of two link rigid manipulator. In their work, H-Infinity controller design method was applied and it achieved good system performance and robustness. However, the controller results showed high system overshoot and this could be as a result of the system modeling method used in their work and the joint torque calculations. Wang et al (2008) carried out a research work on robust tracking control of robotic manipulator using dissipativity theory based on H_∞ controller technique. It was confirmed in their work that the scheme improved the robustness of the system. Sato et al (2008) presented a work on adaptive H_∞ control for robotic manipulator with compensation of input torque uncertainty; which examines the problem of link position tracking control for robot manipulators with input torque uncertainty. They stated that, an approximated error of input nonlinearities and external disturbances are attenuated by means of H_∞ control performance. Yim and Park (1999) in their work titled "Nonlinear H-infinity Control of Robotic Manipulator", designed a nonlinear *H-infinity* controller for a two degree-of-freedom planar robot manipulator with uncertainty in its mass. However, the source of uncertainties cannot be only in the mass of the plant as a physical system in real environment.

However, the H-infinity controller design method applied in these works for the compensation of input torque uncertainty was based only on the reference tracking performance of the manipulator arm angle, which is not enough point to ascertain the ability of system to address the problem of uncertainty.

2.2 Robotic Manipulator

A robot is a reprogrammable multifunctional machine designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks. Gupta et al (2013) defined robotic arm as a mechanical arm which is designed to perform a function similarly as a human arm does. A robotic manipulator is usually programmable and the arm may be a sum total of the mechanism or can be part of a complex robot. Manipulators are composed of links (or arms) connected by joints into a kinematic chain. The force that moves the links is produced by the actuator which is housed in the joint mechanism. Joints are typically rotary (revolute) or linear (prismatic). A revolute joint is like a hinge and allows relative rotation between two links. A prismatic joint allows a linear relative motion between two links. The convention (R) is used for representing revolute joints and (P) for prismatic joints. Figure 2.1 shows the symbolic representation of robot joints in two and three dimensional drawings.

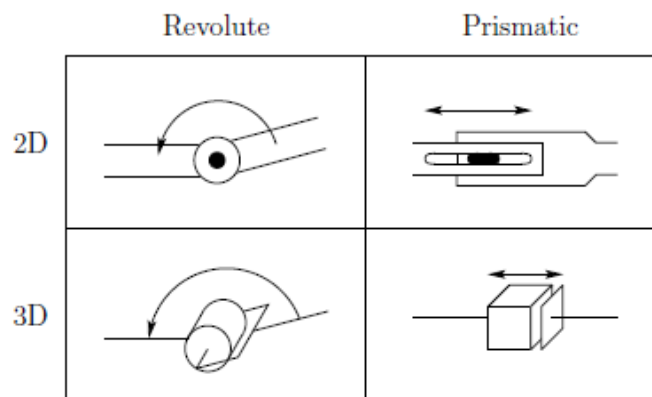


Figure 2.1: Symbolic representation of robot joints (Spong et al, 2006)

Each joint represents the interconnection between two links, say l_i and l_{i+1} . The axis of rotation of a revolute joint, or the axis along which a prismatic

joint slides is denoted by z_i if the joint is the interconnection of links l_i and l_{i+1} . The joint variables, denoted by θ_i for a revolute joint and d_i for the prismatic joint, represent the relative displacement between adjacent links.

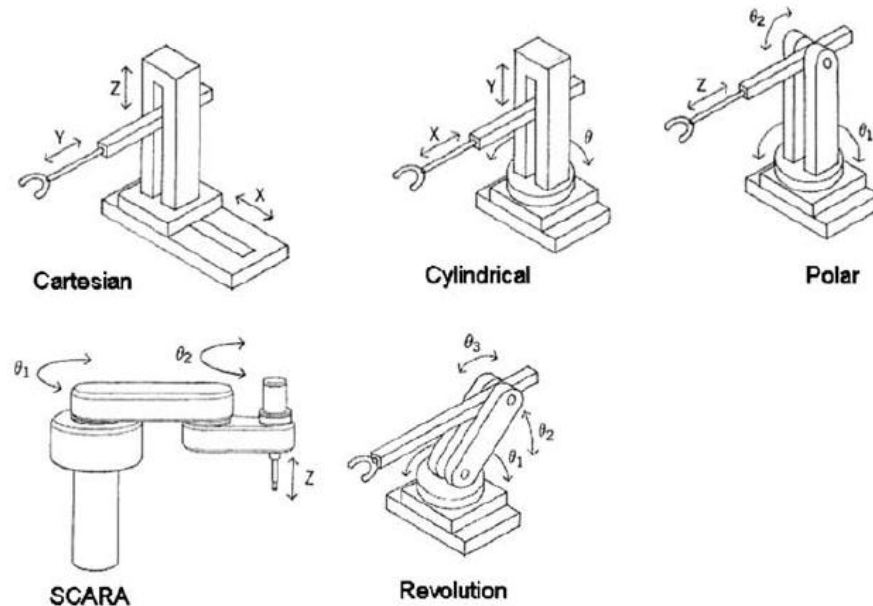


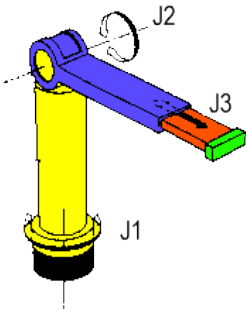
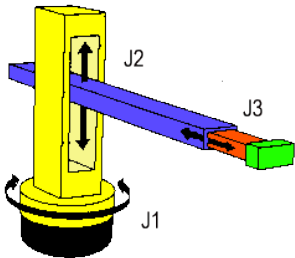
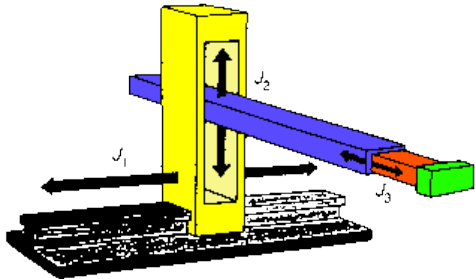
Figure 2.2: Types of arms used with actual robot manipulators (Pires, 2007)

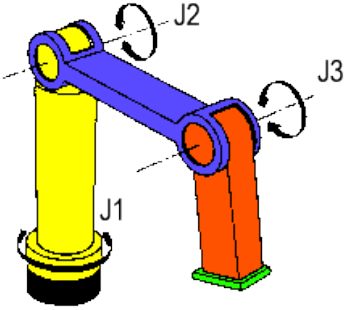
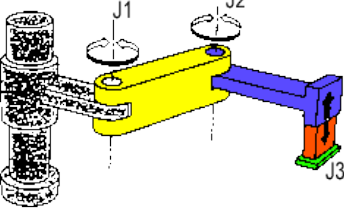
2.3. Robotic Manipulator Kinematic Arrangements

The industrial robots are basically composed of rigid links, connected in series by joints, having one end fixed (base) and another free to move and perform useful work when properly tooled (end-effector). The structure of the robot consists of a number of links and joints, a joint will allow relative motion between two links (Crowder, 1998). The different arrangements of the rigid links and the type of joints applied in the design of a robotic arm gave rise to the types of arms of the manipulator. There are five types of arms commonly used by actual industrial robot manipulators as shown in figure 2.2: *cartesian*, *cylindrical*, *polar*, *SCARA* and *revolution*. Although there are many possible ways prismatic and revolute joints are used to construct

kinematic chains, in practice only a few of these are commonly used (Spong et al, 2006). Crowder (1998) stated that the basic robot arm has three joints, this allows the tool at the end of the arm to be positioned anywhere in the robots working envelope. Even though there are a large number of robot configurations that are possible, only five configurations are commonly used in industrial robotics as summarized in table 2.1.

Table 2.1: Configurations commonly used in industrial robots (Crowder, 1998)

Polar		The linear extending arm is capable of being rotated around the horizontal and vertical axes.
Cylindrical		The linear extending arm can be moved vertically up and down around a rotating column.
Cartesian and Gantry:		Three orthogonal sliding or prismatic joints.

Jointed Arm or Articulated		Three joints arranged in an anthropomorphic configuration.
Selective Compliance Assembly Robotic Arm, SCARA		Two rotary axes and a linear joint.

The robotic manipulator arrangements that are most typical in the industries are briefly described as follows.

2.3.1. Articulated manipulator (RRR)

This type of robotic manipulators comprises of revolute joints and the linkages. Articulated robotic manipulators are very useful in the industries due to their small cross section and projected ability to change elevation and maneuver over obstacle (Pachaiyappan et al, 2014). The small cross section and the loads associated with suspension of the Robot while changing elevation or maneuvering over obstacles require large joint torque to weight ratios for joint actuation. The articulated manipulator shown in figure 2.3 is also called a revolute or anthropomorphic manipulator. This kind of manipulator is known as an elbow manipulator. The structure and terminology associated with the elbow manipulator are shown in Figure 2.5. Another common revolute joint design is the parallelogram linkage shown in Figure 2.4. The revolute manipulator provides for relatively large freedom of movement in a compact space. The parallelogram linkage is typically less dexterous or agile than the elbow manipulator.



Figure 2.3: The articulated revolute manipulator (Spong et al, 2006)



Figure 2.4: The parallel revolute manipulator (Spong et al, 2006)

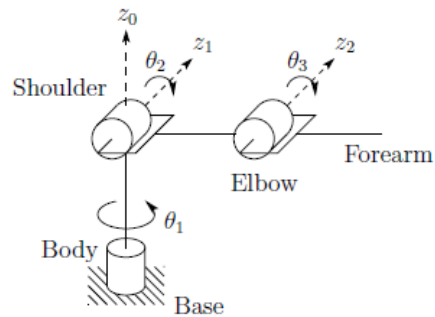


Figure 2.5: Structure of the elbow manipulator (Spong et al, 2006)

2.3.2 Spherical Manipulator (RRP)

The spherical (polar) manipulator was obtained by replacing the third or elbow joint in the revolute manipulator by a prismatic joint, as shown in figure 2.6. The term spherical manipulator is derived from the fact that the spherical coordinates defining the position of the end-effector with respect to a frame whose origin lies at the intersection of the three z axes are the same as the first three joint variables.

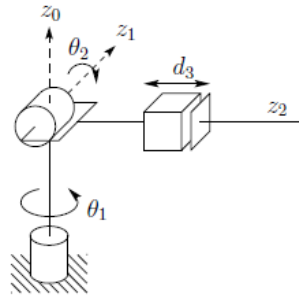


Figure 2.6: The spherical manipulator (Spong et al, 2006)

2.3.3 SCARA Manipulator (RRP)

The SCARA (Selective Compliant Articulated Robot for Assembly) arm shown in Figure 2.7 has an RRP structure but it is quite different from the spherical manipulator in both appearance and in its applications. Unlike the spherical design, which has z_0 perpendicular to z_1 , and z_1 perpendicular to z_2 , the SCARA has z_0 , z_1 , and z_2 mutually parallel.

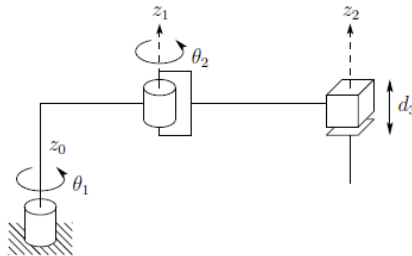


Figure 2.7: The SCARA (Selective Compliant Articulated Robot for Assembly) (Spong et al, 2006)

2.3.4 Cylindrical Manipulator (RPP)

In the cylindrical manipulator is shown in Figure 2.8, the first joint is revolute and produces a rotation about the base, while the second and third joints are prismatic. As the name suggests, the joint variables are the cylindrical coordinates of the end-effector with respect to the base.

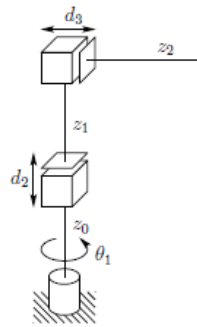


Figure 2.8: The cylindrical manipulator (Spong et al, 2006)

2.3.5 Cartesian manipulator (PPP)

A manipulator whose first three joints are prismatic is referred to as a Cartesian manipulator, shown in figure 2.9. For the Cartesian manipulator, the joint variables are the Cartesian coordinates of the end-effector with respect to the base. Cartesian manipulators are useful for table-top assembly applications and, as gantry robots, for transfer of material or cargo.

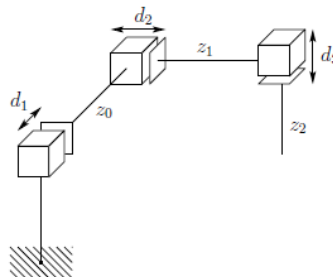


Figure 2.9: The Cartesian manipulator (Spong et al, 2006)

2.4 Robotic Manipulator Kinematics

The kinematics model in (Banga et al, 2011; Iqbal et al, 2005; Elfasakhany et al, 2011; Ghuffar et al, 2006; Gupta et al, 2013; Spong et al, 2006) is useful in the trajectory planning of the robotic manipulator movements within its work envelop. With the assumption made by Spong et al (2006) that each joint has a single degree-of-freedom, the action of each joint can be described by a single real number; the angle of rotation in the case of a revolute joint or the displacement in the case of a prismatic joint. Kucuk et al (2006) opined that kinematics studies the motion of bodies without consideration of the forces or moments that cause the motion. Robot kinematics refers the analytical study of the motion of a robot manipulator. Formulating the suitable kinematics models for a robot mechanism is very crucial for analyzing the behaviour of industrial manipulators. The transformation between two Cartesian coordinate systems can be decomposed into a rotation and a translation. The robot kinematics can be divided into forward kinematics and inverse kinematics.

2.4.1 Forward Kinematics

A manipulator is composed of serial links which are affixed to each other revolute or prismatic joints from the base frame through the end-effector. Al-Tabey (2012) stated that the forward kinematics problem is to determine the position and orientation of the end effector, given the values for the joint variables of the robot. The objective of forward kinematic analysis is to determine the cumulative effect of the entire set of joint variables, that is, to determine the position and orientation of the end effector given the values of these joint variables. The joint variables are the angles between the links in the case of revolute or rotational joints, and the link extension in the case of prismatic or sliding joints. Hence, calculating the position and orientation of the end-effector in terms of the joint variables is known as forward

kinematics(Kucuk et al, 2006). It is customary to establish a fixed coordinate system, called the world or base frame to which all objects including the manipulator are referenced.

2.4.2 Inverse Kinematics

The objective of inverse kinematic analysis is, in contrast, to determine the values for these joint variables given the position and orientation of the end effector frame.Kucuk and Bingul, (2006) stated that solving the inverse kinematics is computationally expansive and generally takes a very long time in the real time control of manipulators. Tasks to be performed by a manipulator are in the Cartesian space, whereas actuators work in joint space.Cartesian space includes orientation matrix and position vector. However, joint space is represented by joint angles. The conversion of the position and orientation of a manipulator end-effector from Cartesian space to joint space is called as inverse kinematics problem.In order to command the robot to move from one location to another,the inverse kinematics is needed. This is the problem of inverse kinematics.

2.4.3 Robotic Manipulator Dynamics

Dynamics of the robotic manipulator studies the motion of bodies with the consideration of the forces or moments that cause the motion. The dynamic equation of the manipulator describes the relationship between torque and motion. In robotics, the link dynamic model is obtained starting from well known physical laws like the Newtonian mechanics and the Lagrange mechanics (Goldstein, 1980; Symon, 1987). Different robot arm dynamic models have been developed:Lagrange-Euler,Newton-Euler, D'Alembert(Craig J.J, 1989; Sciavicco and Siciliano, 1996; Fu et al, 1989; De Wit, 1996). Nevertheless, they are equivalent to each other because they define the same physical phenomenon, i.e., the dynamics of rigid bodies assembled together to constitute a robot. Obviously, the structure of the

motion equations is much different because each formulation was developed to achieve different objectives such as computation efficiency, simplicity to analyze and/or to simulate the structure, etc.

The development of a dynamic model for the torque control of a robotic manipulator continues to attract more research attention. The Lagrange-Euler model applied in (Asada, 1998; Spong et al, 2006; Yadav and Singh, 2015) considered only the dynamics of the rigid bodies (arm) which comprises of the mass, length, inertia, angular position, velocity and acceleration of the links without considering the joint actuator dynamics. On the other hand, In (Izadbakhsh and Fateh, 2007; Bellicoso et al, 2013; Jaiswal and Kumar, 2014; Zhu and Fan, 2012; Spong et al, 2006; Fateh, 2008), the joint actuator dynamics was considered in the development of dynamic model for joint torque control of the robotic manipulator. Actually, since the torque needed to move the rigid bodies are generated by the actuators and the torque control law must be applied to the actuator as input voltage, therefore neglecting the actuator dynamics in the joint torque control dynamic model makes the model to be incomplete and difficult to be applied. The robot link dynamics is very important as well as the joint actuator dynamics in the dynamic model of the robotic manipulator. Hence, the combination of the arm dynamics and the actuator dynamics yields a complete dynamic model for joint torque control of the manipulator as suggested in (Lewis et al, 2004). The independent joint control strategy in (Assada, 2010; Lewis et al, 2004; Spong et al, 2006) helps to apply this model for better precision and efficiency. This model is more preferred for the robust control of the manipulator since it involves a better dynamical description of the system.

However, despite the improvements in the mathematical models of the manipulator, there exist different ranges of problems in the form of

uncertainties in the models caused by frictional forces, payloads, and the mistakes in the parameters of the model or unmodelled parameters etc.

2.5 Manipulator JointActuator

Pachaiyappan et al., (2014) described robot actuator as the mechanism that provides the necessary forces to move the mechanical structure. Muhammad (2013) opined that, in order to control the parameters of industrial processes, there are various kinds of actuators on field. Actuator is a driver that runs some mechanical activity. For example, if a process needs to open a valve for fluid motion or move a robotic arm for some appropriate action, there will be a motor with specific applied controls such as the speed and angular position control. The proper selection of actuator will dictate how effective a robot can perform a specific task. Actuators can be either mechanical or electrical and have varying strengths and weaknesses as demonstrated in table 2.2. The basic actuators used for controlling motion include: air, hydraulic, clutch/brake, stepper and servo motors. The mathematical models are as follows:

Air Motors (Luo et al, 2008):

$$V_a(0) = \frac{1}{2}L(B^2 - r^2)\pi + V_n \quad V_b(0) = \frac{1}{2}L(B^2 - r^2)\pi - V_n$$

Where V is volume of chambers, r is rotor radius, L is vane active length.

Hydraulic Motors (Madanipour, 2014):

$$D_m(P_{lm} - P_{rm}) = T_{friction} + T_{Load} + J \frac{dw_m}{dt}$$

Where D_m is motor displacement, P is pressure, T is torque, J is inertia, w is speed.

Clutch/Brake (Quang, 1998):

$$T_c = T_m + \dot{\omega}_m \theta_m$$

Where T_c is torque of clutch, T_m is torque of driven machine, θ_m is inertia of driven machine, w_m is angular speed of driven machine.

Stepper and servo motors use same mathematical model and the model is stated and applied in chapter three.

Table 2.2: Comparison of Actuators (Pachaiyappan et al, 2014)

Actuator Types	Strength	Weakness
Air Motor	Low Cost, Easily Maintained, Simple to Operate	Audible Compressor Noise, Inefficient System, Difficult to Regulate Speed
Hydraulic Motor	High loads possible, Simple to operate	Slow System, Inefficient System, High maintenance Requirement
Clutch/ Brake	Low Cost, Effective for Light Load, Easy to Perform System Matching	Uncontrolled Acceleration, Components Prone to Wear, Non-repeatable System
Stepper Motor	Simple Control, Constant Load, Accurate Position	Cannot Vary Load, Can Lose Steps, Resonance Problem
Servo Motor	High Performance, Small motor Size, Can Operate at High Speed	High Cost System, Performance Limited by Controls, Speed Limited by Electronics

According to Pachaiyappan et al (2014), Steppers can be grouped into three categories that differ in terms of internal construction based on the use of permanent magnets and/or iron rotors with laminated steel stators: Permanent magnet, Variable reluctance, Hybrid. The term "servomotor" does not refer to one single kind of motor. Instead it refers to any type of motor that receives a command signal from a controller. In this same respect, any closed loop system can be referred to as a servo system.

According to Farhan (2013), motion control is a sub-field of control engineering, in which the position or velocity of a given machine are controlled using some type of actuating device. Most used actuating devices

in mechatronics applications are electric actuating machines (DC motors), which are used in many, if not most, modern machines (e.g. electric cars, locomotives, fans, turbines, and drills), robotics (e.g. Mobile robot and robot arm). Two main motion control applications are of concern namely, mobile robots and robotic arms. The motion of robotic manipulators are described in Degrees of Freedom (DOF) which is the number of possible movements a robotic arm is capable of completing. Shweta and Sanjay, (2012) designed an articulated 5-DOF robotic arm by direct and inverse kinematic analysis methods which is capable of completing five degrees of freedom in different directional movements. The possibility of the number of DOF depends on the number of the actuators or motors used to complete the possible number of different movements. In the 5DOF (five functions) robotic arm, each movement or DOF has a DC motor (Kumar and Raja, 2014) and a controller attached to it.

Direct Current (DC) motors are often used in various industrial applications where a wide range of responses are required to follow a predetermined trajectory of speed or position under variable load (Faramarzi&Sabahi, 2011).According to Fateh (2013), single joint robot arm system consists of three parts; arm, connected to actuator through gear train with gear ratio, n . The DC motor is an example of electromechanical systems with electrical and mechanical components, a simplified equivalent representation of DC motor's two components are shown in figure 2.10a. DC motor turns electrical energy into mechanical energy and produces the torque required to move the robotic bodies or linkage to the desired angular position, θ , or rotate with the desired angular speed, ω .

The objective in the development of the mathematical model for joint torque control is to relate the voltage applied to the armature of the actuator to the position of the link. In the same vein, the objective in the development of

the mathematical model for the actuator is to relate the voltage applied to the armature of the actuator to the position of the shaft. Two balance equations can be developed by considering the electrical and mechanical characteristics of the system (Atlas, 2007) for the formulation of the joint actuator dynamic model. The torque produced by the joint is equal to the product of torque constant and actuator current.

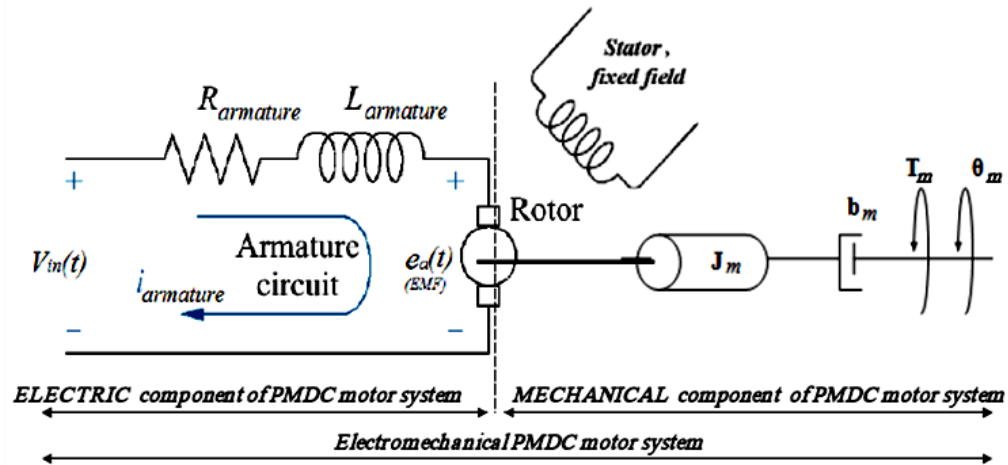


Figure 2.10a: A simplified equivalent representation of the PMDC motor's electromechanical components (Ahmad et al, 2013; Salem, 2013)

The joints of the manipulator experience different amount of torques depending on their locations in the structure of the manipulator. Joint actuators are designed and manufactured in different sizes and with different torque ratings and other parameters such as voltage, current, inertia etc. Since the actuator generates the force that moves the link, then it must produce a force greater or equal to the force produced due to link and possible payload. Therefore, the selection of the joint actuators becomes very vital in the development of a robust manipulator.

2.5.1 Joint Actuator (Motor) Selection

Every possible movement of the robotic manipulator is determined by the joints, and the performance of the joint is influenced by the motor torque

and the total torque generated at that joint due to load. Thus, it becomes necessary to design the joint properly for adequate joint torque control. The design of the joint involves calculating the torque due to the load connected at every joint, selecting the motors based on the calculated load torques at the respective joints and deriving the parameters that are not given based on the given parameters of the selected motors. This method of joint design was carried out in Kafuko et al (2015).

In the steps for actuator selection, first it is necessary to determine the maximum torque required for each joint motor. The torques are calculated by estimating the weight the motor shaft would have to be holding and multiplying it by the distance from the center of gravity of the weight back to the motor shaft. The weight included motors farther up the arm, the weight of the link arm etc. The calculated maximum torques for each joint motor is as shown in Table 2.3 (Emerich, 2007).

The second step is choosing joint motors based on the calculated torque values at the respective joints, so a solid model of the design could be made in Autodesk Inventor. The motors chosen by Emerich(2007) are high torque stepper motors from Anaheim Automation. The specific motor model numbers are shown in Table 2.3. Stepper motors output good amount of torque as well as high precision. These motors are small and light weight which is crucial in keeping the torque lower for the motors father back in the arm.

Table 2.3: Torque requirements and motor selections (Emerich, 2007)

Joint	Maximum Required Torque (oz.-in.)	Motor Selected (Anaheim Automation Model #)	Motor Maximum Torque Rating (oz.-in)
1	4315	42N3	4365

2	983	34K2	1535
3	109	23L1	126
4	83	17Y4	100
5	59	17L1	61
6	19	14Y1	25
End-Effector	10.7	GM12	14

2.6 Robotic Manipulator Control Schemes

Milchiorri(2006) stated that the robot performances are mainly influenced by the mechanical design and by the actuation system. He also explained that there are two types of robot control schemes: Decentralized (or independent) control schemes and Centralized control schemes. The decentralized control scheme also known as independent joint control (IJC) model has Single Input Single Output (SISO) configuration while the centralized control scheme has a Multiple Input Multiple Output (MIMO) configuration. From the review, SISO configuration (which can equally be in the form of multiple SISO) is more common and simpler than the MIMO in practice. According to Alassar (2010), the basis of IJC is that the robotic manipulator is treated as a set of **independent actuators** working independently. This means that each link of the robotic manipulator is considered as single input single output (SISO) system with an independent controller.

Independent joint control scheme is widely adopted in most industrial manipulator controller because of its simplicity (Voung and Ang, 2009). According to Richter, to derive the independent joint control model, it is assumed that the DC motor is connected to a gear reduction of ratio $r : 1$ and moment of inertia J_g . The reduced-speed shaft drives a rotational inertia J_l , which represents the link driven by the motorized joint. The motion of the

other links should influence the DC motor as well. For the independent joint model, however, the influences of the other links are treated as disturbances and then the controller is designed to be robust (tolerant) against them (Richter). When applying the decentralized control scheme, Melchiorri (2006) stated that each joint of the robotic manipulator is considered independently, and the term $d (= \tau_l/r)$ is considered as an external disturbance. These considerations can be applied with proficiency when there is no direct coupling between the actuator and the joint (i.e., when there are gears).

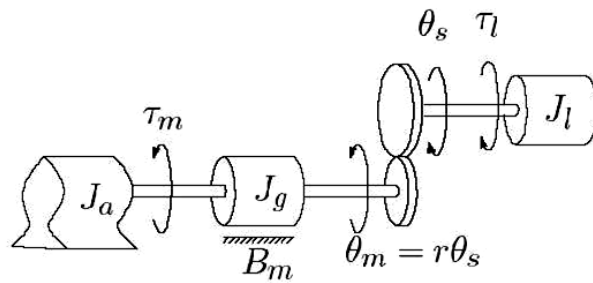


Figure 2.10b: The lumped model of a single link with actuator/gear train (Melchiorri, 2006; Spong, 2006; Bozma, 2015)

Figure 2.10b illustrates the independent control scheme where the actuator and link dynamics are considered. Where J_a , J_g and J_l are respectively, the actuator, gear, and load inertias. B_m is the coefficient of motor friction and includes the friction in the brushes and gears. τ_l is the link torque, r is the gear ratio. Considering a DC motor actuator, the independent joint model of the robotic manipulator is derived in (Melchiorri, 2006; Spong, 2006; Bozma, 2015). Figure 2.10c shows the block diagram of a robotic manipulator. Figure 2.10d represents a third order system from input voltage $V(s)$ to output position θ_m .

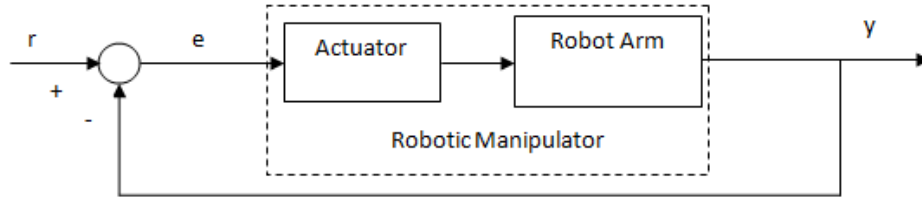


Figure 2.10c: Robotic manipulator block diagram (SISO)

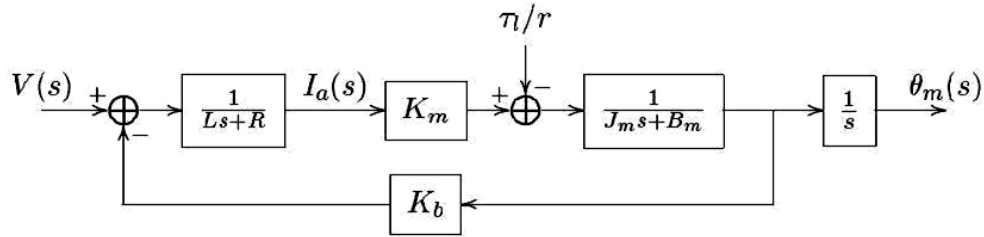


Figure 2.10d: Block diagram of actuator model with link (Melchiorri, 2006; Spong et al, 2006; Bozma, 2015)

2.7 Robotic Manipulator System Uncertainties

The robotic manipulator is a physical system which is modeled mathematically in many research works for control purpose and to enhance the performance of the system. Garulli et al stated that when modeling physical systems for control purposes, it is necessary to provide model descriptions that capture the main features of the system behavior and are mathematically tractable at the same time. An extremely accurate model of a physical process may turn out to be unsuitable for application of the available analysis and design techniques. By contrast, an over-simplified model, which misses significant information on the system structure, may lead to unacceptable design performance. They opined that the discrepancy between the system and the adopted nominal model is usually represented as a perturbation on the nominal model. The resulting model, which is therefore composed of the nominal one and the perturbation, is usually referred to as the *uncertain model* or *model set*.

Uncertainties are unavoidable in physical systems and they are classified into two categories: disturbance signals and dynamic perturbations. The former includes input and output disturbance, sensor noise and actuator noise, *etc.* The latter represents the discrepancy between the mathematical model and the actual dynamics of the system in operation. A mathematical model of any real system is always just an approximation of the true, physical reality of the system dynamics. Typical sources of the discrepancy include unmodelled (usually high-frequency) dynamics, neglected nonlinearities in the modeling, effects of deliberate reduced-order models, and system-parameter variations due to environmental changes and torn-and-worn factors. These modeling errors may adversely affect the stability and performance of a control system. The consideration of the uncertainties in the design of a control system gave rise to the development of the robust control. Dynamic perturbations are usually described so that they can be well considered in system robustness analysis and design.

Scherer (2001) stated that all mathematical models of a physical system suffer from inaccuracies that result from non-exact measurements or from the general inability to capture all phenomena that are involved in the dynamics of the considered system. Even if it is possible to accurately model a system, the resulting descriptions are often too complex to allow for a subsequent analysis, not to speak of the design of a controller. Hence one rather chooses for a simple model and takes a certain error between the simplified and the more complex model into account. Therefore, there is always a mismatch between the model and the system to be investigated. In control engineering this mismatch is referred to as *uncertainty*. The uncertainties in a physical system model can be classified into disturbance

signals and dynamic perturbation. However, the dynamic perturbation is used to describe the mismatch between the model and the system.

According to Gu et al (2005), many dynamic perturbations that may occur in different parts of a system can, however, be lumped into one single perturbation block Δ , for instance, some unmodelled, high-frequency dynamics which is referred to as “unstructured” uncertainty. The unstructured dynamics uncertainty in a control system can be described in different ways, such as shown in figures 2.11 and 2.12, where $G_p(s)$ denotes the actual, perturbed system dynamics and $G_o(s)$ a nominal model description of the physical system.

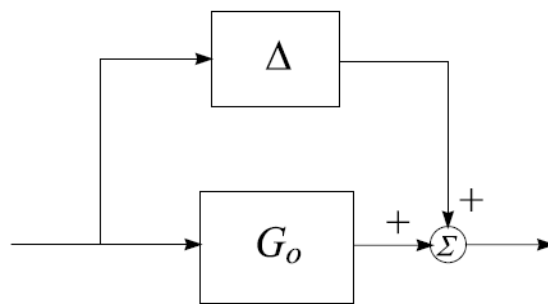


Figure 2.11: Additive perturbation (Gu et al, 2005)

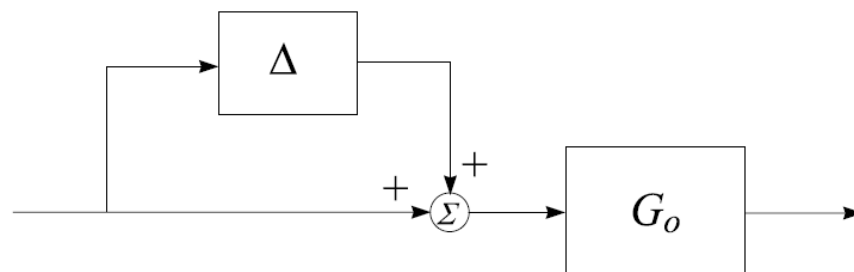


Figure 2.12: Input multiplicative perturbation configuration (Gu et al, 2005)

The additive uncertainty represents the absolute error between the actual manipulator dynamics and the nominal model representation of the system,

while the multiplicative representations show relative errors. According to Garulli et al, the unstructured uncertainty representation is used to describe unmodelled or difficult to model dynamics and it is usually given as a bound on some measure of the error signal between system and nominal model outputs for a chosen class of input signals. However, dynamic perturbations in many industrial control systems may also be caused by inaccurate description of component characteristics, torn-and-worn effects on plant components, or shifting of operating points, etc. Such perturbations may be represented by variations of certain system parameters over some possible value ranges. They affect the low-frequency range performance and are called parametric uncertainties (Gu et al, 2005). The structured uncertainty is a dynamic uncertainty represented by an element such as a finite dimensional vector or an operator, in some pre-specified uncertainty set of a suitable space. The uncertain elements in this case may be the coefficients of a transfer function, or the components of system matrices in a state-space realization.

2.8. Basic Control Systems

There are key basic subsystems of a control system based upon whether it is a closed or open loop system. In the open loop control system the basic subsystems are input, controller, plant and the output. It is simple and economical but lacks feedback element which makes it suffer from inaccuracies and optimization becomes impossible. However, a closed loop control system solves most problems of the open loop control system. The basic closed loop control subsystems are: reference input, controller, plant, measured output (actual output), transducer (or sensor) (Norman, 2011; Tarek et al). In the closed loop control system the output is compared with the reference input and error signal is generated. The error signal generated is fed to the controller to compensate the error so that the desired output is

produced. Therefore, the controller becomes a key factor in the design of a control system.

The joint torque at every joint of the robotic manipulator is optimized by the application of feedback subsystem and a controller designed based on the parameters of that joint and its motor. The term controller design refers to the process of selecting feedback gains that meet design specifications in a closed-loop control system. Most controller design methods are iterative, combining parameter selection with analysis, simulation, and insight into the dynamics of the plant (Ahmad et al, 2013; D'Azzo et al, 1988; Hedaya, 2011; Farhan, 2013). A controller is a device which can sense information from linear or nonlinear system (e.g., robot manipulator) to improve the systems performance (Kurfess, 2005; Slotine et al, 1991; Ogata, 2009; Cheng et al, 2008). Harry et al (2002) opined that any physical controller device that has a feedback structure is called a feedback controller. In terms of its mathematical model, the feedback structure of a controller is often represented by certain variables (representing what the controller 'sees') being mapped to other variables (representing the influence that the controller exerts on the system). The first kind of variables are called *measured outputs* of the system, the second kind of variables are called *control inputs* to the system. Typically, the input variables are considered to be *caused* by the measured outputs. Often, designing a controller for a given joint can be formulated as the problem of finding a suitable map between measured outputs of the joint and control inputs. It is always difficult for the joint to reproduce its input (desired output) at its output (actual output) due to uncertainties existing in the system. Thus, the performance of the joints is much affected by uncertainties.

2.8.1. Open Loop Control System

An open-loop control system is a control system which uses a controller and an actuator to obtain the desired output response, as shown in figure 2.13. An open-loop control system has no feedback sub-system; hence it cannot measure the nature of the actual output response to use it to control the input response.

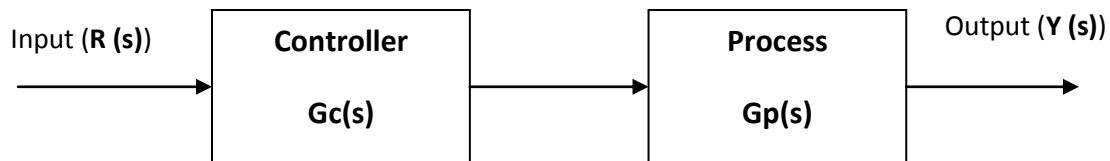


Figure 2.13: Block diagram of an open-loop control system

The output and the transfer function of an open-loop control system are generally express as follows:

$$Y(s) = R(s)G_c(s)G_p(s) \quad (2.1)$$

$$Tf(s) = \frac{Y(s)}{R(s)} = G_c(s)G_p(s) \quad (2.2)$$

Where

$Y(s)$ = output

$R(s)$ = Input

Tf = Transfer function of the open-loop control system

2.8.2. Closed-Loop Control System

A closed-loop control system is a system which utilizes an additional measure of the actual output to compare it with the input (desired output) response. A simple closed-loop control system is shown in figure 2.14. The measure of the actual output is called the feedback signal. Feedback control system is a control system that tends to maintain a prescribed relationship of one system variable to another by comparing the functions of these

variables and using the difference called error signal, as a measure of control. With an accurate sensor, the measured output is a good approximation of the actual output of the system.

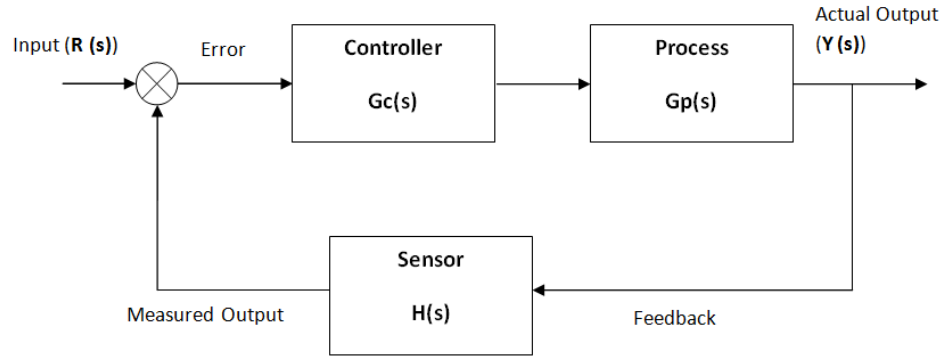


Figure 2.14: A block diagram of a closed-loop control system

The transfer function of the simple closed-loop control system is given as follows:

$$Tf = \frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)H(s)} \quad (2.3)$$

Where $H(s)$ is the feedback gain

According to Kambiz and Augustin, (2012), feedback control is a control mechanism that uses information from measurements. In a feedback control system, the output is sensed and fed back into the system. In the positive feedback control, the measured output $Y(s)$ is fed back into the system and adds to the reference input to give the error $e(s)$ which is fed into the controller. This approach increases the error signal. However, in the negative feedback control, the measured output $Y(s)$ is fed into the system and compared with the reference input $R(s)$ to produce the error $e(s)$ signal which is fed into the controller. This method reduces the error signal. The negative feedback systems are usually stable.

DC motors are quite popular in process industry due to various control characteristics. However due to some disadvantages, the robotic manipulators are mostly developed using stepper motors and induction motors. The operation of stepper motors is open-loop which ultimately produces very low performance results. The step response is very poor having significant overshoot and long settling time. Therefore, feedback control systems have been proposed for stepper motor position control systems. The same is also applicable on DC motor position control systems with acceptable results (Muhammad, 2013). The negative feedback was applied in (Farhan, 2013; Muhammad, 2013) for the control of DC motor for speed and position control.

According to Jaiswal and Kumar (2014), controlling robot manipulator is essential problem to guarantee the robot executing the desired task with minimum error. Many control techniques have been used to enhance the performance of the robot manipulators in different fields starting from conventional controllers such as Proportional integral derivative (PID) controller to intelligent controller such as neural network (NN) and fuzzyLogic controllers (Al-Khedher and Alshamasin, 2012; Alshamasin et al, 2012). However, considering the presence of uncertainties in the physical system and its model representation, the system controller design does not only base on the performance specifications such as settling time and overshoot but focuses on achieve system robustness in order to cancel the effects of the uncertainties and perform optimally in the presence of the uncertainties. This leads directly to the requirement of control design *robustness*, which demands that satisfactory performance is achieved for the uncertain model, i.e., the nominal model and the class of possible perturbations (Garulli et al).

2.9 Robust Control

In physical systems theory mathematical representation, output control is a problem of choosing the input u of the plant G such that the output y of this system approaches some specified reference input signal r with time. This is achieved in the presence of disturbance signals w . In the feedback control system in figure 2.15, K is an error feedback controller, which produces the input u of the system G using the control error $e = y - r$. In robust control, it is required that the output y of the system G converges to the reference signal r even if the parameters of the plant experience some perturbations. Robustness is an essential property for the controllers used in practical applications or physical systems, because the mathematical modeling of the physical system often require making approximations and linearization, and also assumptions which brings about mismatch or uncertainty between the real physical system and the model. The system uncertainties can be in two forms: unknown disturbances affecting the process, and incompletely known plant dynamics, i.e., model uncertainties.

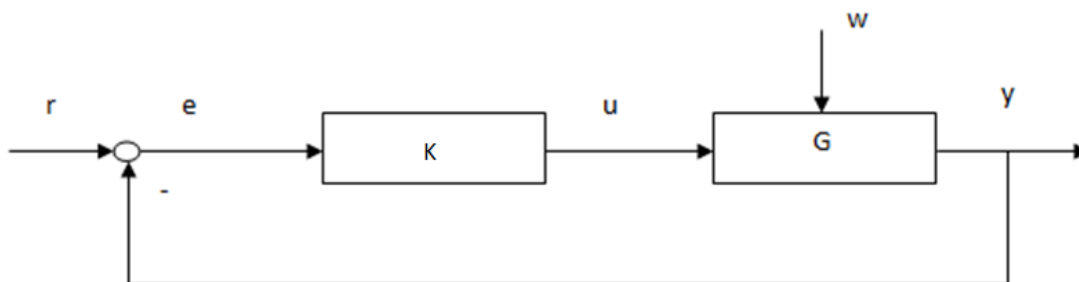


Figure 2.15: Feedback control structure of a physical system

The feedback control system can be restructured as illustrated in figure 2.16. In order to compensate for the uncertainties in the system the natural approach is to use measurements (y) from the process, and let the manipulated variable u be a function of the measured process outputs y as

well. In figure 2.16, v represents disturbance, y represents measured output, and K represents controller.

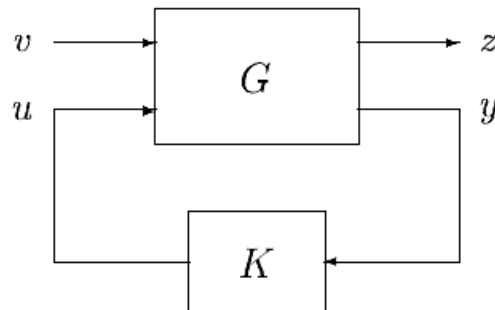


Figure 2.16: Feedback control system (Toivonen, 1998)

Gu et al (2005) stated that a control system is *robust* if it remains stable and achieves certain performance criteria in the presence of possible uncertainties. The robust design is to find a controller, for a given system, such that the closed-loop system is robust. Scherer (2001) opined that the main goal of robust control techniques is to take the uncertainties in a systematic fashion into account when analyzing a control system or when designing a controller for it.

The ultimate objective of the robust control is to minimize the effect of disturbance on output; the sensitivity S and the complementary function T are to be reduced (Bansal and Sharma, 2013). Considering $G(s)$ and $K(s)$ as the open loop transfer function of the plant and controller transfer function respectively, this will ensure robustness and good performance of closed loop system (Bansal and Sharma, 2013).

The design of a controller for the robotic manipulator involves generating the controller gains which can compensate the feedback error to produce a control signal u , which is then fed into the system in order to produce satisfactory performance of the manipulator. There are many controller

design approaches that have been applied depending on the control goal and the functions of the robotic manipulator. In this work, the goal is to design a robust controller for the manipulator joint torque control and the two controller design techniques employed are: the H-Infinity synthesis and the PID controllers.

In robust controller design, it is well-known that a well-designed control system should meet the following requirements besides the nominal stability (Tan et al, 1999):

- Disturbance attenuation,
- Set-point tracking,
- Robust stability and/or robust performance.

2.9.1 Proportional-Integral-Derivative (PID) Controller

PID controller has been implemented in many controller designs for automatic control systems. Shuaib and Ahmed, (2014) presented a work on robust PID control system design using Integral Time Absolute Error (ITAE) performance Index. Ahuja and Tandon (2013) designed a robust PID Controller Tuning for DC Motor Speed Control. In these works, PID controller was implemented to compensate the error due to the mismatch between the desired output and actual output for the improvement of the system performance. The PID controller design method is a common and effective method of controller gain selection approach. The controller gains are selected based on the system dynamics in order to compensate the error generated through feedback system and improve the system output. This approach employs three factors in its control technique: Proportional, Integral and Derivative. These factors contribute to the achievement of good control ability of the approach. The mathematical function of PID controller is represented as:

$$u(t) = K_p \left(e(t) + K_I \int e(t).dt + K_D \frac{de(t)}{dt} \right) \quad (2.6)$$

$$u(t) = Gc.e(t) \quad (2.7)$$

Where $u(t)$ is the controller output, $e(t)$ is the error signal.

The PID controller model can be represented as follows:

$$Gc = K_p \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (2.8)$$

where K_p is the proportional gain, T_I is the integral time constant, T_D is the derivative time constant, $K_I = K_p/T_I$ is the integral gain and $K_D = K_p/T_D$ is the derivative gain. The terms K_p , T_I and T_D definitions are:

- The proportional term: providing an overall control action proportional to the error signal through the all pass gain factor.
- The integral term: reducing steady state errors through low frequency compensation by an integrator.
- The derivative term: improving transient response through high frequency compensation by a differentiator.

These three variables K_p , T_I and T_D are usually tuned within given ranges. Therefore, they are often called the *tuning parameters* of the controller. By proper choice of these tuning parameters a controller can be adopted for a specific plant to obtain a good behaviour of the controlled system. The transfer function of the PID controller is give as:

$$Gc(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s} \quad (2.9)$$

There are several prescriptive rules used in PID tuning. The most effective methods generally involve the development of some form of process model, and then choosing P, I, and D based on the dynamic model parameters.

2.9.2 H-Infinity Synthesis

As presented by Toivonen(1998), robustness against model uncertainties is classically handled by phase and gain margins to ensure closed-loop stability in spite of modeling errors. The classical methods are, however, not readily generalized to multivariable plants, and they do not handle the problem of simultaneously achieving good performance against disturbances as well as robustness against model uncertainties. The modern approach to design controllers which are robust against model uncertainties is provided by the so-called H-Infinity control theory. This theory has been developed largely during the 1980's. In this approach, it is assumed that the plant is represented as:

$$G_p = G_0(s) + \Delta(s) \quad (2.10)$$

where G_0 is the 'nominal' plant model (often linear), and Δ represents a model uncertainty. The uncertainty is unknown but assumed to belong to some kind of uncertainty set D , i.e., $\Delta \in D$. For example, even though the nominal model may be linear, the uncertainty Δ may be nonlinear. In this way nonlinearities around a nominal linearized model G_0 valid at an operating point may be captured in the uncertainty Δ . The control system in figure 2.17 is called robustly stable with respect to the uncertainty D if it is stable for all $\Delta \in D$. For an important and realistic class of bounded uncertainties, it turns out that the condition that the closed loop is stable for all possible $\Delta \in D$ is equivalent to a bound on the so-called H-Infinity norm of the closed-loop transfer function.

The condition that the closed loop is stable for all possible $\Delta \in \mathcal{D}$ is equivalent to a bound on the H-Infinity norm of the closed-loop transfer function. The H-Infinity norm of the stable scalar transfer function $G(s)$ is defined as:

$$\|G\|_{\infty} = \max_{\omega} |G(j\omega)| \quad (2.11)$$

This means that, the H-Infinity norm is the largest gain of the system taken over all frequencies. This can be traceable from the Bode plot of the controlled system gains.

According to Nawash (2001), the real problem in robust multivariable feedback control system is to synthesize a control law which maintains system response and error signals to within pre-specified tolerances despite the effects of uncertainty on the system. Uncertainty may take many forms but among the most significant are noise, disturbance signals and transfer function modeling errors. Another source of uncertainty is unmodelled nonlinear distortion. These forms of uncertainties have different negative effects on the performance of the system, therefore they must be considered in the design of the system controller.

Two categories of uncertainties in robotic arm are the disturbance signals and dynamic perturbations. The disturbance signals include input and output disturbance, sensor noise and actuator noise, *etc.* The dynamic perturbation (also known as model uncertainty or error) results from unmodelled dynamics, neglected nonlinearities in the modeling, effects of deliberate reduced-order models, and system-parameter variations due to environmental changes and torn-and-worn factors.

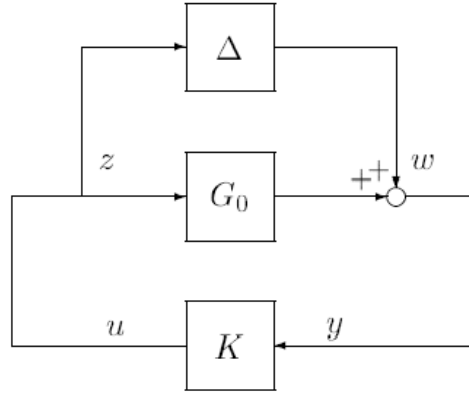


Figure 2.17: Control of uncertain plant (Toivonen,1998).

Conventionally, H_∞ controller synthesis employs two transfer functions which divide a complex control problem into two separate sections, one dealing with stability and disturbance rejection, the other dealing with performance and noise suppression: The sensitivity function, S , and the complementary sensitivity function, T , which are required for the controller synthesis and are given in (Bansal and Sharma, 2013; Nair, 2011).

Sensitivity function is the ratio of output to the disturbance of a system and complementary sensitivity function is the ratio of output to input of the system (Bansal and Sharma, 2013; Nair, 2011).

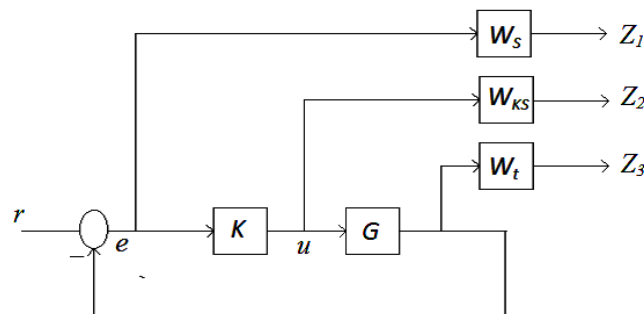


Figure 2.18: Plant model for the H-infinity synthesis (Bansal and Sharma, 2013)

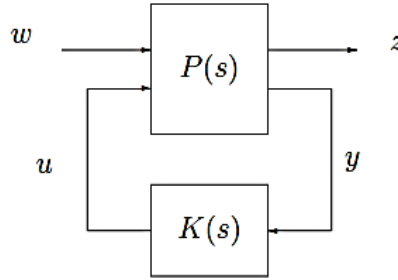


Figure 2.19: General control problem(Bansal and Sharma, 2013)

Now the objective is to find a controller K , which, based on the information in y , generates a control signal u , which counteracts the influence of w on z , thereby minimizing the closed loop norm w to z . The ultimate objective of the robust control is to minimize the effect of disturbance on output; hence the sensitivity S and the complementary function T are to be reduced. This can be achieved by minimizing the magnitude of $|S|$ and $|T|$ by making

$$|S(j\omega)| < \frac{1}{W_s(j\omega)} \text{ and } |T(j\omega)| < \frac{1}{W_t(j\omega)}$$

W_s is the performance weighting function to limit the magnitude of the sensitivity function and W_t is the robustness weighting function to limit the magnitude of the complementary sensitivity function. This technique known as loop shaping technique is widely used for selecting the weight functions for the synthesis of the controller. The loop shaping follows the trajectories for the plot of S and T as shown in figure 2.20 and figure 2.21.

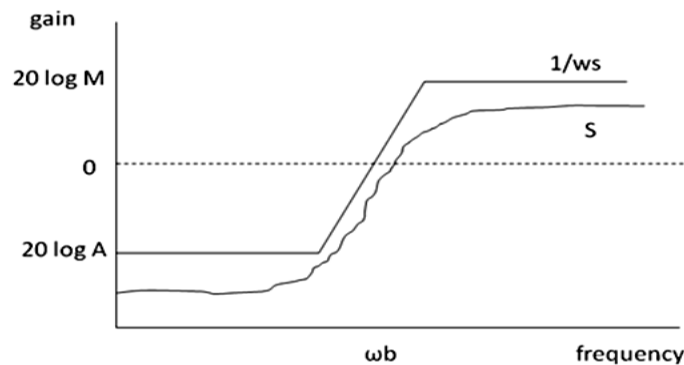


Figure 2.20: Sensitivity graph (Nair, 2011)

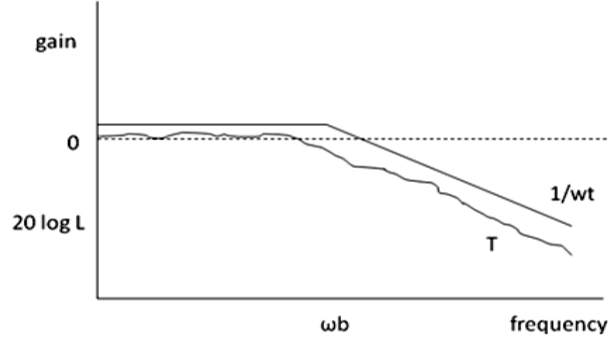


Figure 2.21: Complementary sensitivity graph (Nair, 2011)

Loop shaping is done to make the frequency response of the plant with the weight functions to come in the desired manner. In loop shaping the parameters of the weight functions are changed to make the frequency response of the whole system to remain within limits. The control synthesis requires the plant transfer function, controller transfer function and the various weight functions to augment together. Thus an augmented plant model is made as shown in figures 2.18 and 2.19. The generalized plant $P(s)$ is given as (Bansal and Sharma, 2013; Nair, 2011):

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ e \end{bmatrix} = \begin{bmatrix} W_s & -W_s G \\ 0 & W_{ks} \\ 0 & W_t G \\ I & -G \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \quad (2.12)$$

A possible state space realization for $P(s)$ can be written as

$$P = \begin{bmatrix} W_s & -W_s G \\ 0 & W_{ks} \\ 0 & W_t G \\ I & -G \end{bmatrix} \quad (2.13)$$

From (2.13) a mixed sensitivity problem or the optimization problem is written as follows:

$$P = \begin{bmatrix} W_s S \\ W_{ks} K S \\ W_t T \end{bmatrix} \quad (2.14)$$

Where W_s , W_{ks} and W_t are the tuning parameters called weights and it typically requires some iteration to obtain the weights which will yield a good controller.

As stated by Bansal and Sharma(2013) and Nair(2011), in case of optimization problem the objective is to find a rational function controller $K(s)$ to make the closed loop system stable, optimize performance and satisfying the following expression

$$\min \|P\| = \min \begin{bmatrix} W_s S \\ W_{ks} K S \\ W_t T \end{bmatrix} = \gamma$$

Where P is the transfer function from w to z . i.e

$|T_{zw}| = \gamma$ (is the cost function)

H-infinity optimization problem is formulated as the task of designing a stabilizing controller $K(s)$, which internally stabilizes the closed-loop system with good reference tracking, reduces sensitivity to disturbances and suppresses noise; thereby achieving a robust and performance optimized system.

In summary, the robot suffers from the problem of uncertainties which limits its performance due to large tracking error; to solve this, a robust controller which can reduce the error and maintain stability of the system is required.

CHAPTER THREE

DESIGN METHODOLOGY

The materials/tools, modeling methods and analysis employed in the development of a robust control for an articulated robotic manipulator under uncertainties are discussed in this chapter. The mathematical models for the

robotic manipulator are very important for the development of the system. In order to achieve robust joint torque control, the robot manipulator analysis must involve the source of the joint torque. Most existing models applied either only the arm or the actuator dynamics. Some of the existing works neglected some actuator or arm dynamics. The development of mathematical model for joint torque control of the manipulator must involve a complete dynamical description of the entire system comprising of the links (rigid bodies) and the actuators. Therefore, a complete dynamic model comprising of the link dynamics and actuator dynamics was proposed in this work for torque control of the manipulator. Independent joint control strategy was adopted in order to separately control the joint torques to enable precise robust controller design for every joint of the manipulator.

However, due to the mismatch between the manipulator and the mathematical model, robust control was introduced to address the problem of uncertainties in the system by designing a controller for the joint torque control based on strict robustness design specifications (i.e. Reference tracking, sensitivity etc.) that can annul the effects of the uncertainties.

Computer Aided Design (CAD) model of the 3DOF articulated robotic manipulator is generated using Autodesk inventor application software. The purpose of employing this software is to provide a 3D model representation of the system and generate equivalent digital block diagram representation called SimMechanics model of the robot manipulator in Matlab with SimMechanics add-on for simulating the 3Drobot manipulator mechanical system. The SimMechanics model provides mass properties of the manipulator parts designed in 3D and also allows attachment of the designed controller gains to the manipulator joints of the SimMechanics model through the joint actuators. The actuator model for the 3DOF articulated robotic manipulator was developed and implemented in Simulink model.

3.1 Materials/Tools

There are basically three application software used in the realization of this research work namely: Autodesk Inventor Professional 2015, Simmechanics add-on Toolbox, and Matlab/Simulink. Table 1 shows a detailed summary of the materials.

Table 3.1: Summary of materials/Tools

Material/Tool	Description	Purpose
Autodesk Inventor Professional 2015	Autodesk Inventor, developed by U.S. based software company Autodesk, is a computer-aided design application for creating 3D digital prototypes used in the design, visualization and simulation of products	To create an accurate 3D design of the articulated robotic manipulator. To enable digital model conversion of the robot arm from the 3D model for analysis and control
SimMechanics	SimMechanics is an add-on toolbox in Matlab. It is a part of Physical Modeling. Physical Modeling runs within the Simulink environment and interfaces seamlessly	The purpose of SimMechanics is the engineering design and simulation of mechanical systems of rigid bodies connected by joints, with the standard Newtonian dynamics of forces and torques.

	with the rest of Simulink and with MATLAB	To provide a suite of tools to specify bodies and their mass properties, their possible motions, kinematic constraints, coordinate systems, and the means of initiating and measuring motions
Simulink/MatlabR2014a	Simulink is a block diagram environment for multi-domain simulation and Model-Based Design in Matlab.	Its purpose is to create block diagram environment of the design for simulation, automatic code generation, and continuous test and verification.

3.2. Robotic Manipulator Parameters

The robotic manipulator parameters angular position, velocity, and the acceleration of the arms, these parameters are attributed to the torque produced by the actuators moving the arms.

3.2.1 Angular Velocity

Motors are devices that convert electrical energy into mechanical energy. The D.C. motors convert electrical energy into rotational energy. That rotational energy is then used to lift, propel, or turn objects, etc. When specified voltage is supplied to a motor, it rotates the output shaft that carries the link of the manipulator at some speed. This rotational speed or angular velocity ($\omega = \dot{\theta}$), is typically measured in radians/second {rad/s}, revolutions/second {rps}, or revolutions/minute {rpm}

3.2.2 Power

When a torque (with respect to the axis of rotation) acts on a body that rotates with angular velocity, its power (which is defined as the rate of doing work) is the product of the torque and angular velocity and it is calculated using the following relationship:

$$Power = \tau \times \omega \quad (3.1)$$

3.2.3 Torque

Torque is defined as the turning or twisting **force** and it is calculated using the following relationship:

$$Torque (\tau) = Force(F) \times Length (L) \quad (3.2)$$

$$\tau = F \times L \quad (3.3)$$

$$F = W = m \times g \quad (3.4)$$

Where W is the weight, m is the mass, and g is the acceleration due to gravity, therefore,

$$\tau = m \times g \times L \quad (3.5)$$

$$\tau = W \times L \quad (3.6)$$

In order to estimate the torque required at each joint, the worst case scenario must be chosen. In figure 3.1, a link of length L is rotated clockwise. Only the perpendicular component of length between the pivot and the force is taken into account. We observe that this distance decreases from L3 to L1 (L1 being zero). Since the equation for torque is length (or distance) multiplied by the force, the greatest value will be obtained using L3, since F does not change. The link can be similarly rotated counterclockwise and the same effect will be observed.

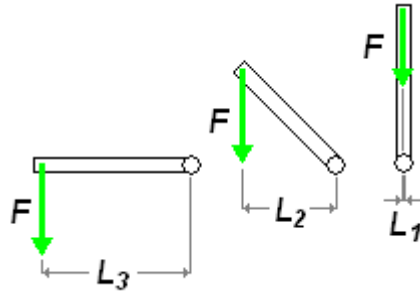


Figure 3.1: Rotation of link length (Benson, 2013)

This means that a joint revolving vertically is not involved in weight lifting and thus has a zero torque.

3.2.4 Calculations of Joint Torques

The major purpose of joint force calculations in figure 3.2 is for motor selection. In most robot arm designs the weight of the robot arm and the weight of a possible load are considered in choosing a motor in order to achieve a good design that can support the weight of the arm and the load.

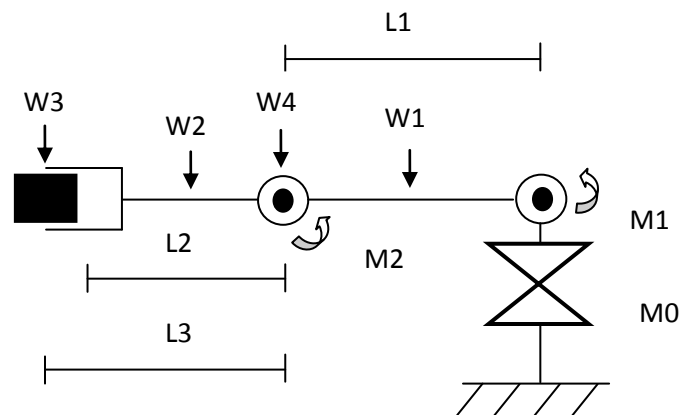


Figure 3.2: Force calculation of joints

To carry out a moment arm calculation, multiplying downward force times the linkage lengths. This calculation must be done for each lifting actuator. The manipulator design in this work has two DOF that requires lifting, and

the center of mass of each linkage is assumed to be Length/2. The remaining degree of freedom does not require lifting. For the manipulator to function properly the following point must be met:

- The total torque about a joint must be equal or less than the torque produced by the actuator

$$\sum \tau_{about \ a \ joint} \leq \tau_{produced \ by \ the \ actuator} \quad (3.7)$$

The robotic manipulator joint torque is determined as follows:

Torque about Joint 1:

$$M_1 = \frac{L_1}{2} * W_1 + L_1 * W_4 + \left(L_1 + \frac{L_2}{2}\right) * W_2 + (L_1 + L_3) * W_3 \quad (3.8)$$

Torque about Joint 2:

$$M_2 = \frac{L_2}{2} * W_2 + L_3 * W_3 \quad (3.9)$$

Where:

m1=mass of link1

m2=mass of link2

m3=mass of load

m4=mass of actuator2

W1 = Weight of link1

W2 = Weight of link2

W3 = Weight of load

W4 = Weight of actuator (servo) 2

L = Length

L1 = Length of link1

L2 = Length of link2

L_{Load} = Length of the load (end-effector)

L3 = Length2 + (1/2*L_{Load})

M0 = Base Actuator (Cylindrical movement)

M1 = Actuator 1 (Joint I)

M2 = Actuator 2 (Joint II)

The above equations for torque calculation only deal with the case where the robot arm is being held horizontally (not in motion). For the arm to move from a rest position, acceleration is required. To solve for this added torque, it is known that the sum of torques acting at a pivot point is equal to the moment of inertia (**J**) multiplied by the angular acceleration (**a**):

$$\tau = J \times a \quad (3.10)$$

To calculate the extra torque required to move (i.e. create an angular acceleration) the moment of inertia of the part from the end to the pivot is calculated using the equation (Benson, 2013):

$$J = \frac{m \times L^2}{2} \quad (3.11)$$

Where m is the mass of the robot arm link and L is the length of the link. Note this equation calculates the moment of inertia about the center of mass.

Therefore, when choosing an actuator for the manipulator, the added torque to the static or holding torque also known as the detent torque is considered. Detent torque is the torque required to rotate the motor's output

shaft with no current applied to the windings. As with all dynamic tools, inefficiencies in the actuators and joints themselves must also be taken into consideration. This way, the motor at each joint will be able to provide more than the required torque to keep the arm stationary. The required torque to accelerate the weight being support by an actuator from a static position can be calculated using the following relation (Benson, 2013):

$$\sum \tau = \tau_{holding} + \tau_{motion} = J * a \quad (3.12)$$

3.3 Forward Kinematics Model

Forward kinematics as illustrated in figure 3.3 is the method for determining the orientation and position of the end effector, given the joint angles and link lengths of the robot arm. Hence the end effector location is calculated with given joint angles and link length of links.

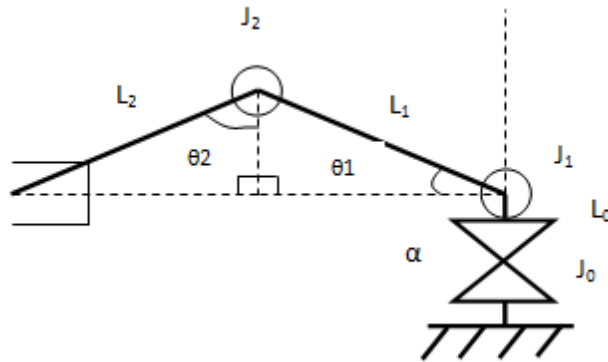


Figure 3.3: kinematics of a two-link 3DOF planar arm

Assume that the base is located at $x=0$ and $y=0$. The first step would be to locate x and y of each joint.

Joint 0 (with x and y at base):

$$x_0 = 0$$

$$y_0 = L_0 = 0$$

Joint 1 (with x and y at J1):

$$\cos \theta_1 = \frac{x_1}{L_1} \quad (3.13)$$

$$x_1 = L_1 \cos \theta_1 \quad (3.14)$$

$$\sin \theta_1 = \frac{y_1}{L_1} \quad (3.15)$$

$$y_1 = L_1 \sin \theta_1 \quad (3.16)$$

Joint 2 (with x and y at J2):

$$\sin \theta_2 = \frac{x_2}{L_2} \quad (3.17)$$

$$x_2 = L_2 \sin \theta_2 \quad (3.18)$$

$$\cos \theta_2 = \frac{y_2}{L_2} \quad (3.19)$$

$$y_2 = L_2 \cos \theta_2 \quad (3.20)$$

End Effector Location:

$$x_0 + x_1 + x_2$$

or

$$0 + L_1 \cos \theta_1 + L_2 \sin \theta_2$$

$$y_0 + y_1 + y_2$$

or

$$0 + L_1 \sin \theta_1 + L_2 \cos \theta_2$$

$z = \alpha$ (alpha), in cylindrical coordinates

The coordinates (x, y) of the tool are expressed in this coordinate frame as

$$x = x_2 = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (3.21)$$

$$y = y_2 = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (3.22)$$

L_1 and L_2 are the lengths of the two links, respectively. Also the orientation of the tool frame relative to the base frame is given by the direction cosines of the x_2 and y_2 axes relative to the x_0 and y_0 axes, that is:

$$\begin{aligned} x_2 \cdot x_0 &= \cos(\theta_1 + \theta_2); \quad x_2 \cdot y_0 = -\sin(\theta_1 + \theta_2) \\ y_2 \cdot x_0 &= \sin(\theta_1 + \theta_2); \quad y_2 \cdot y_0 = \cos(\theta_1 + \theta_2) \end{aligned}$$

which we may combine into an orientation matrix

$$\begin{bmatrix} x_2 \cdot x_0 & y_2 \cdot x_0 \\ x_2 \cdot y_0 & y_2 \cdot y_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (3.23)$$

Equations (3.21), (3.22) and (3.23) are referred to as the forward kinematic equations for this arm.

3.4 Inverse Kinematics Model

Inverse kinematics is referred to as the opposite of forward kinematics. This is when the desired end effector position is known, but the joint angles required to achieve the desired position is to be calculated.

Using the Law of Cosines, the angle θ_2 is given by:

$$\cos \theta_2 = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2} = D \quad (3.24)$$

θ_2 can be determined as

$$\theta_2 = \cos^{-1}(D) \quad (3.25)$$

Alternatively, if $\cos(\theta_2)$ is given by equation (2.4) then $\sin(\theta_2)$ is given as:

$$\sin(\theta_2) = \pm\sqrt{1 - D^2} \quad (3.26)$$

and, hence, θ_2 can be found by

$$\theta_2 = \tan^{-1} \frac{\pm\sqrt{1-D^2}}{D} \quad (3.27)$$

The advantage of this alternative approach is that both the elbow-up and elbow-down solutions are recovered by choosing the positive and negative signs in equation (3.27), respectively. θ_1 is now given as

$$\theta_1 = \tan^{-1}(y/x) - \tan^{-1} \left(\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2} \right) \quad (3.28)$$

Notice that the angle θ_1 depends on θ_2 .

Inverse kinematics is difficult in implementation because of the following points:

- It involves non-linear simultaneous equations
- There is the very likely possibility of multiple, sometimes infinite, number of solutions. How would the robot arm choose which is optimal, based on torques, previous arm position, gripping angle, etc.?
- There is the possibility of zero solutions. Maybe the location is outside the workspace, or maybe the point within the workspace must be gripped at an impossible angle.
- Singularities, a place of infinite acceleration, can blow up equations and/or leave motors lagging behind.

The kinematic models are not involved in the joint torque control of the robotic manipulator

3.5. Robot Arm Dynamic Model

The dynamics of an n -DOF robot manipulator is governed by the following equation (Cheng et al, 2008; Kurfess, 2005; Vivas and Mosquera, 2005):

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (3.35)$$

Where τ is actuation torque, $M(q)$ is a symmetric and positive definite inertia matrix, N is the vector of nonlinearity term.

$$N(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q)$$

Hence

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (3.36)$$

Where $\mathbf{q}$ is the joint variable vector, $\mathbf{M}(\mathbf{q})$ is the completed inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the centripetal and Coriolis torque vector, $\mathbf{G}(\mathbf{q})$ is the gravitational torque vector. Adopting the model in Liu and Liu (2016), the robot arm dynamic equation for 2DOF that requires lifting is:

$$\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (3.37)$$

Where τ_1 and τ_2 are the total torques at joint I and II respectively.

The robot manipulator has two joint variables that require lifting: two angles q_1 and q_2 . The inertia matrix is represented as:

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

where

$$\begin{aligned} M_{11} &= \frac{1}{4}m_1L_1^2 + m_2L_1^2 + J_1 \\ M_{12} = M_{21} &= \frac{1}{2}m_2L_1L_2 \cos(q_1 - q_2) \\ M_{22} &= J_2 + \frac{1}{4}m_2L_2^2 \end{aligned}$$

The centripetal and coriolis matrix is:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Where

$$\begin{aligned} C_{11} &= 0 \\ C_{12} &= \frac{1}{2}m_2L_1L_2 \sin(q_1 - q_2)\dot{q}_2 \\ C_{21} &= \frac{1}{2}m_2L_1L_2 \sin(q_2 - q_1)\dot{q}_1 \\ C_{22} &= 0 \end{aligned}$$

Due to the complexity of this method of robotic manipulator dynamics analysis, Lin et al (1997) assumed that $G(\mathbf{q}) = 0$. However, Kim et al (2010) and Liu et al (2016) did not take such assumption.

$$\begin{aligned} G_1 &= \left(\frac{1}{2}m_1 + m_2\right)gL_1 \cos q_1 \\ G_2 &= \frac{1}{2}m_2gL_2 \cos q_2 \end{aligned}$$

Lin et al (1997) added the friction vector in their dynamic model and ignored gravitational vector, while in Kim et al, (2010), and Liu et al (2016), friction vector was ignored. There are too many inconsistencies in the Lagrange-Euler dynamic model as used in different works due to the complexities in the model. Secondly, the model as used in many works deals with only the

arm parameters (mass, length, angular position, velocity, and acceleration of the arm) without involving the actuator parameters.

3.6 Actuator DynamicModel

The dynamic terms in the equation 3.35 and 3.36 are only manipulator positions. However, the fact that the manipulator is driven by actuator (which in this work is motor), the dynamic equation of a manipulator driven by DC motors (Spong, 1996; Fateh, 2008) is formulated as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = K_t i \quad (3.38)$$

where i is the armature current vector, and K_t is the diagonal matrix of motor torque constant.

The actuator nominal model is derived from the motor internal structure as shown in figure 3.4. In the nominal model, it is assumed that the system has no disturbance. Hence, the model is derived without considering disturbances.

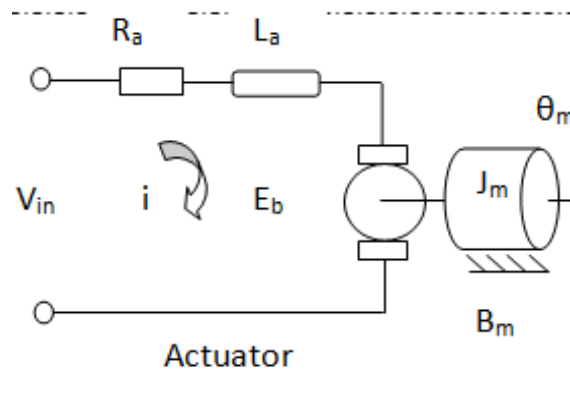


Figure 3.4: The actuator internal structure

$$\tau_m = K_t i \quad (3.38)$$

Sum of torques at the motor gear is equal to zero, that is:

$$J_m \frac{d^2\theta_m}{dt^2} + B_m \frac{d\theta_m}{dt} = K_t i \quad (3.39)$$

$$J_m \frac{d^2\theta_m}{dt^2} = K_t i - B_m \frac{d\theta_m}{dt}$$

$$\frac{d^2\theta_m}{dt^2} = \frac{1}{J_m} \left(K_t i - B_m \frac{d\theta_m}{dt} \right) \quad (3.40)$$

Taking the Laplace transform of equation 3.39 yields;

$$J_m \omega_m s(s) + B_m \omega_m(s) = K_t I(s) \quad (3.41)$$

The electrical circuit of the actuator provides the following equation (Spong and Vidyasagar, 1989):

$$V_{in} = Ri + L_a \frac{di}{dt} + K_e \frac{d\theta_m}{dt} \quad (3.42)$$

$$L_a \frac{di}{dt} = -Ri + V_{in} - K_e \frac{d\theta_m}{dt} \quad (3.43)$$

Taking the Laplace transform of equation 3.42

$$V_{in}(s) = RI(s) + L_a sI(s) + K_e \omega(s) \quad (3.44)$$

Where ω is the angular velocity

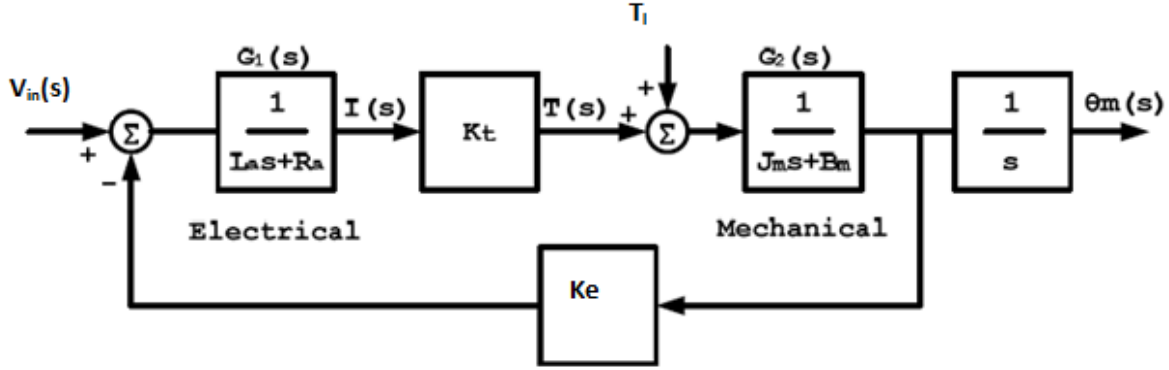


Figure 3.5: Block diagram for DC motor system (Alassar, 2010)

Resolving the equations with the help of the actuator block diagram in figure 3.5 and solving for relationship between voltage input and the position of the shaft in the closed loop system yields the actuator dynamic model as derived in (Ovy et al, 2011; Alassar, 2010; Salem, 2013; Fateh, 2008). When the motor is not connected to the robotic manipulator joint mechanism (i.e. $T_l=0$), its dynamic model does not include the inertia due to the connected linkage or load. Combining the mechanical and electrical subsystem dynamics of the motor, therefore motor dynamic model becomes:

$$\begin{cases} Js\omega_m + B\omega_m = K_t I \\ V_{in} = RI + L_a sI + K_e \omega_m \end{cases}$$

The model for the variable $\theta_m(s)$ becomes:

$$\theta_m(s) = (L_a J_m s^3 + (R_a J_m + B_m L_a) s^2 + (R_a B_m + K_t K_e) s)^{-1} K_t V_{in}(s) \quad (3.45)$$

Where,

J_m is the motor inertia,

R_a is the actuator resistance,

L_a is the actuator inductance,

K_t is the torque constant,

K_e is the back emf constant,

B_m is the frictional damping coefficient of motor

$\omega_m = \dot{\theta}_m$ is the velocity of the motor

3.7 Robotic Manipulator Joint Dynamic Model

According to Fateh (2008), there are some problems in implementing the control law presented in equation 3.36. This control law is not complete since some terms such as frictional torques have been omitted for simplicity and reducing the computing time, and some terms are not precise. Therefore, applying this control law cannot provide a perfect linear and decoupled system, and due to inaccuracy in model, errors will be produced. Moreover, implementing the control law requires feedbacks of all joint positions and their derivatives. Also, the control strategy is complex since the system is highly coupled and multi-input/multi-output. The tracking error increases as velocity increases.

Fateh (2008) also stated that the dynamic model involving the actuator dynamics is preferred to equation 3.36 in the robot manipulator design. This is because all feedbacks are belonging to the actuator (motor). Also, manipulator model is not required to form the control law. As a result, the control law is simple, fast, and more accurate in comparison with the equation 3.36. The control law requires only a feedback of actuator current and current of the actuator. Moreover, the electrical signals can be measured more convenient and more precise than mechanical signals. This control law can be used for tracking control of a high-speed robot since this approach is free of manipulator model. In facts, the dynamical effects are

compensated by currents of motors in high-speed applications. The actuator dynamic equation is used for precise control of each degree of freedom (or each independent joint) of a robotic arm as applied in (Agrawal et al, 2012; Aung et al, 2008; Salem, 2013).

The manipulator is made up of links connected together by joints and each joint consists of actuator and gears (motor and link gears) connecting the arm to the joint as shown in figure 3.6, thus, a complete dynamic model of the system must consists of the robot arm dynamics plus the actuator dynamics. Since these two vital dynamics are derived at the joint, then it is termed joint dynamics model. Also contained as part of every joint is the feedback subsystem which measures the **joint outputs** (e.g. position of link) and sends the measured quantity back to the joint system for improvement of the joint performance with the help of a controller designed for that particular joint and connected to the joint motor. Applying the independent joint control and considering the link connected at the joint as a load whose inertia sums with the inertia of the motor to form the inertia of the joint. The gear ratio of the motor and gear ratio of link are related to the position of the motor $\theta_m(t)$ and the position of the link $q(t)$ as expressed in the gear kinematics in equation 3.50.

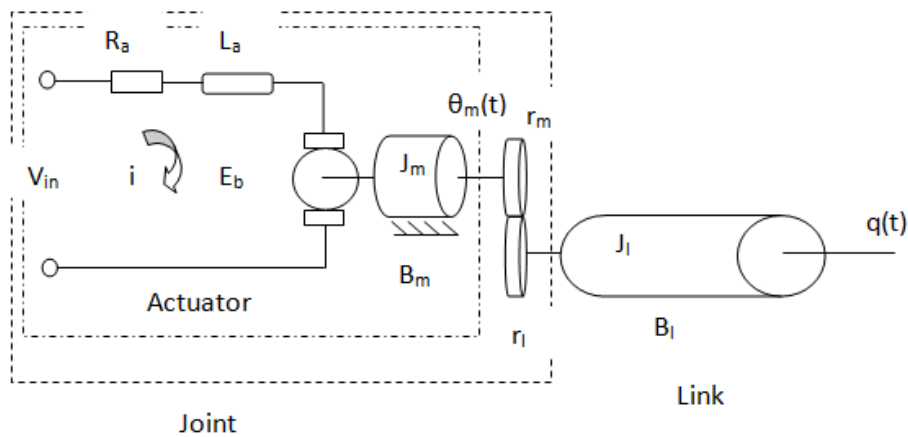


Figure 3.6: Internal structure of the joint

3.7.1 Mechanical Subsystems

The mechanical subsystems of the actuator and the robot arm are connected at the actuator and robot arm gears as shown in figures 3.6 and 3.7.

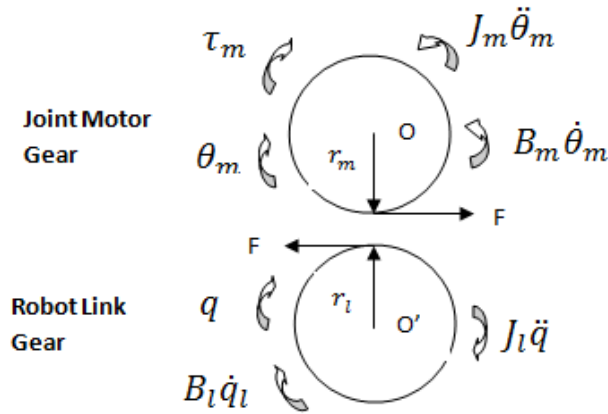


Figure 3.7: Mechanical subsystems of the motor and link gears

Note: i. The Sum of torques acting at a point is equal to the moment of inertia (J) multiplied by the angular acceleration.

ii. Inertia of the link is determined using its mass and length

The sum of torques at the motor gear gives:

$$\sum \text{torques at the motor gear} = J_m \ddot{\theta}_m$$

$$\tau - B_m \dot{\theta}_m - r_m F = J_m \ddot{\theta}_m$$

$$J_m \ddot{\theta}_m + B_m \dot{\theta}_m + r_m F = \tau \quad (3.46)$$

The sum of torques at the robot arm gear gives:

$$\sum \text{torques at the link gear} = J_l \ddot{q}$$

$$r_l F - B_l \dot{q} = J_l \ddot{q}$$

$$J_l \ddot{q} + B_l \dot{q} = r_l F \quad (3.47)$$

$$F = (J_l \ddot{q} + B_l \dot{q}) / r_l \quad (3.48)$$

Substituting equation into equation 3.48 into equation 3.46 yields;

$$J_m \ddot{\theta}_m + B_m \dot{\theta}_m + \frac{r_m}{r_l} (J_l \ddot{q} + B_l \dot{q}) = \tau \quad (3.49)$$

The gear kinematics for the motor and link (robot arm) gears is as follows

$$\frac{r_m}{r_l} = \frac{N_m}{N_l} = \frac{\dot{q}}{\dot{\theta}_m} = \frac{q}{\theta_m} \quad (3.50)$$

Where

r_m is the radius of motor gear

r_l is the radius of link gear

N_m is the number of teeth of the motor gear

N_l is the number of teeth of the link gear

F is the contact force

J_l is the link or arm inertia

B_l is the link or arm damping coefficient

Therefore, the angular position of the link is derived from the motor position and the gear ratio of motor and link gears as:

$$\theta_m = \left(\frac{N_l}{N_m}\right) q \quad (3.51)$$

$$\dot{\theta}_m = \left(\frac{r_l}{r_m}\right) \dot{q} \quad (3.52)$$

$$\ddot{\theta}_m = \left(\frac{r_l}{r_m}\right) \ddot{q} \quad (3.53)$$

Substituting equations 3.51, 3.52 and 3.53 into equation 3.49, yields:

$$\begin{aligned} \frac{r_l}{r_m} J_m \ddot{q} + \frac{r_l}{r_m} B_m \dot{q} + \frac{r_m}{r_l} (J_l \ddot{q} + B_l \dot{q}) &= \tau \\ \frac{r_l}{r_m} J_m \ddot{q} + \frac{r_l}{r_m} B_m \dot{q} + \frac{r_m}{r_l} J_l \ddot{q} + \frac{r_m}{r_l} B_l \dot{q} &= \tau \\ \left(\frac{r_l}{r_m} J_m + \frac{r_m}{r_l} J_l\right) \ddot{q} + \left(\frac{r_l}{r_m} B_m + \frac{r_m}{r_l} B_l\right) \dot{q} &= \tau \end{aligned} \quad (3.54)$$

These equations describe the model for torque control. Combining the mechanical and electrical subsystems of the actuator and arm dynamics yields:

$$\begin{cases} \left(\frac{r_l}{r_m} J_m + \frac{r_m}{r_l} J_l\right) s^2 q + \left(\frac{r_l}{r_m} B_m + \frac{r_m}{r_l} B_l\right) s q = K_t I \\ V_{in} = RI + L_a s I + \frac{r_l}{r_m} K_e \dot{q} \end{cases} \quad (3.55)$$

The dynamic model for joint torque control relating angular position of the link and the voltage input into the actuator becomes:

$$\begin{aligned} G &= \left(s \left[\left(\frac{r_l}{r_m} J_m + \frac{r_m}{r_l} J_l \right) s + \left(\frac{r_l}{r_m} B_m + \frac{r_m}{r_l} B_l \right) \right] (L_a s + R) + K_e K_t \right)^{-1} K_t \\ q &= \left(s \left[\left(\frac{r_l}{r_m} J_m + \frac{r_m}{r_l} J_l \right) s + \left(\frac{r_l}{r_m} B_m + \frac{r_m}{r_l} B_l \right) \right] (L_a s + R) + K_e K_t \right)^{-1} K_t V(s) \end{aligned} \quad (3.56)$$

Simplifying the joint mechanical subsystem dynamics yields:

$$\begin{cases} J_T \ddot{q} + B_T \dot{q} = \tau \\ V_{in} = RI + L_a s I + \frac{r_l}{r_m} K_e \dot{q} \end{cases} \quad (3.57)$$

Where $J_T = \frac{r_l}{r_m} J_m + \frac{r_m}{r_l} J_l$ is the total inertia at the joint and $B_T = \frac{r_l}{r_m} B_m + \frac{r_m}{r_l} B_l$ is the total torsional viscous damping coefficient

Applying independent joint control for the torque control of the two joints of the robotic manipulator requires deployment of multiple SISO configurations.

$$\begin{cases} J_{T1} \ddot{q}_1 + B_{T1} \dot{q}_1 = \tau_1 \\ V_1 = R_1 I_1 + L_{a1} s I_1 + \frac{r_{l1}}{r_{m1}} K_{e1} \dot{q}_1 \end{cases} \quad (3.58)$$

$$\begin{cases} J_{T2} \ddot{q}_2 + B_{T2} \dot{q}_2 = \tau_2 \\ V_2 = R_2 I_2 + L_{a2} s I_2 + \frac{r_{l2}}{r_{m2}} K_{e2} \dot{q}_2 \end{cases} \quad (3.59)$$

Where τ_1 and τ_2 are torques at joint I and II respectively.

The joint general variable becomes:

$$q_1 = \left(s \left[\left(\frac{r_{l1}}{r_{m1}} J_{m1} + \frac{r_{m1}}{r_{l1}} J_{l1} + J_{l2} \right) s + \left(\frac{r_{l1}}{r_{m1}} B_{m1} + \frac{r_{m1}}{r_{l1}} B_{l1} \right) \right] (L_{a1} s + R_1) + K_{e1} K_{t1} \right)^{-1} K_{t1} V(s)$$

$$q_2 = \left(s \left[\left(\frac{r_{l2}}{r_{m2}} J_{m2} + \frac{r_{m2}}{r_{l2}} J_{l2} \right) s + \left(\frac{r_{l2}}{r_{m2}} B_{m2} + \frac{r_{m2}}{r_{l2}} B_{l2} \right) \right] (L_{a2} s + R_2) + K_{e2} K_{t2} \right)^{-1} K_{t2} V(s)$$

These describe the angular position of the robotic manipulator links I and II.

Figure 3.8 shows the internal structure of the 3DOF articulated robotic manipulator. It also shows the respective link load on the joints.

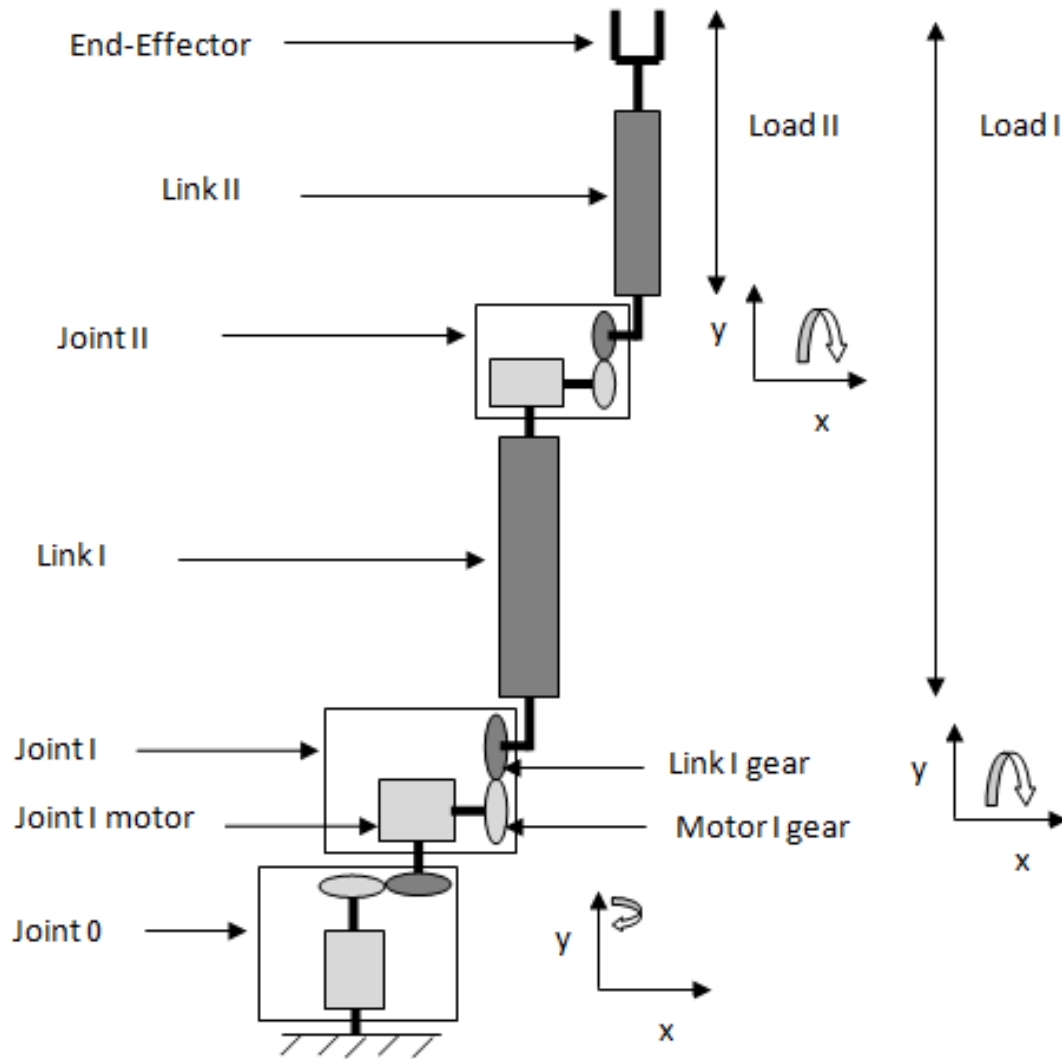


Figure 3.8: Internal structure of the 3DOF articulated robotic manipulator

3.8 Uncertainty Model for Robotic Manipulator

The robust control law here solves the robust control problem which is the uncertainty in the manipulator. Since the uncertainties are unavoidable, thus the robust control goal here is to involve all the non-negligible dynamics and consider the unmodeled dynamics as uncertainties. Any coupling effect due to the motion of the other links is treated as a disturbance. Considering the uncertainties in the manipulator arm movement, the dissipative torques such

as frictional torques and the provided load torque can be added to the left hand of the general dynamic model. Taking the nominal model of the robot arm dynamic model as:

$$M_n(q)\ddot{q} + C_n(q, \dot{q})\dot{q} + g_n(q) = \tau \quad (3.60)$$

Under external disturbances and plant uncertainties, the true link dynamics are assumed to be:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau - D(q, \dot{q}, t) \quad (3.61)$$

Where $M(q) = M_n(q) + \Delta M(q)$, $C(q, \dot{q}) = C_n(q, \dot{q}) + \Delta C(q, \dot{q})$, $g(q) = g_n(q) + \Delta g(q)$, and $D(q, \dot{q}, t)$ represents the disturbance input.

Therefore,

$$\ddot{q} = M(q)^{-1}(\tau - C(q, \dot{q})\dot{q} - D(q, \dot{q}, t) - g(q)) \quad (3.62)$$

$$\ddot{q}_d = M_n(q)^{-1}(\tau - C_n(q, \dot{q})\dot{q} - g_n(q)) \quad (3.63)$$

where $\ddot{q}_d$ is the desired joint variable derived from the nominal model

Therefore, the error model based only on the mechanical arm becomes:

$$\ddot{e} = -\ddot{q}_d + M(q)^{-1}(\tau - C(q, \dot{q})\dot{q} - D(q, \dot{q}, t) - g(q)) \quad (3.64)$$

Considering the complete dynamic model for the joint torque control, the nominal model becomes equation 3.57. The presence of the gears introduces friction, drive train compliance and backlash in the robotic arm joints. In the case of a direct-drive robot, the problems of backlash, friction, and compliance due to the gears are eliminated. However, the coupling among the links becomes significant, and the dynamics of the motors themselves may be much more complex. Under external disturbances and plant

uncertainties, the true mechanical dynamics of the complete torque control model are assumed to be:

$$J\ddot{q} + B\dot{q} = \tau - D(q, \dot{q}, t) \quad (3.65)$$

Where $J = J_{T1} + \Delta J$, $B = B_{T1} + \Delta B$, and $D(q, \dot{q}, t)$ is the disturbance inputs such as unmodeled dynamics.

The model can be represented as:

$$(J_{T1} + \Delta J)\ddot{q} + (B_{T1} + \Delta B)\dot{q} = \tau - D(q, \dot{q}, t)$$

Hence,

$$\ddot{q} = (J)^{-1}(\tau - B\dot{q} - D(q, \dot{q}, t)) \quad (3.66)$$

Ignoring the uncertainty, the model becomes

$$\ddot{q}_d = (J)^{-1}(k_t I - B\dot{q}) \quad (3.67)$$

The difference between the desired $\ddot{q}_d$ and actual joint variables $\ddot{q}$ is the error in the robotic manipulator model. The uncertainty model or error model (e) based on the robotic arm plus actuator dynamics model becomes:

$$e = -\ddot{q}_d + (J)^{-1}(\tau - B\dot{q}) \quad (3.68)$$

3.9 Robust Controller Design

The second problem in the robotic manipulator development is the control problem due to effects of the uncertainties. Solving the problem involves designing a controller that can cancel the effects of the uncertainties by generating a control signal input into the actuators. To achieve a robust control, the system must sense its output through feedback control technique in order to compare the perturbed output with the desired output. Due to the presence of uncertainties in physical systems, the output is

termed perturbed. The effects of the possible perturbations in the system can be canceled in the robust system with the introduction of controller gains to compensate the error signal. The design objective here is to choose the compensator in such a way that the plant (robotic manipulator joint) output “tracks” or follows a desired output, given by the reference signal. The control signal, however, is not the only input acting on the system. Considering the robotic manipulator in a real environment in figure 3.9, with uncertainties the inputs to the system becomes the reference input r , and uncertainty inputs: the disturbance D , and measurement noise N .

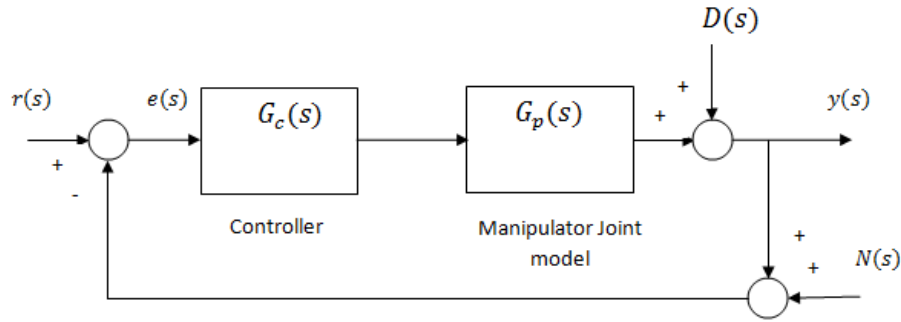


Figure 3.9: Control system with disturbance and noise inputs in real environment

The general transfer function of the feedback controlled system is represented as follows:

$$Y(s) = \frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}(R(s) - N(s)) + \frac{1}{1+G_c(s)G_p(s)}D(s) \quad (3.69)$$

$$Y(s) = \frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}R(s) - \frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}N(s) + \frac{1}{1+G_c(s)G_p(s)}D(s) \quad (3.70)$$

$$E(s) = \frac{1}{1+G_c(s)G_p(s)}(R(s) - D(s) + N(s)) \quad (3.71)$$

$$E(s) = \frac{1}{1+G_c(s)G_p(s)}R(s) - \frac{1}{1+G_c(s)G_p(s)}D(s) + \frac{1}{1+G_c(s)G_p(s)}N(s) \quad (3.72)$$

From equation 3.69, the following functions are derived

$$T(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}$$

$$S(s) = \frac{1}{1 + G_c(s)G_p(s)}$$

$$Lg(s) = G_c(s)G_p(s)$$

$T(s)$ (i.e. complementary sensitivity function) is the transfer function between the output and the reference input of the system through the feedback. $S(s)$ (i.e. Sensitivity function) is the transfer function between the output and disturbances of a system. $Lg(s)$ is the open loop function.

The robust controller design and analysis will be based on the three functions T , S and Lg . T is also called closed loop transfer function. A robust system must have the following characteristics:

- i. Good set-point or reference tracking
- ii. Good disturbance rejection
- iii. Good noise suppression
- iv. Robust stability

3.9.1 Robust Controller Design Objectives/Specifications

The following objectives must be met for the system to achieve full robustness characteristics.

- i. For good set-point tracking $|T(j\omega)|$ must follow the zero gain line at low frequencies. This can also be achieved if $|G_c(j\omega)G_p(j\omega)| \gg 1$ i.e., Loop gain transfer function must be very much greater than one at low frequencies.
- ii. For good disturbance rejection, $S(s)$ or $|S(j\omega)| \ll 1$, i.e., sensitivity must be very much less than one. This can be achieved if

$|G_c(j\omega)G_p(j\omega)| \gg 1$ i.e., Loop gain transfer function must be very much greater than one at low frequencies.

If $|S(j\omega)| = 0$ yields perfect output disturbance rejection

- iii. For good noise suppression, $|T(j\omega)| \ll 1$ at frequencies of noise i.e., $T(s)$ must be very much less than zero at high frequencies.

3.10 Robustness Analysis

This involves the examination of control system design to understand the system behavior considering the uncertainties and changes the system may face in real environment. The areas of interest include the reduction of sensitivity to model uncertainties, disturbance rejection, measurement noise attenuation, steady state errors and transient response characteristics (Dorf and Bishop, 2008). This will involve the use of some mathematical models such as Bode plot and reference tracking to analyze the system for stability, performance and robustness. The transient response is the output response of the system as a function of time and it must be adjusted (through the controller) to be satisfactory in order to achieve desired goal of the control system design. The robust controller design is based on shaping the sensitivity and complementary sensitivity transfer functions graphs to the desired shape. The singular value plot for S and T for robustness analysis in (Nair, 2011) was simplified and modified in figures 3.10a and 3.10b.

3.10.1 Sensitivity/Tracking Error Signal

The sensitivity of a control system to parameter variations is very important. A main advantage of a closed-loop feedback system is its ability to reduce the system's sensitivity. Robustness is the low sensitivity of the controlled system to effects that are not considered in the analysis and design phase such as disturbances, measurement noise and unmodeled dynamics. The system should be able to withstand these uncertainty effects when performing its operations. The relationship between complementary

sensitivity function $T(s)$ and sensitivity function $S(s)$ of the closed-loop controlled robot manipulator are as follows:

$$T(s) + S(s) = 1 \quad (3.73)$$

The ability of the controlled system output to track the input to the system determines the performance of the system. The tracking error e , of the closed-loop control system can be related to the reference input $r(s)$ and the actual output $y(s)$ of the controlled system as follows:

$$E(s) = Y(s) - R(s)$$

One of the objectives in designing a control system is that the controlled system's output should exactly and instantaneously reproduce its input (Dorf and Bishop, 2008). This implies that $y(s) = r(s)$. Hence, error $e(s)$ will tend to zero.

The function $Lg(s)$, is known as the loop gain and it plays a fundamental role in control system design and analysis. In terms of the loop gain $Lg(s)$, tracking error $e(s)$ function becomes:

$$E(s) = \frac{1}{1+Lg(s)}R(s) - \frac{1}{1+Lg(s)}D(s) + \frac{1}{1+Lg(s)}N(s) \quad (3.74)$$

The magnitude of the loop gain $Lg(s)$ can be described by considering the magnitude $|Lg(j\omega)|$ over the range of frequencies, ω , of interest. Considering the tracking error, for a given $G_p(s)$, to reduce the influence of the disturbance $D(s)$ on the tracking error $e(s)$, $Lg(s)$ should be made large over the range of frequencies that characterize the disturbances. In that way, the transfer function $1/(1+G_c(s)G_p(s))$ will be small and it implies that the controller $G_c(s)$ (or K) should be designed to have a large magnitude. Conversely, to attenuate the measurement noise, $N(s)$, and reduce the influence on the tracking error, $Lg(s)$ should be made small over the range

of frequencies that characterize the measurement noise. Hence, the transfer function $G_C G_P / (1 + G_C(s) G_P(s))$ will be small, thereby reducing the influence of $N(s)$ and this implies that the controller $G_C(s)$ should be designed to have small magnitude. The conflict that exists in making the controller $G_C(s)$ to be large to reject disturbances and at the same time making $G_C(s)$ to be small to attenuate measurement noise can be addressed in the design phase by making the loop gain, $L_g(s) = G_C(s) G_P(s)$, to be large at low frequencies (associated with frequency range of disturbances), and making $L_g(s)$ small at high frequencies (associated with measurement noise).

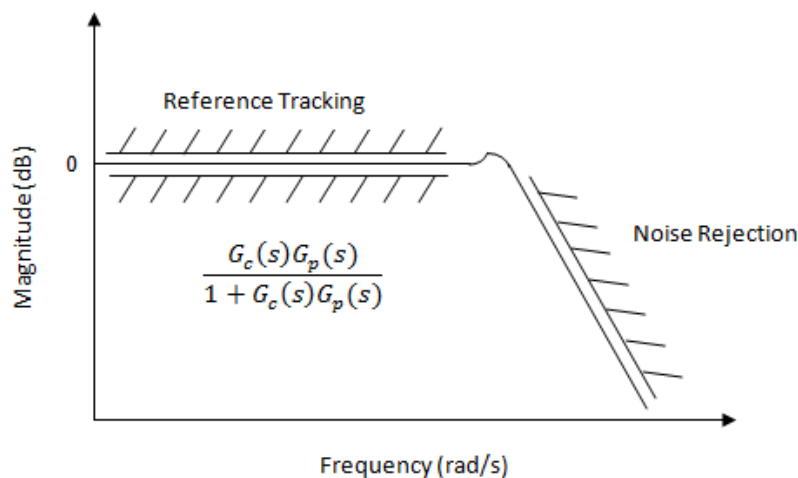


Figure 3.10a: Complementary sensitivity (T) graph

For good reference tracking, the complementary sensitivity graph must follow the zero dB magnitude at low frequencies and for good noise rejection, the graph must reduce very much below zero with sharp slope (roll-off) at high frequencies as shown in figure 3.10a; (Nair, 2011).

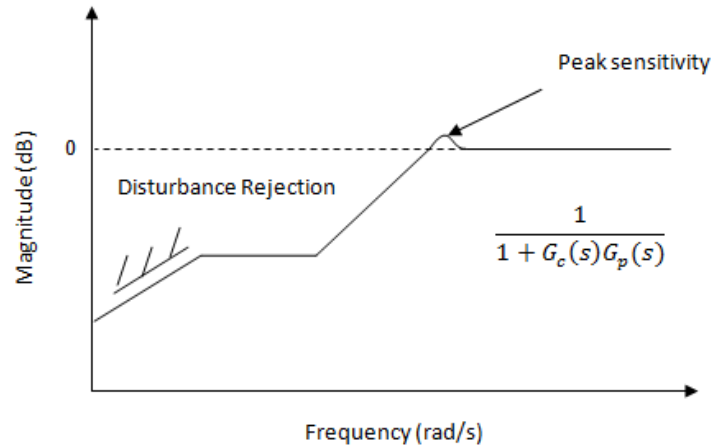


Figure 3.10b: Sensitivity (S) graph

For good disturbance rejection, the sensitivity graph must be very much less than zero as shown in figure 3.10b, and the peak sensitivity must be reduced as much as possible (peak sensitivity of zero is perfect).

The goal of the design should be to minimize the sensitivity and steady state error in order to achieve robustness and optimization of the controlled system. The system should continue to maintain a zero steady state error.

3.10.2 Stability Robustness Analysis

In control system engineering, it is imperative to study the stability of control systems in order to be equipped with the behavior of the system under both steady and transient conditions (Dukkipati, 2006). In order to investigate system stability, Root-locus, Bode and Nyquist plots are applied. Bode plot is used in this work to demonstrate stability of the robot manipulator because it shows more clearly the stability margins: gain margin and phase margin. It also illustrates the stability robustness behavior of the system in the magnitude graph by showing the loop gain levels at low and high frequencies. Stability robustness must be achieved in the design of a controlled system to withstand unforeseen significant uncertainties neglected during the design phase of the robot manipulator.

Gain and phase margins are common terms to describe how stable a system is and the behavior of the system at high frequencies. Gain and phase margins are used more because they are simple and ideal measurements of stability. Gain margin (GM) is the reciprocal of the magnitude when the phase of the open-loop transfer function crosses -180 . Good value of GM $> 5\text{dB}$ and for high robustness GM $\geq 20\text{dB}$. Phase margin (PM) is the difference between the phase angle minus 180 when the magnitude of the open-loop transfer function crosses 0dB . Good value of PM $\geq 40\text{degrees}$. From the illustration in figure 3.10c, the higher the loop gain at low frequencies, the better the performance of the controlled system and the lower the loop gain at high frequencies the lower the sensitivity to sensor noise and model uncertainty i.e., high robustness.

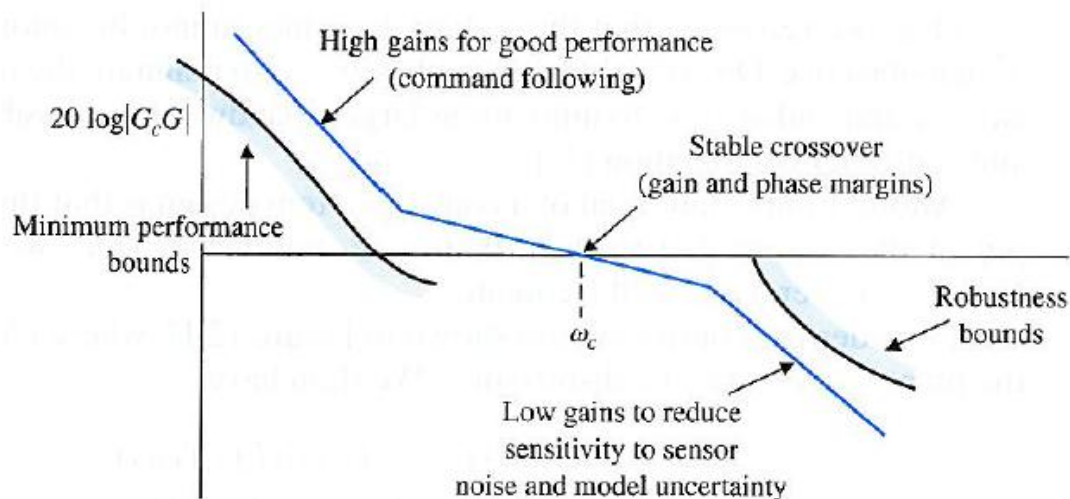


Figure 3.10c: Demonstration of system behavior on Bode plot (Dorf and Bishop, 2008)

3.11. H-Infinity Controller Design

The H-Infinity controller design method here involves the control of the joint models with weights W_1 and W_2 based on the joint parameters. The weight

parameters are varied in order to improve the iteration results. The weighting functions have been chosen according to industrial performance specifications (Filardi et al, 2003):

Wp (W1): the inverse of the weighting function $W_p(s)$ is used to impose a performance specification in terms of the sensitivity function S . W_p is chosen:

$$W_p(s) = \frac{s/M_s + w_b}{s + w_b A_s} \quad (3.75)$$

where M_s is to introduce a margin of robustness on the peak of S , w_b helps to have a sensible attenuation of disturbances and A_s helps to reduce the steady-state position error.

Wu (W2): the control output u is weighted according to the actuator limitations. $W_u(s)$ is set to:

$$W_u(s) = \frac{s + \omega_{bc}/M_u}{s\varepsilon + \omega_{bc}} \quad (3.76)$$

where M_u helps to impose limitations on the maximum value of the controller output signal, w_{bc} helps to limit the effect of measurement noise and plant uncertainties at high frequencies, and ε helps to ensure a high-frequency controller gain.

Proposed H-Infinity Algorithm:

- Establish the joint model $G(s)$ for joint I and II
- Apply weight $W1$ to control the joint sensitivity to disturbance
- Apply moderate control $W2$ on the control signal u
- Ignore the closed loop system (T) control by applying no control, $W3=0$
- Augment or connect the plant $G(s)$ with weighting functions $W1(s)$, $W2(s)$ and $W3(s)$ (design specifications) to form an "augmented plant" $P(s)$

- Apply H-Infinity synthesis for loop shaping and generate K
- Form the loop gain (Lg) = K*P
- Form the system sensitivity function $S = (1+Lg)^{-1}$
- Form T, (1-S)
- Analyze Lg, S and T for performance and robustness of the controlled system

This technique allows very precise loop shaping via suitable weighting strategies and thereby achieves robust control. Augmenting the joints with frequency dependent weights W_1 , W_2 and W_3 , the Matlab script `hinfsyn` will find a controller that "shapes" the signals to the inverse of these weights. The Matlab function `augw` (or `connect`) forms the augmented joint plant function. H-Infinity synthesis technique does not require simulation turning, rather it is achieved in computation using Matlab program codes and the results were noted as the weights were varied. Appendix D presents the program codes for the computation of H-Infinity controller design in Matlabm.file.

3.12. Proportional-Integral-Derivative (PID) Controller Design

The proportional-integral-derivative controller algorithm is derived as follows:

$$U(t) = K_p e(t) + K_I \int e(t) dt + K_D \frac{d}{dt} e(t) \quad (3.77)$$

Applying Laplace transformation;

$$U(s) = K_p e(s) + K_I \frac{1}{s} e(s) + K_D s e(s) \quad (3.78)$$

$$U(s) = (K_p + K_I \frac{1}{s} + K_D s) e(s) \quad (3.79)$$

Hence,

$$G_c(s) = K_p + K_I \frac{1}{s} + K_D s \quad (3.80)$$

Generating the loop gain of the controlled system for robust control analysis:

$$Lg(s) = (K_p + K_I \frac{1}{s} + K_D s) G_p \quad (3.81)$$

Proposed PID Algorithm:

- Establish the joint model $G(s)$ for joint I and II,
- Select the controller gains with the help of PID tuner in MATLAB
- Form the controller model with the gains
- Form the loop gain $(Lg) = K * P$
- Form the system sensitivity function $S = (1 + Lg)^{-1}$
- Form $T, (1 - S)$
- Analyze Lg, S and T for performance and robustness of the controlled system

Appendix E presents the program codes for the computation of PID controller design in Matlabm.file.

3.13. Implementation/Simulation

The implementations involved realization of the 3DOF articulated robotic manipulator in 2D and 3D models, conversion of the 3D model into SimMechanics model and determining the masses of the rigid bodies, determining the total torques at the joints and selecting proper actuators for the respective joints. Other parameters involved in the dynamic modeling and control of the joint torques of the manipulator are to be derived here. The experiment for the determination of total damping coefficient of the joints and the application of the controller design techniques were carried out.

3.13.1 Robotic Manipulator 2D and 3D Structures

The robotic manipulator was designed in autodesk inventor application software using the dimensions as shown in figure 3.11. The figures 3.12a, 3.12b, and 3.12c show different orientations of the robotic manipulator links positions in autodesk inventor software. The purpose of these models is to achieve the physical image of the manipulator, import and convert the 3D model into the SimMechanics model and get the mass of the rigid bodies.

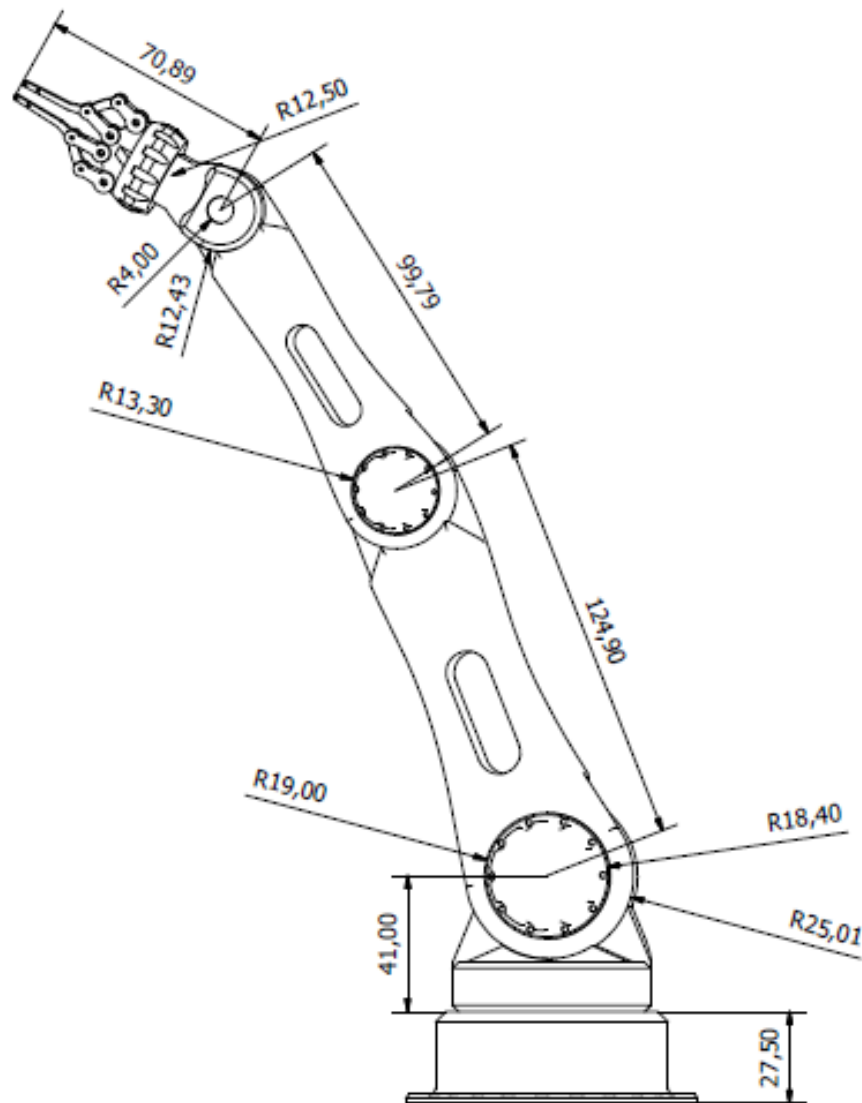


Figure 3.11: Robot arm 2D structure and dimensions

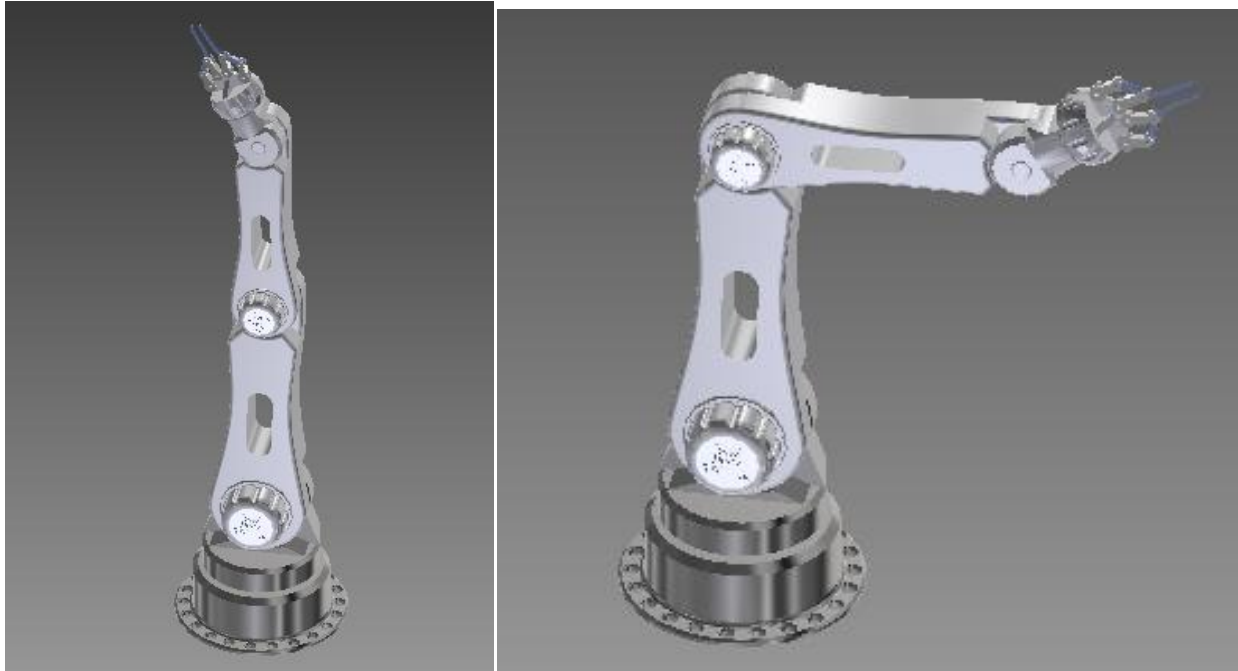


Figure 3.12a: 3D structure of the manipulator showing vertical position of first link, vertical and horizontal positions of the second link.

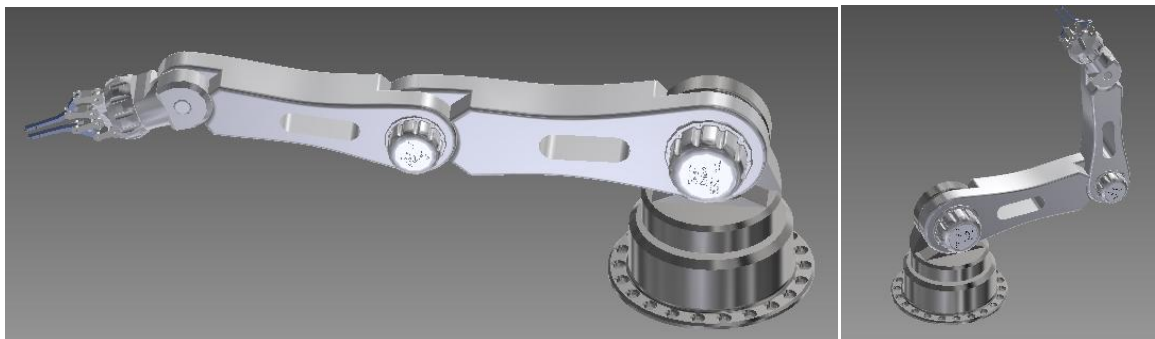


Figure 3.12b: 3D structure of the manipulator showing horizontal positions of first link, horizontal and vertical positions of the second link.

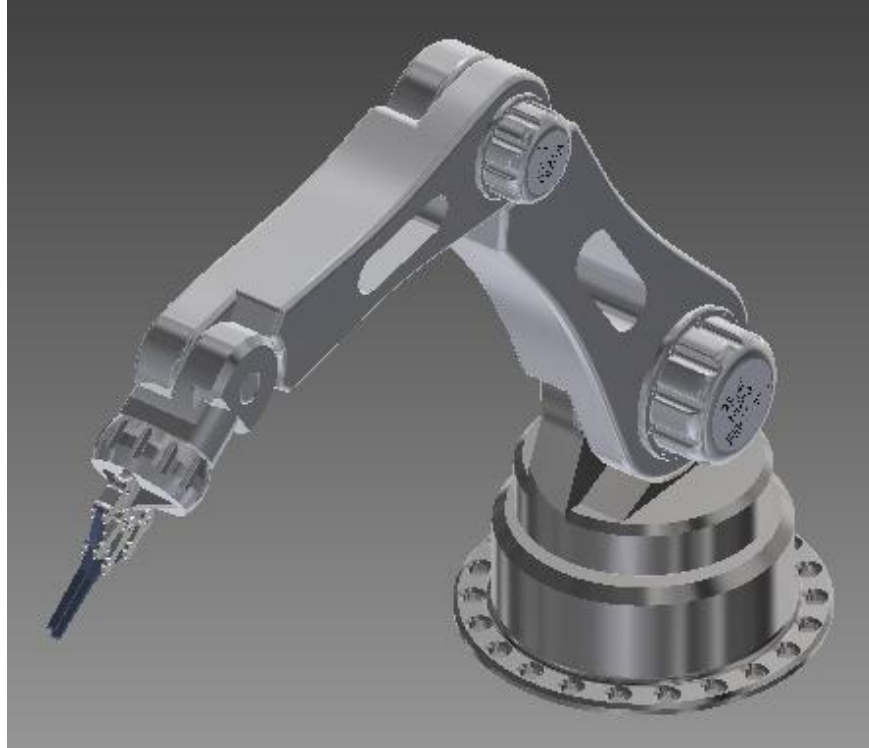


Figure 3.12c: 3D structure of the manipulator showing angular positions

3.13.2 Joint Torque Calculation

The total torques at the manipulator joints are determined as follows applying equations 3.8 and 3.9.

$m_1=1.09\text{kg}$, $m_2=0.058\text{kg}$, $m_3=0.012\text{kg}$, $m_4=0.004\text{Kg}$, $W_1 = 10.9\text{N}$, $W_2 = 0.58\text{N}$, $W_3 = 0.12\text{N}$, $W_4 = 0.04\text{N}$, $L_1 = 1.249\text{m}$, $L_2 = 0.998\text{m}$, $L_{\text{Load}}= 0.7089\text{m}$, $L_3 = 1.3524\text{m}$

M_0 = Base Actuator

M_1 = Actuator 1

M_2 = Actuator 2

Torque at Joint 0: M0

:- 0 (It is not affected by gravity)

Torque at Joint 1: M1

$$\begin{aligned} & \frac{12.49}{2} * 0.109 + 12.49 * 0.004 + \left(12.49 + \frac{9.98}{2}\right) * 0.058 + (12.49 + 13.524) * 0.012 \\ &= 2.0567\text{kg-cm} \\ &= 2056.71\text{g-cm} \end{aligned}$$

Torque at Joint 2: M2

$$\begin{aligned} & \frac{9.98}{2} * 0.058 + 13.524 * 0.012 \\ &= 0.4517\text{kg-cm} \\ &= 451.7\text{g-cm} \end{aligned}$$

3.14Parameter Calculations for Joints Design

The implementation of the proposed robust control in the development of robotic manipulator is based on the joint torque control. Considering the uncertainties in the physical system, the torque of the actuator selected must be greater than calculated torque at each of the joints. The difference between the selected actuator torque and the calculated torque will help to counteract the effects as a result of other unmodeled torques.

The purpose of joint design is to select the proper joint actuator and derive the parameters needed to complete the joint model. From the joint torque calculations above, the calculated torques at the two joints that require lifting and the selected actuators for the respective joints are shown in table 3.2.

Table 3.2: Calculated and Selected Torques for the Joints I and II

Joint	Calculated Torque (g-cm)	Selected Actuator Torque(g-cm)
I	2056.7	2200
II	451.7	500

With the calculated joint torques, the joint actuators were selected from the actuator step motor manufacturer's catalog of NMB Corporation and the selected motors are shown in Appendix B.

In order to derive other parameters of the joint actuator such as the torque constant k_t , and electromotive constant K_e , for the joint analysis, the torque and inertia of the actuators were converted to the standard units. Table 3.3 shows the parameter values of joints I and II.

Joint I:

Actuator I Torque (τ) = 2200g-cm = 0.022Kg-m = 0.2157N.m

Actuator Inertia (J_{Actuator}) = 56g-cm² = 0.0000056Kg-m²

To calculate the inertia generated by the arm at joints 1

Inertia of link1 (J) = ($m_1 \times (L_1)^2$)/2

$$J_{l1} = \frac{m_1 * L_1^2}{2}$$

$$J_{l1} = \frac{0.109 * 0.1249^2}{2}$$

=0.0008502Kg-m²

Inertia of link II

$$J_{l2} = \frac{m_2 * L_2^2}{2}$$

$$J_{l2} = \frac{0.058 * 0.098^2}{2}$$

$$= 0.0002889 \text{Kg-m}^2$$

m_3 = Mass of load3 = Mass of load = 0.012Kg

L_3 = Length of load = 7.089cm = 0.07089m

$$J_{Load} = \frac{m_3 * L_3^2}{2}$$

$$J_{Load3} = \frac{0.012 * 0.07089^2}{2}$$

$$= 0.0000302 \text{Kg-m}^2$$

Total load inertia at joint1 = $J_{l1} + J_{l2} + J_{Load}$

$$= 0.0008502 + 0.0002889 + 0.0000302$$

$$= 0.0012 \text{Kg-m}^2$$

Total Inertia (J) at joint1 = $J_{Actuator} + J_l$

$$= 0.0000056 + 0.0012$$

$$= 0.0012056 \text{Kg-m}^2$$

Torque constant (Km or kt) = Torque/Current

$$=0.2157/1.2$$

$$=0.1798\text{N.m/A}$$

$$\text{Electromotive force constant } (K_e) = K_t = 0.1798\text{V.s/rad}$$

Joint II:

$$\text{Actuator Torque } (\tau) = 500\text{g-cm} = 0.005\text{Kg-m} = 0.049\text{N.m}$$

$$\text{Actuator Inertia } (J_{\text{Actuator}}) = 13\text{g-cm}^2 = 0.0000013\text{Kg-m}^2$$

$$\text{Total load inertia at joint II} = J_{I2} + J_{\text{Load3}}$$

$$= 0.0002889 + 0.0000302$$

$$= 0.0003191\text{Kg-m}^2$$

$$\text{Total Inertia } (J) \text{ at joint II} = J_{\text{Actuator}} + J_{I2}$$

$$= 0.0000013 + 0.0003191$$

$$= 0.0003204\text{Kg-m}^2$$

$$\text{Torque constant } (K_t) = \text{Torque/Current}$$

$$=0.049/0.9$$

$$=0.0544\text{N.m/A}$$

$$\text{Electromotive force constant } (K_e) = K_m = 0.0544\text{V.s/rad}$$

Taking $r_m = r_l$

Table 3.3: The parameters of designed joints I and II

Parameters	Joint I	Joint II
Inertia (J)	0.0012061Kg-m ²	0.0003204Kg-m ²
Resistance (R)	3Ω	4.4Ω
Inductance (L_a)	0.0044H	0.0016H
Current (i)	1.2A	0.9A
Torque Constant (km)	0.1798N.m/A	0.0544N.m/A
Electromotive Force Constant (Ke)	0.1798V.s/rad	0.0544V.s/rad

3.15 Determination of JointDamping Coefficients

Damping which according to Mizrahi (2015) is important in determining the performance of many tasks assigned to the links and in counteracting undesired effects of applied loads and disturbances. Damping coefficient gives a force based on how fast the joint is moving. It is used to control the speed of the joint such that it can balance or reject the effects of disturbances.

Shamrao et al(2013) stated thatviscous damping coefficient of the robotic joints has to be found from experiment for complete dynamic modeling of the robotic joint. In this work, the open loop gain plot of the plantwas used in MATLAB to determine the damping coefficient for stability.

The first stage of the simulation is to determine the damping coefficient such that the system will be stable, $|G_p(j\omega)| \gg 0$ at low frequencies and $|G_p(j\omega)| \ll 0$ at high frequencies. In Dorf et al (2008), it was presented that

a system with loop gain of $|G_p(j\omega)| \gg 0$ at low frequencies will achieve good performance and with loop gain of $|G_p(j\omega)| \ll 0$ at high frequencies it will achieve good robustness. Appendix C shows the damping coefficient experiment codes computed in Matlab.

Stability criteria for joint design:

- Peak gain $\gg 0$
- Both Gain Margin (GM) and Phase Margin (PM) must exist
- PM should be greater than GM

3.16 Robotic Manipulator SimMechanics Model

In order to demonstrate the need for independent joint control of the robotic manipulator, the SimMechanics model is employed. Actually, the model uses the Lagrangian principle of robotic manipulator modeling whereby, the system is designed based on the parameters such as the inertia matrix and Coriolis vectors without considering the dynamic model of the actuator. The SimMechanics model of the robotic arm is generated from the 3D model of Autodesk Inventor Professional by importing the 3D model vectors from Autodesk Inventor into Matlab/Simulink environment. The SimMechanics model represents an equivalent model of the 3D Autodesk Inventor design of the robotic manipulator. The purpose of this method is to obtain an equivalent model of the 3D design of the robotic manipulator which can be controlled by adding compensators to the model. However, in this model, the controller output is meant to be connected to the inputs of the revolute joints. This technique helps to study the performance of the robotic manipulator designed without considering the parameters of the joint actuators. The figure A1 in appendix A shows the equivalent SimMechanics models of the 3D robotic manipulator designed in Autodesk Inventor and

figure Ai shows a practical application of feedback control based on the independent joint control law. Table 3.4 shows the description of some SimMechanics components.

Table 3.4: Description of some SimMechanics components

Name	Description
Env	The block Machine Environment defines environment for calculation of the scheme. Each SimMechanics model contains one such block that is connected with the block Ground. Except of inputting the precision of calculation and parameters of the environment, the required analysis type can be set up: • Forward Dynamics – based on initial values and forces in the system the program calculates values of positions and speeds. • Linearization – this mode calculates the system linear model. • Trimming – finds the machine steady states. • Inverse Dynamics for open loop In this mode the SimMechanics calculates forces necessary for performing the motion forced by kinematic excitation. • Kinematics does the same for closed loop systems by including extra internal invisible constraints arising from those structures.
Ground	The Ground block represents a fixed point having infinite mass. At least one block Ground connected with the Machine Environment must be involved.
Body	The block Body in SimMechanics replaces all fixed rigid

	bodies among which the degrees of freedom are added. The bodies are defined by their final and non-zero masses, inertia, positions, directions, and by coordinate systems that are connected to them.
Weld	The blocks Joints interconnect blocks of the Body type and they are added degrees of freedom. The blocks determine direction and type of motion. In difference to the physical joints, in SimMechanics they present massless bodies and a physical connection of the bodies is not required. In the block there are shown ports: B-Base and F-Follower which means, the Follower performs a motion regarding to the Base. The block Weld – means a body without any degree of freedom
Revolute	The Revolute block from the Joints group represents one degree of freedom (rotation)
Driver Actuator	Sensors and Actuators are the blocks used as interfaces between non- SimMechanics Simulink blocks and SimMechanics blocks. By the Actuators it is possible to transform a Simulink signal into physical one actuating the bodies in SimMechanics diagram. The Sensors perform reverse functions – they transform signal from SimMechanics into Simulink environment. These blocks can be connected to Joints, Drivers and Constraints (only Sensors) into special purpose- oriented ports. In Bodies they are connected directly to the chosen CS. Outputs from the sensors are: positions, speeds, accelerations, reaction forces, etc.
Parallel Constraints	The Parallel Constraint block ensures that vectors of axes of two bodies are parallel

However, SIMMECHANICS does not provide the means for complete modeling of the actuator; hence it can easily suffer from constraints due to the presence of unmodeled parameters. Secondly, it does not provide any means to input the electrical parameter configurations of the motors, therefore the physical model of the joints will vary greatly when compared to the mathematical model of the system.

3.17 Simulink Model of the Joint Model

Since the SimMechanics does not provide a means of modeling the actuator parameters based on the physical features of the motor used at the joints, the SIMULINK model is employed to model the actuator parameters for proper implementations of the robust control through independent joint control. The manipulator joint was modeled in two stages connected together; the actuator electrical dynamics and the actuator plus rigid-body mechanical dynamics. The angular acceleration $\frac{d^2q}{dt^2}$ is equal to (I/J) multiplied by the sum of $K_t i$ and $-B \frac{dq}{dt}$. Similarly, the derivative of the current is equal to $1/L_a$ multiplied by the sum of three terms; $-Ri$, V and $-K_e \frac{dq}{dt}$. Hence, the joint model was realized in SIMULINK as illustrated in figure 3.13. Figure 3.14 shows the simulation model of the PID controlled joint.

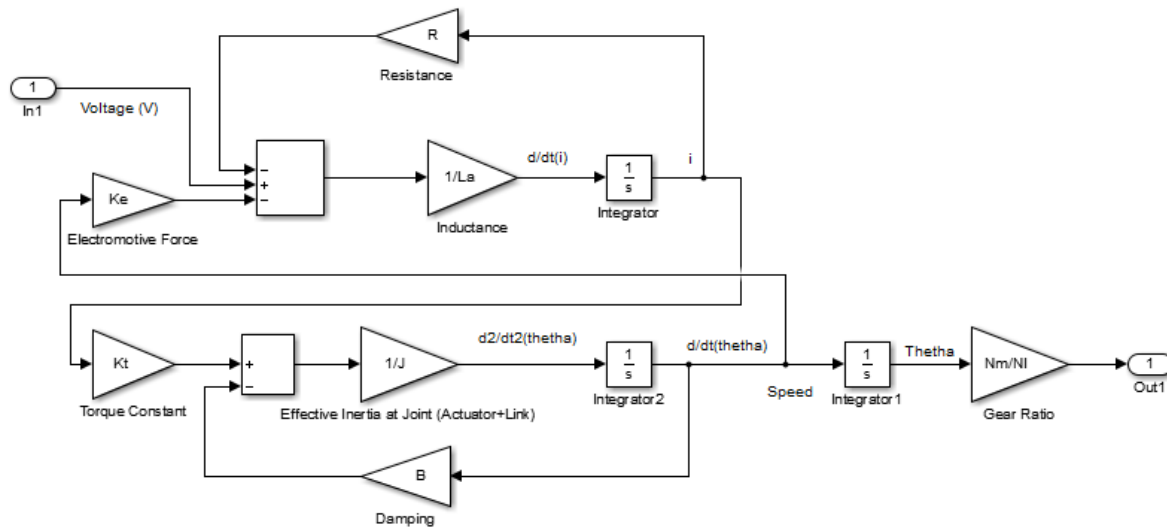


Figure 3.13: Manipulator Jointsimulinkmodel

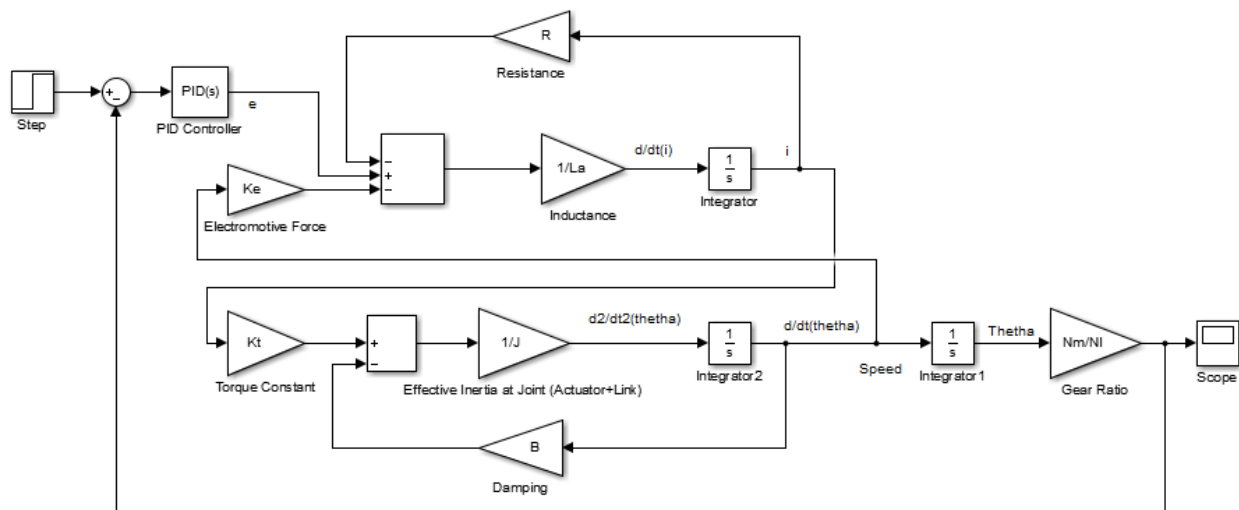


Figure 3.14: Manipulator Jointsimulinkmodel with PID controller

CHAPTER FOUR

RESULTS AND DISCUSSIONS

4.1. Robot Arm DynamicsComputation Results

The results of the computation of the rigid-bodies based on the Lagrange-Euler equation of robot arm model (equation 3.37) in appendix F are as shown in figures 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6. Figures 4.1 and 4.2 show the results of the angular velocity and position against time for joints I and II with the initial values of: $q_1=30$, $q_2=30$, $\dot{q}_1=\dot{q}_2=0$. Figures 4.3 to 4.6 show the angular positions of joints I and II with varied initial conditions.

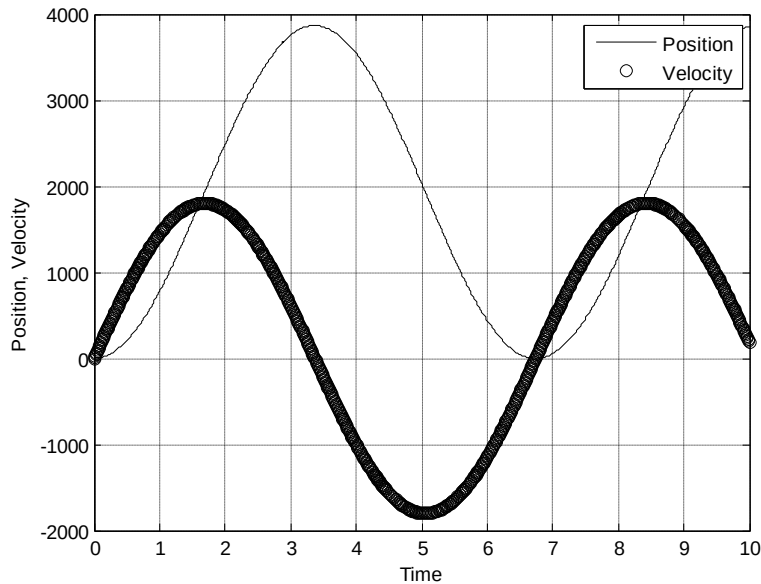


Figure 4.1: Examining velocity and position of the robot arm for joint I with initial values; $q_1=30$, $q_2=-30$, $\dot{q}_1=\dot{q}_2=0$

Figure 4.1 examines the movement of the link I and that of its velocity. Since they changed with time, then the computation is acceptable.

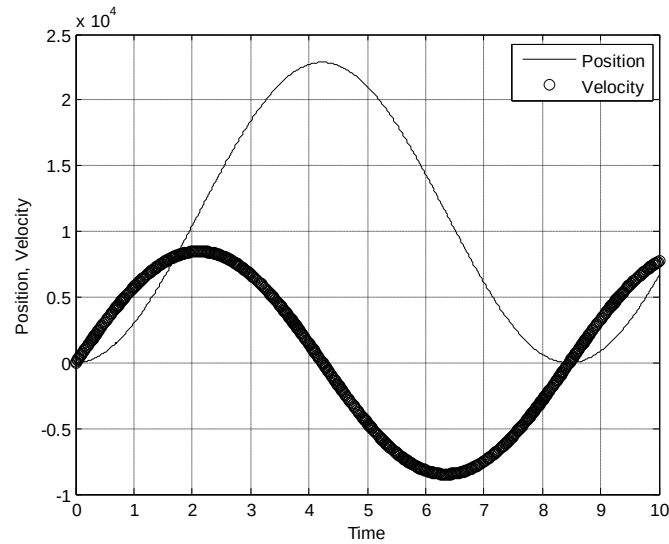


Figure 4.2: Examining velocity and position of the robot arm from joint II with initial values; $q_1=30$, $q_2=-30$, $\dot{q}_1=\dot{q}_2=0$

Figure 4.2 examines the movement of the link I and that of its velocity. Since they changed with time, then the computation is acceptable.

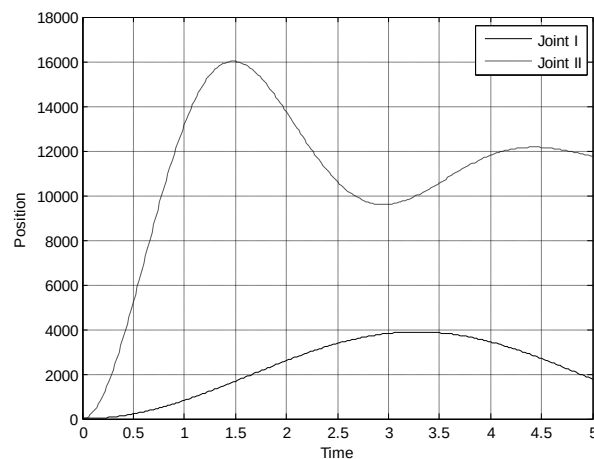


Figure 4.3: Comparing the positions of Joints I and II of the robot arm with initial values; $q_1=30$, $q_2=-30$, $\dot{q}_1=\dot{q}_2=0$

Figure 4.3 shows that with the initial positions, joint I and II are not going on same graph because Joint II adds to joint I position for its own position.

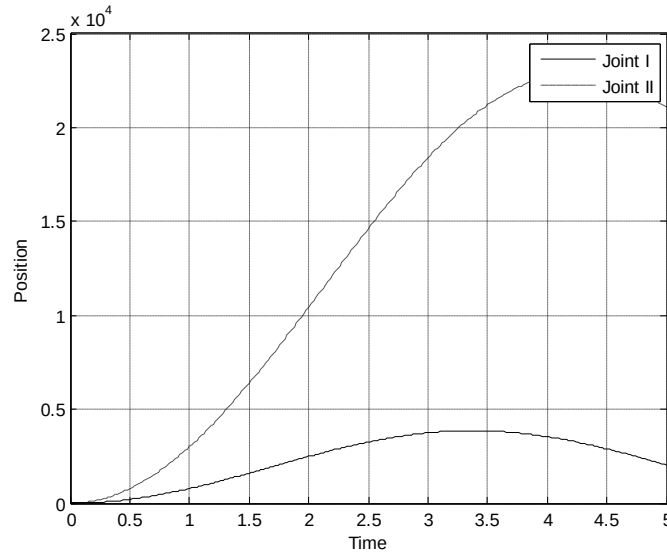


Figure 4.4: Comparing the positions of Joints I and II of the robot arm with initial values; $q_1=30$, $q_2=30$, $\dot{q}_1=\dot{q}_2=0$

Figure 4.4 shows that with the initial positions, joint I and II are not going on same graph because Joint II adds to joint I position for its position.

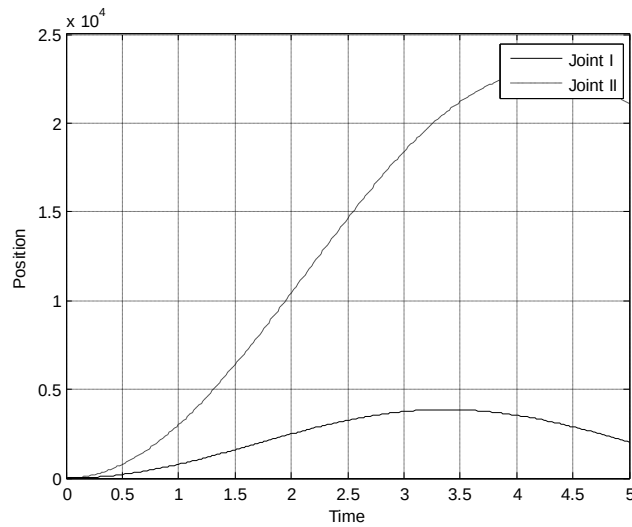


Figure 4.5: Comparing the positions of Joints I and II of the robot arm with initial values; $q_1=-30$, $q_2=-30$, $\dot{q}_1=\dot{q}_2=0$

Figure 4.5 shows that with the initial positions, joint I and II are not going on same graph because Joint II adds to joint I position for its own position.

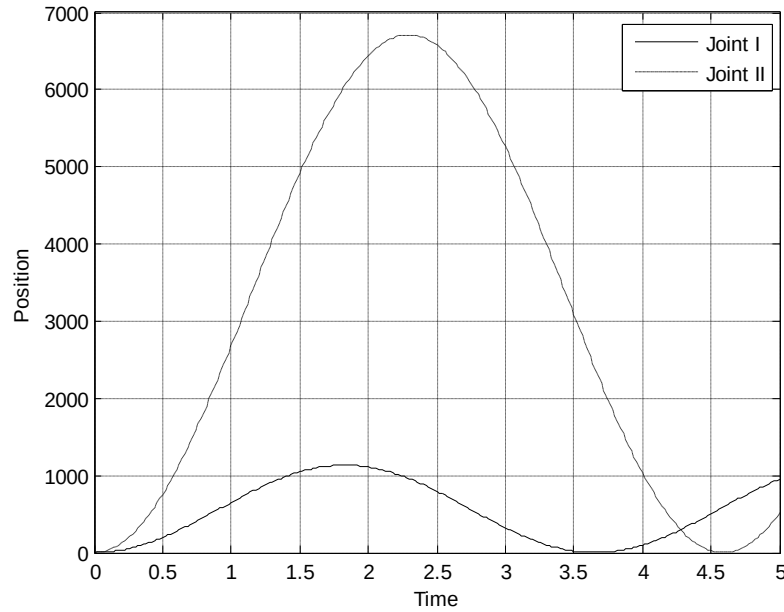


Figure 4.6: Comparing the positions of Joints I and II of the robot arm with initial values; $q_1=45$, $q_2=45$, $\dot{q}_1=\dot{q}_2=0$

Figure 4.6 shows that with the initial positions, joint I and II are not going on same graph because Joint II adds to joint I position for its position.

4.2. Damping Coefficient Results for Joint I and II

The results in figures 4.7, 4.8, 4.9, 4.10 and 4.11 illustrate the bode plots of joint 1 model loop gain to determine the joint model damping coefficient for the best performance. The figures examine the gain margin, phase margin and the peak gain of G for the values of K . In order to make clearer analysis, the Bode plots of the joint 1 model were compared in figure 4.12 for different values of damping coefficient.

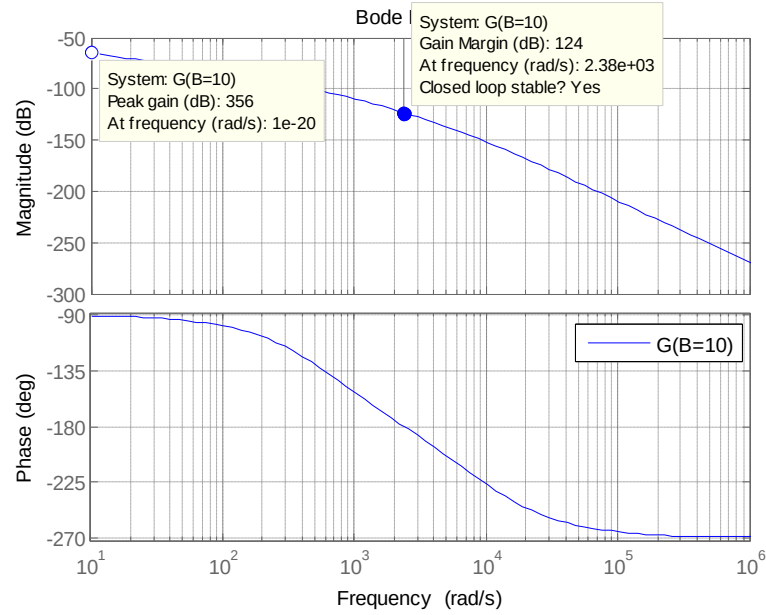


Figure 4.7: Bode plot for Joint 1 model G when $B = 10$

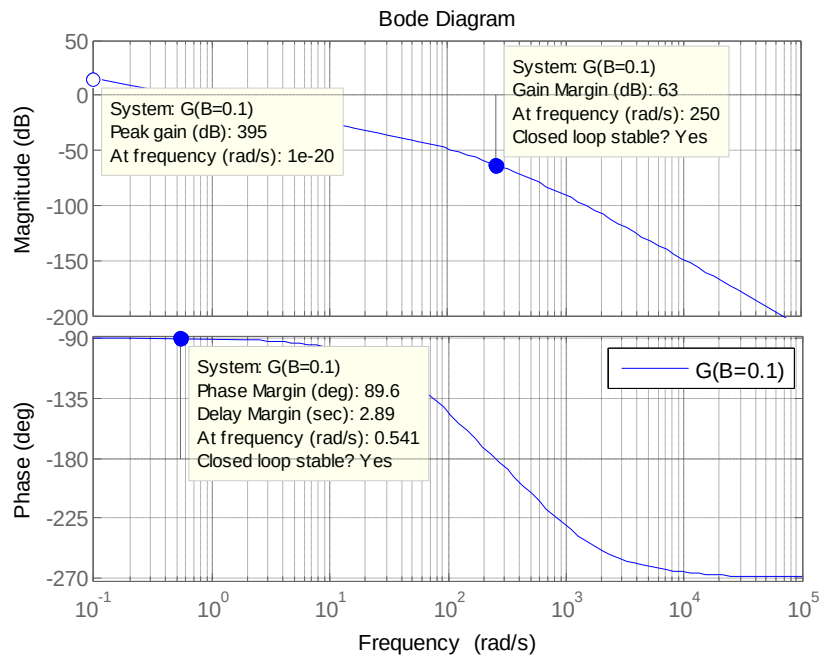


Figure 4.8: Bode plot for Joint 1 model G when $B = 0.1$

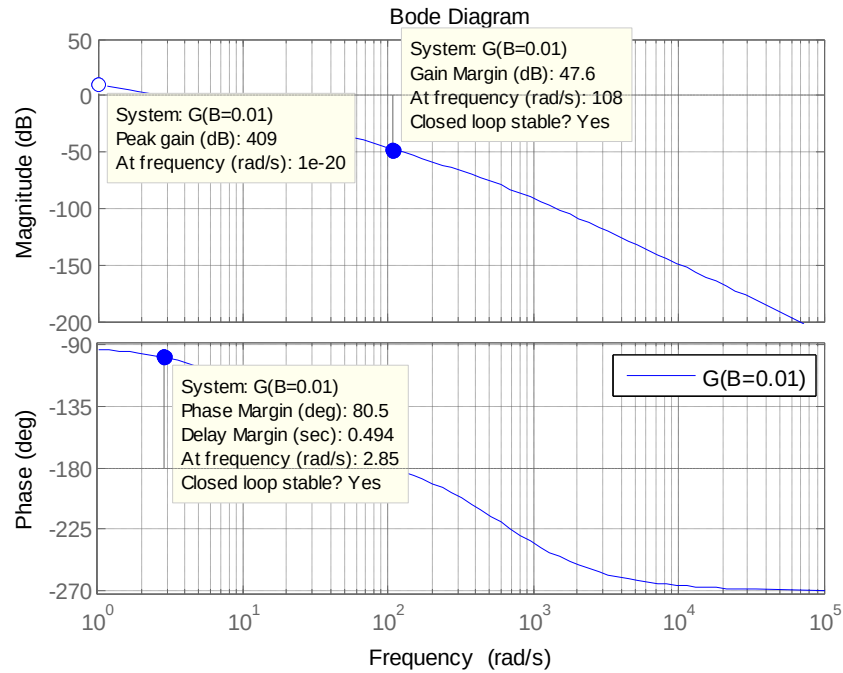


Figure 4.9: Bode plot for Joint 1 model G when B = 0.01

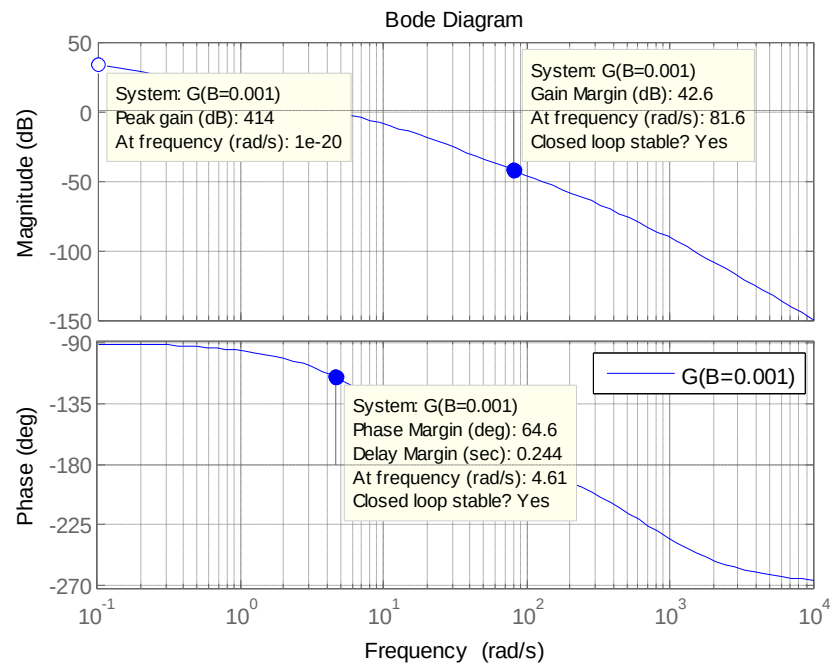


Figure 4.10: Bode plot for Joint 1 model G when B = 0.001

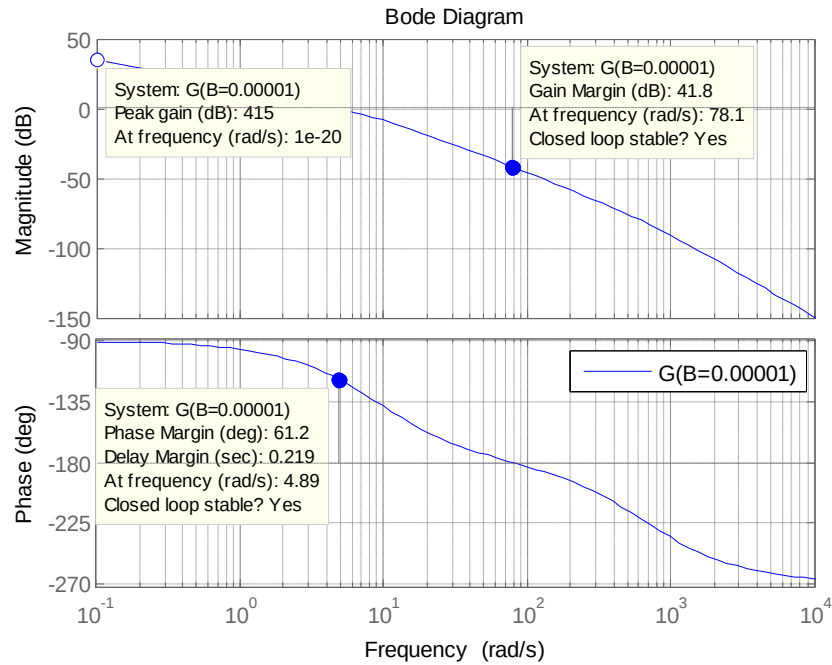


Figure 4.11: Bode plot for Joint 1 model G when $B = 0.00001$

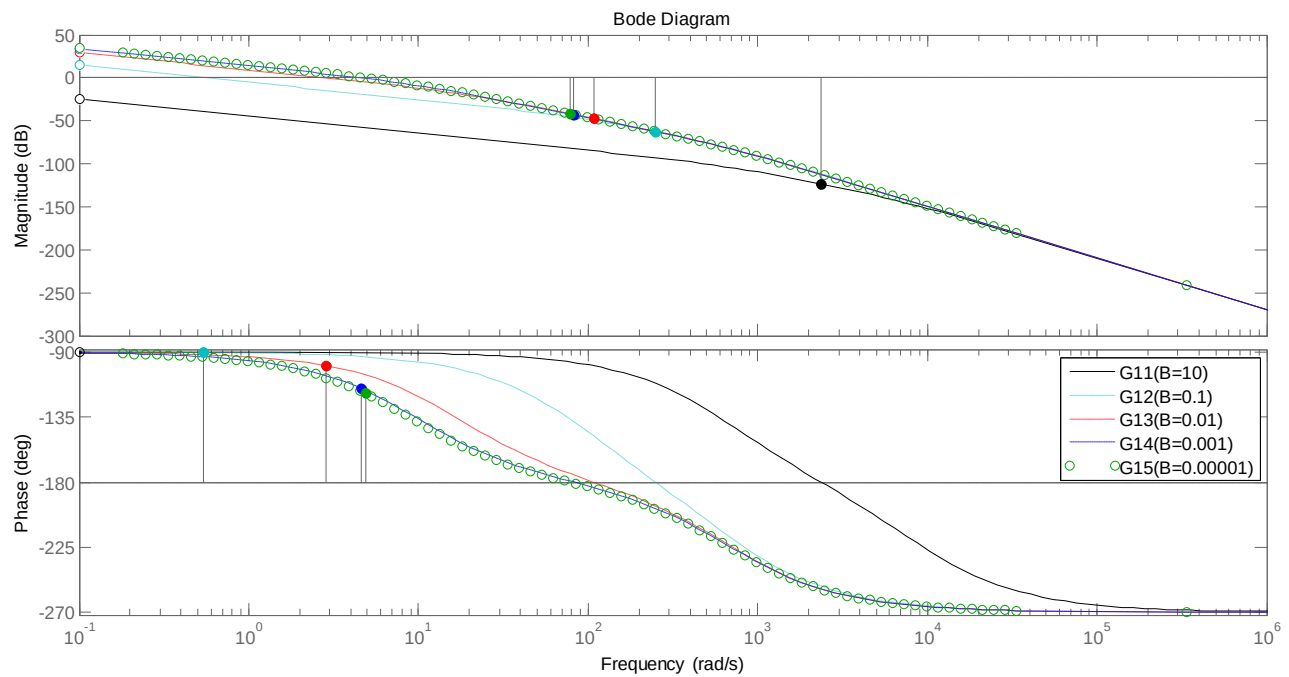


Figure 4.12: Comparing the Bode plots of the joint 1 model for B values

Table 4.1: Damping Coefficient experiments for joint 1

Damping Coefficient (B)	Peak Gain	Gain Margin (GM)	Freq. at GM	Phase Margin (PM)	Freq. at PM
10	356	124	2380	-	-
0.1	395	63	250	89.6	0.541
0.01	409	47.6	108	80.5	2.85
0.001	414	42.6	81.6	64.6	4.61
0.00001	415	41.8	78.1	61.2	4.89

From the table 4.1, with the damping coefficient $B = 10$, the joint model achieved a high peak gain and gain margin. However, it did not record a phase margin value because the gain did not cross the 0dB line. When $B=0.1$, 0.01, 0.001, and 0.00001, the system met the design stability criteria. However, the system achieved highest value of peak gain at low frequencies when $B = 0.00001$, hence it was most accepted.

The results in figures 4.13, 4.14, 4.15, 4.16 and 4.17 illustrate the bode plots of joint 2 models loop gain to determine the joint model damping coefficient for the best performance. Figure 4.18 compares the Bode plots of the joint II model for B values.

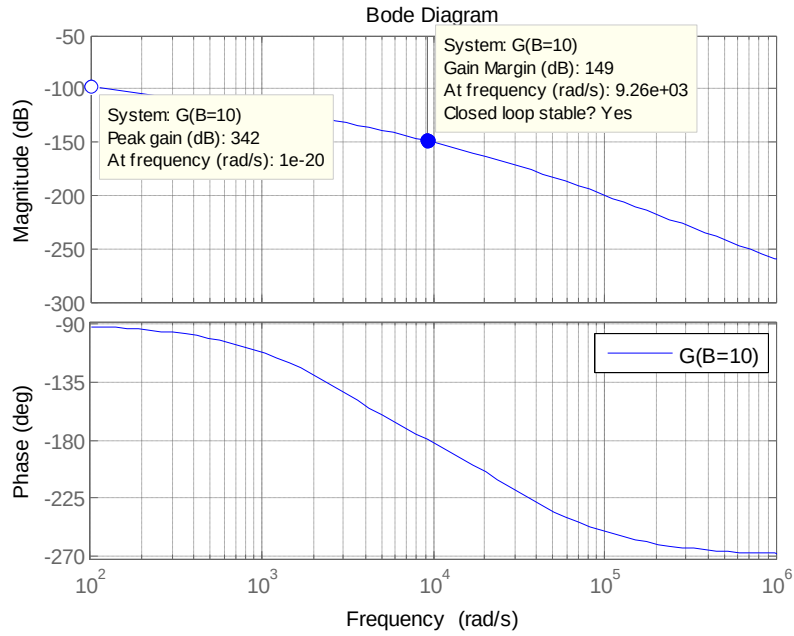


Figure 4.13: Bode plot for Joint II model G when $B = 10$

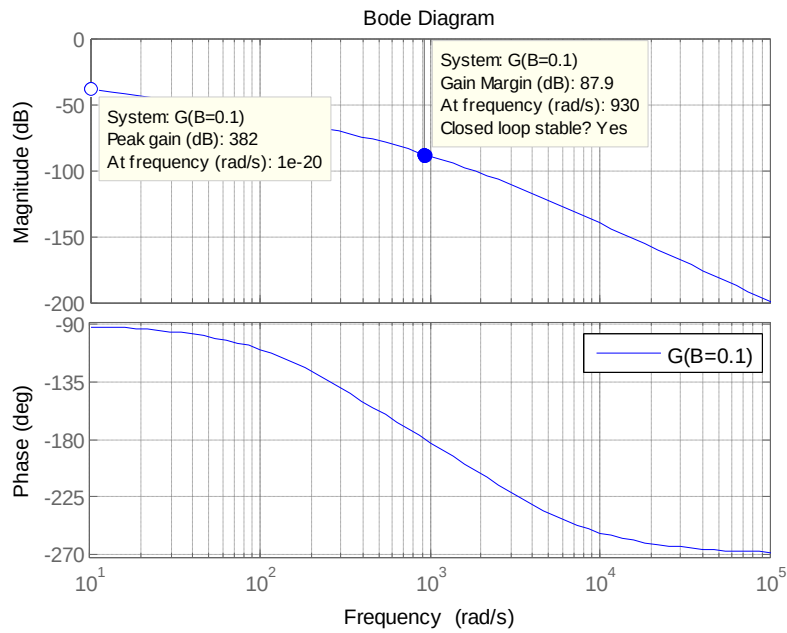


Figure 4.14: Bode plot for Joint II model G when $B = 0.1$

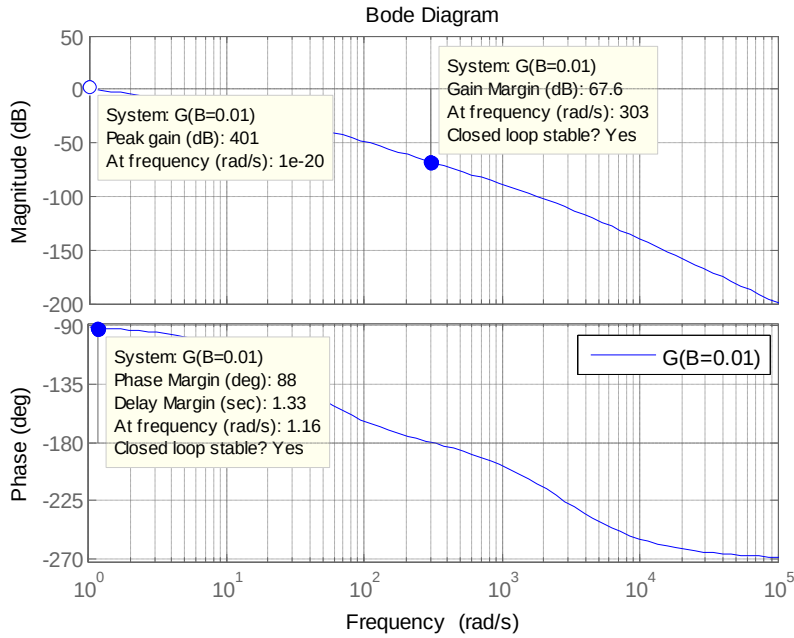


Figure 4.15: Bode plot for Joint II model G when $B = 0.01$

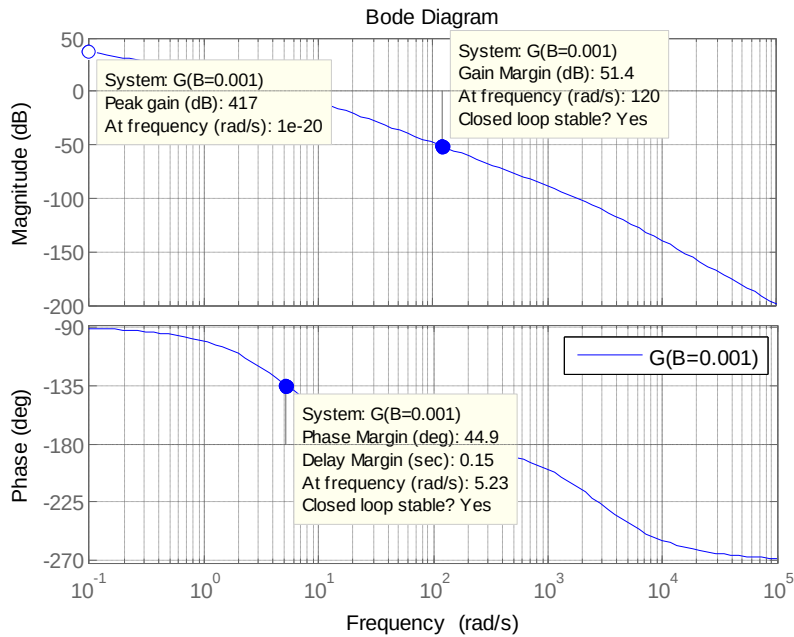


Figure 4.16: Bode plot for Joint II model G when $B = 0.001$

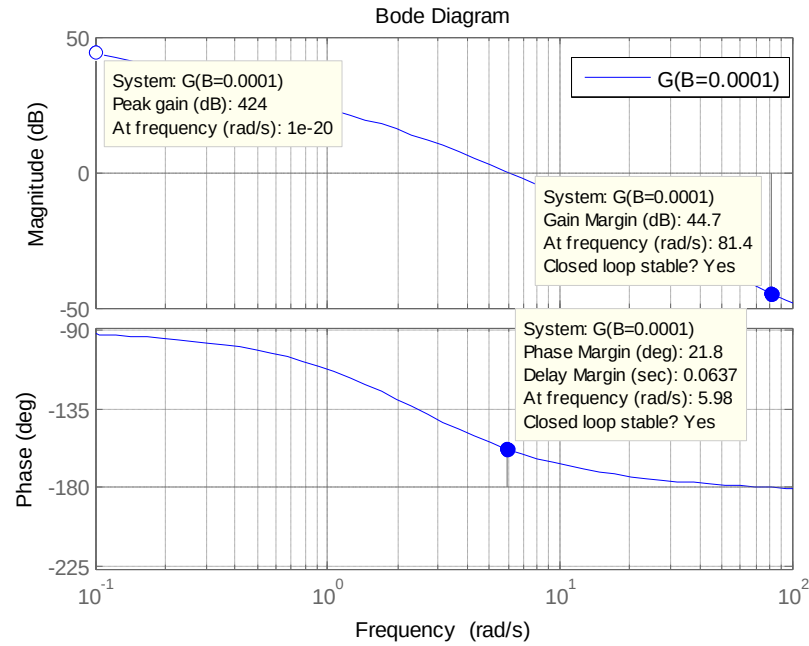


Figure 4.17: Bode plot for Joint II model G when $B = 0.0001$

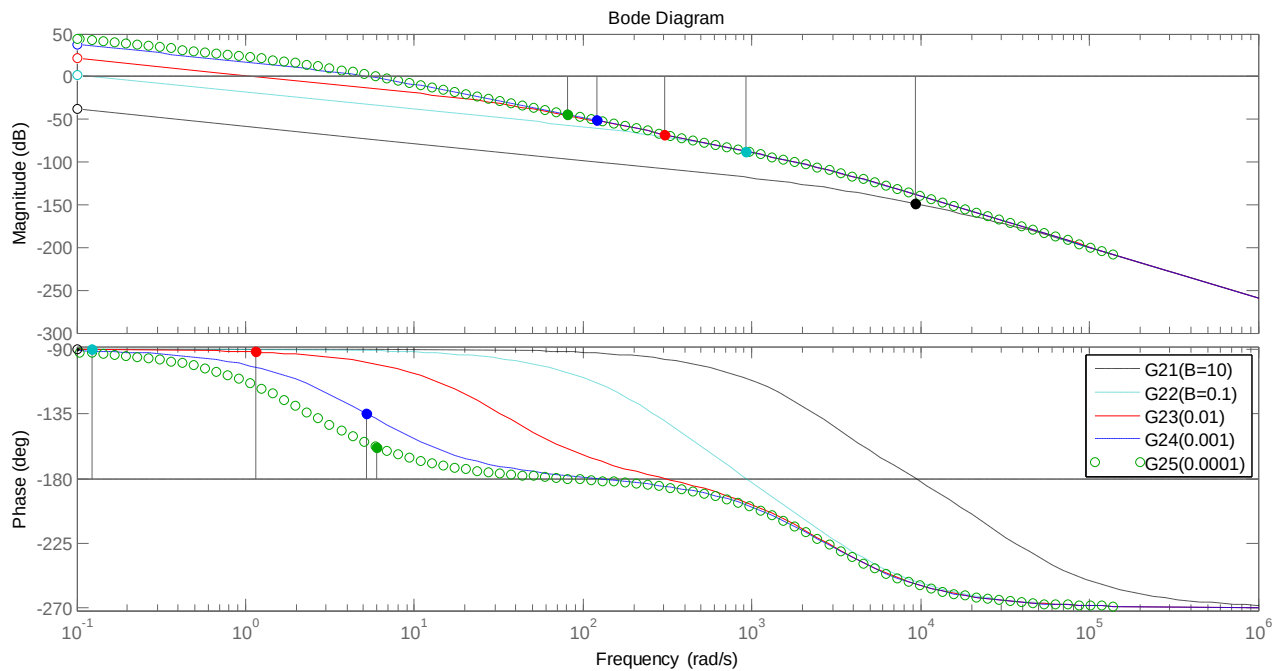


Figure 4.18: Comparing the Bode plots of the joint II model for B values

Table 4.2: Damping coefficient experiment results for joint II

Damping Coefficient (B)	Peak Gain	Gain Margin (GM)	Freq. at GM	Phase Margin (PM)	Freq. at PM
10	342	149	9260	-	-
0.1	382	87.9	930	-	-
0.01	401	67.6	303	88	1.16
0.001	417	51.4	120	44.9	5.23
0.0001	424	44.7	81.4	21.8	5.98

From the results in table 4.2, the system (joint II) recorded no phase margin values when $B = 10$ to 0.1 because the gains did not cross the 0dB line. When $B = 0.01$, the system recorded the best and acceptable values for robust stability with phase margin of 88dB greater than the gain margin of 67.6dB. However, when $B = 0.001$ and 0.0001 , the phase margins are less than the gain margins which makes the system unstable.

4.3. Joint I Controller Design Results

The results of the two methods of controller design applied in the manipulator joint I controller design are presented as follows:

4.3.1 PID Controller Results for Joint1

The PID controller selected gains for joint I are as follows:

$P = 11.70$, $I = 6.79$, $D = 1.51$, and $N = 153.27$.

Hence, substituting the values into equation, and simplifying, yields

$$K = 11.7 + 6.79 \frac{1}{s} + 1.51 \frac{153.27}{1 + 153.27 \frac{1}{s}}$$

$$K = \frac{243.1s^2 + 1800s + 1041}{s^2 + 153.3s}$$

Figure 4.19 shows the reference tracking for PID controlled joint I. These show that the system achieved a good transient and steady state trajectories. Figure 4.20 shows the bode plot of the loop gain to determine the gain and phase margins and figure 4.21 shows the sigma plot of L, S and T of the PID controlled joint I.

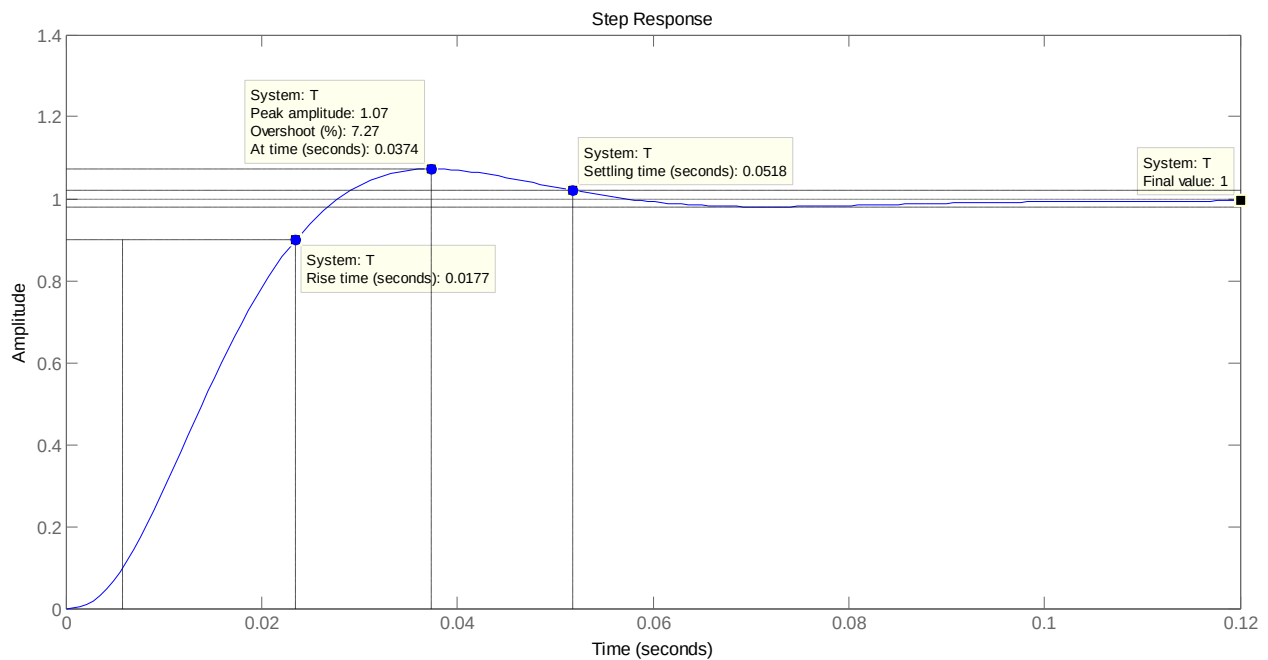


Figure 4.19: Reference tracking plot of PID controlled Joint I

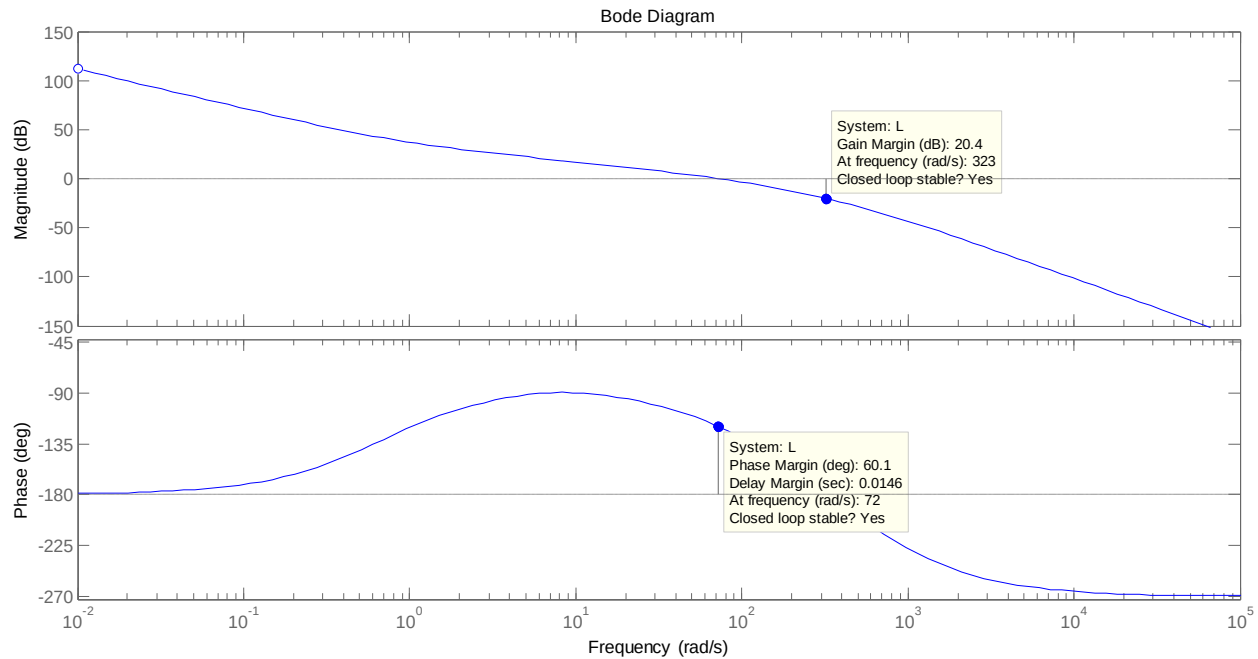


Figure 4.20: Bode plot of loop gain for PID controlled joint I

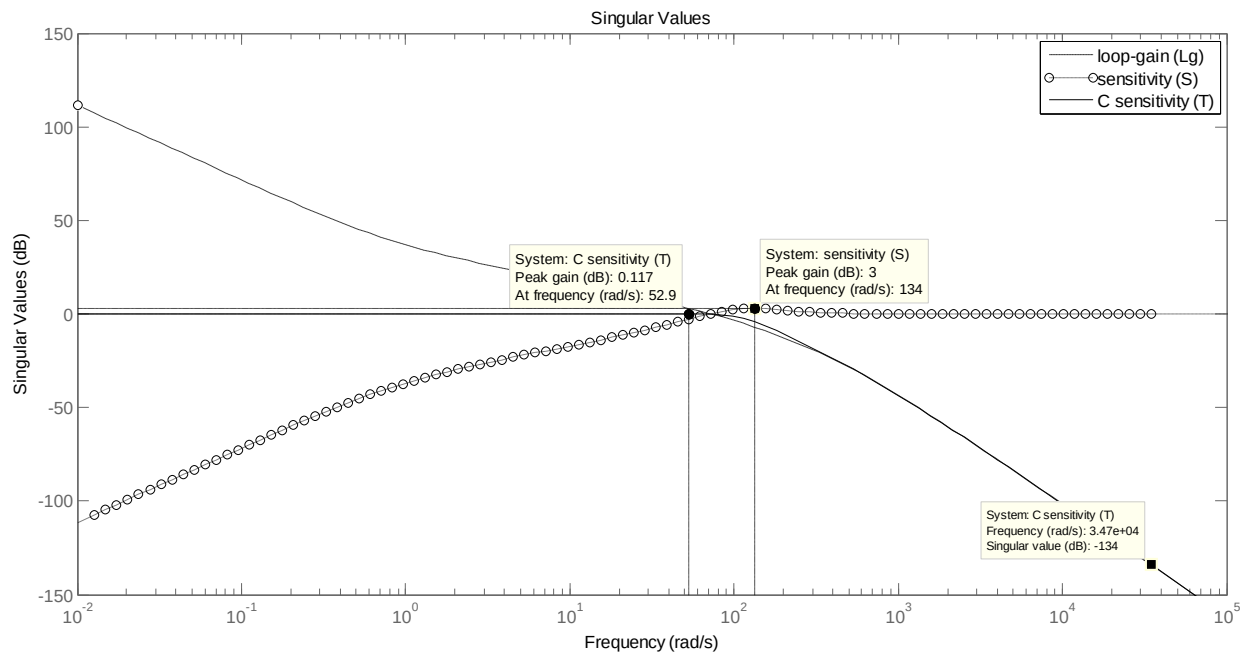


Figure 4.21: Sigma plot of L, S and T for PID controlled joint I

Table 4.3: Performance and robustness results forPID controlledjoint I

Parameter	Value
Loop gain at high freq. for T	-134
System sensitivity at low frequency	-107
Peak Sensitivity	3
Settling Time	0.0518
Overshoot	7.27
Steady state Error (SSE)	0.117
Gain Margin	20.4
Phase Margin	60.1

4.3.2 H-Infinity Controller Design Results for Joint I

Three experiments were carried out by varying the values of the weights especially W1 to determine the best performance and robustness parameter values for the system. Figures 4.22 to 4.30 show the reference tracking response, loop gain bode plot and the sigma plots of the H-Infinity controller design.

Experiment 1

$$W1 = \frac{s + 1000}{100s + 1}$$

$$W2 = 0.1$$

$$W3 = 0$$

Applying H-Infinity synthesis yields:

$$K = \frac{119901.7s^3 + 81715151.28s^2 + 890033510.41s - 3409110783.96}{s^4 + 15540.01s^3 + 10483353.54s^2 + 330407574.02s + 3263467.3}$$

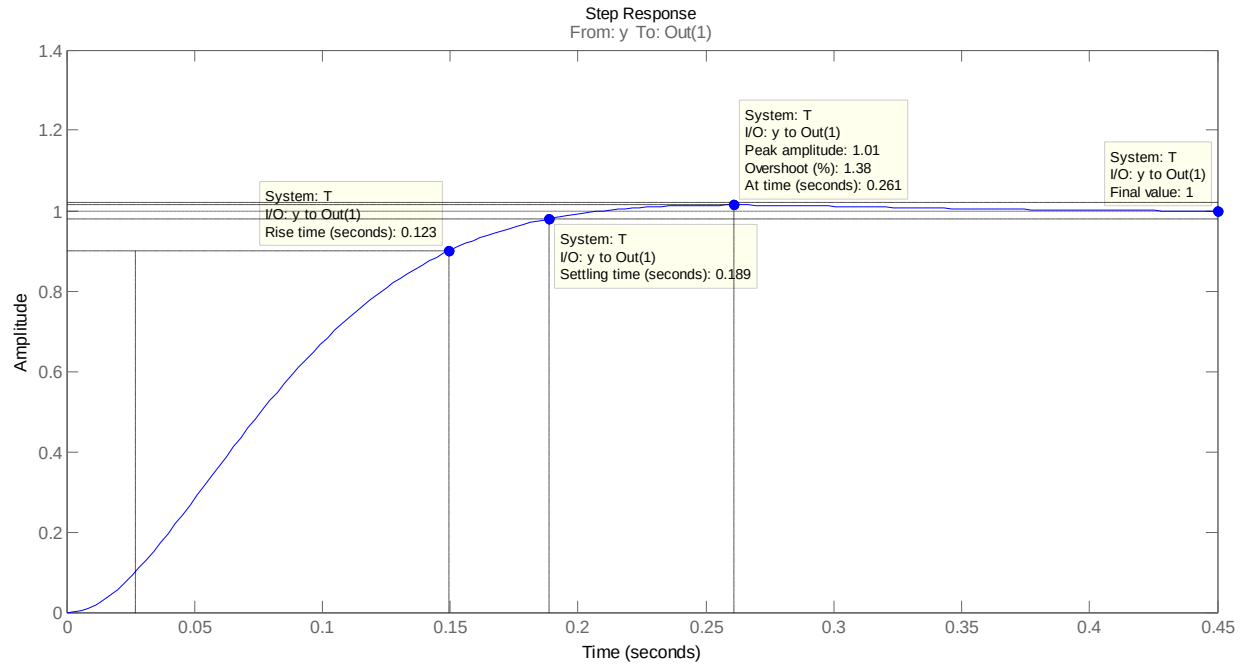


Figure 4.22: Reference tracking response plot of H-Infinity controlled Joint I for Exp.I

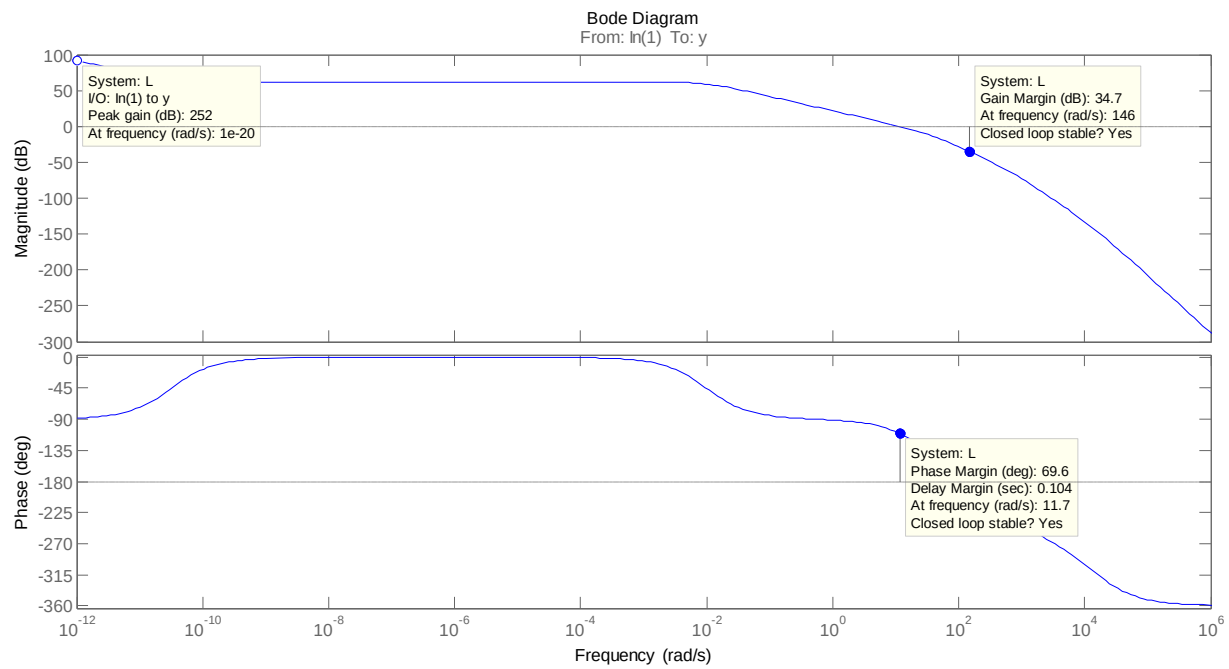


Figure 4.23: Bode plot of loop gain for H-Infinity controlled joint I for Exp.I

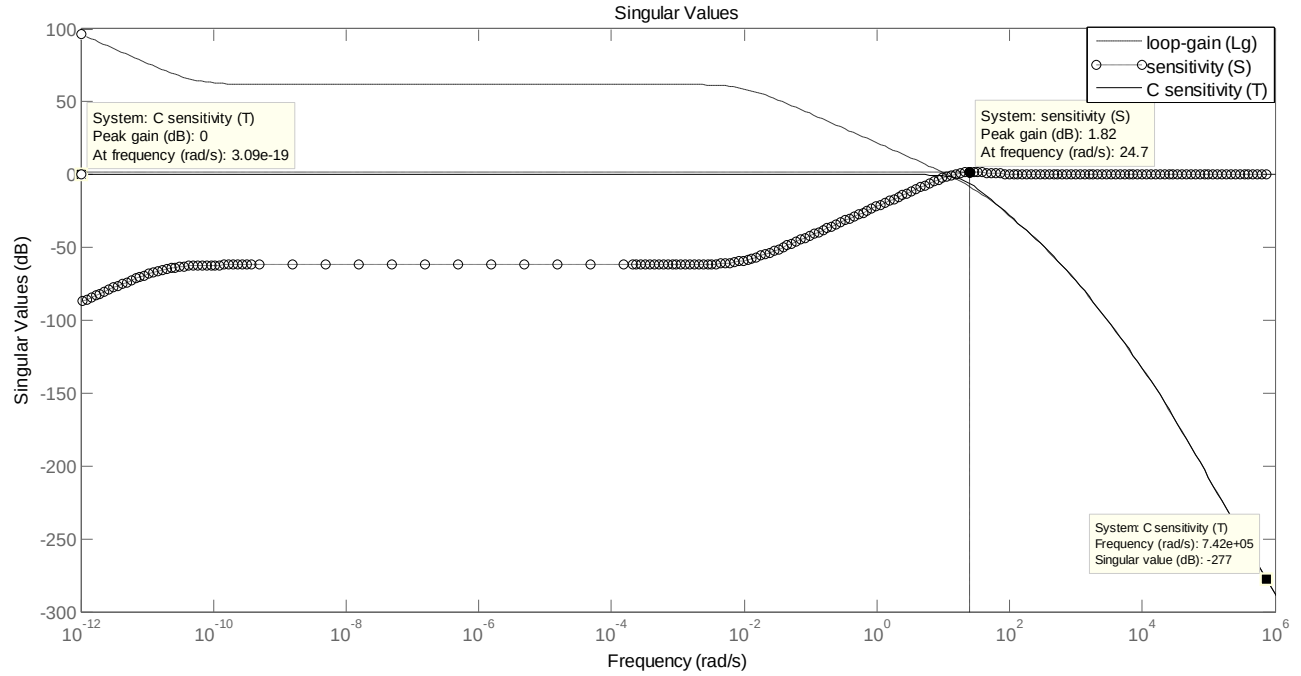


Figure 4.24: Sigma plot of L, S and T for H-Infinity controlled joint I for Exp.I

Experiment II

The weights are:

$$W1 = \frac{0.1(s + 100)}{s + 0.1}$$

$$W2 = 0.1$$

$$W3 = 0$$

The controller transfer function model is:

$$K = \frac{15445.89s^3 + 10524250.9s^2 + 85326973.44s - 115660006.34}{s^4 + 2620.1s^3 + 1373060.01s^2 + 43253036.67s + 4279437.35}$$

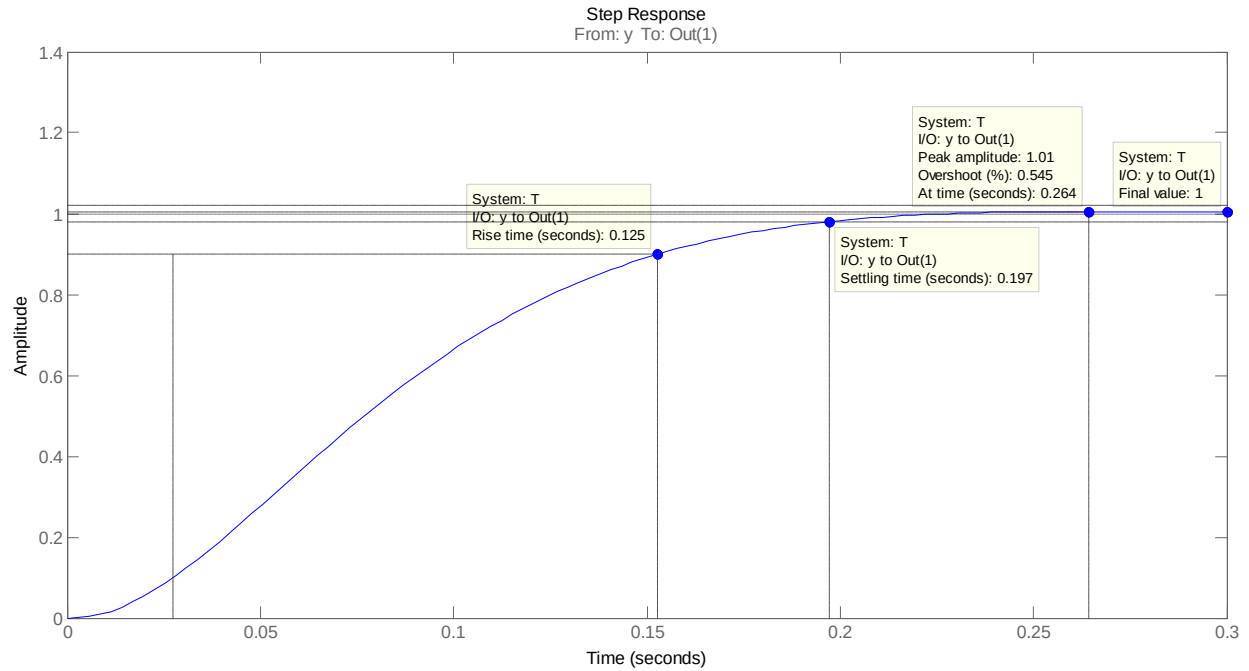


Figure 4.25: Reference tracking response plot of H-Infinity controlled Joint I for Exp.II

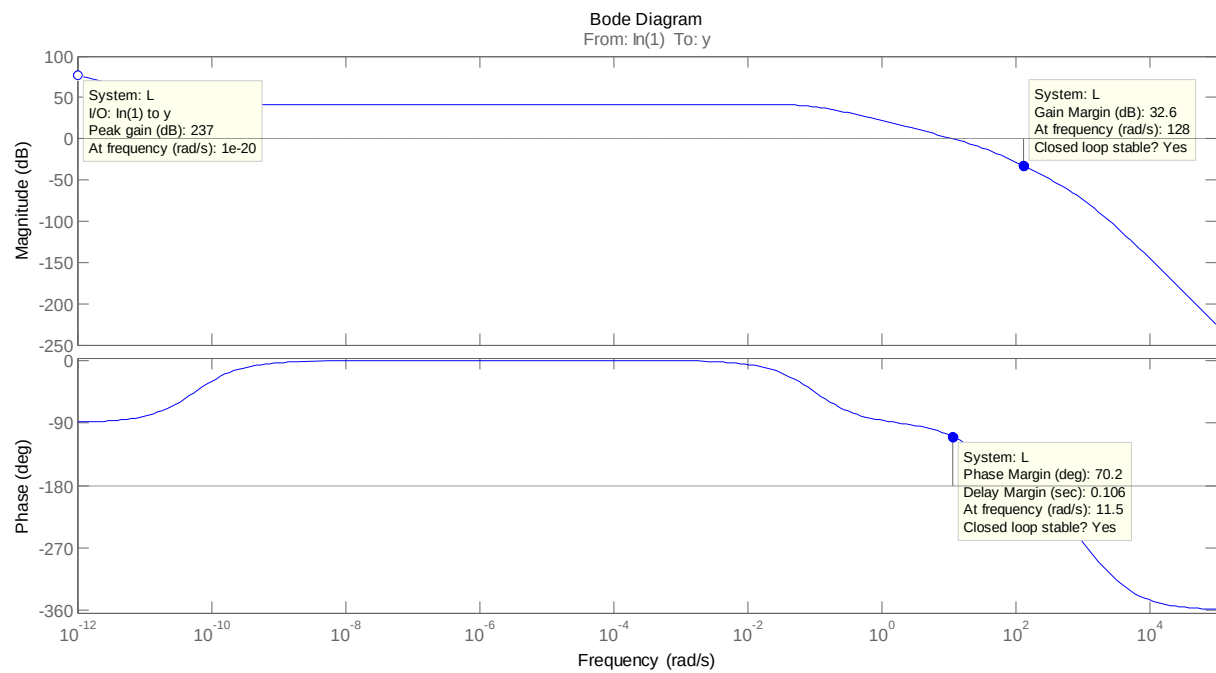


Figure 4.26: Bode plot of loop gain for H-Infinity controlled joint I for Exp.II

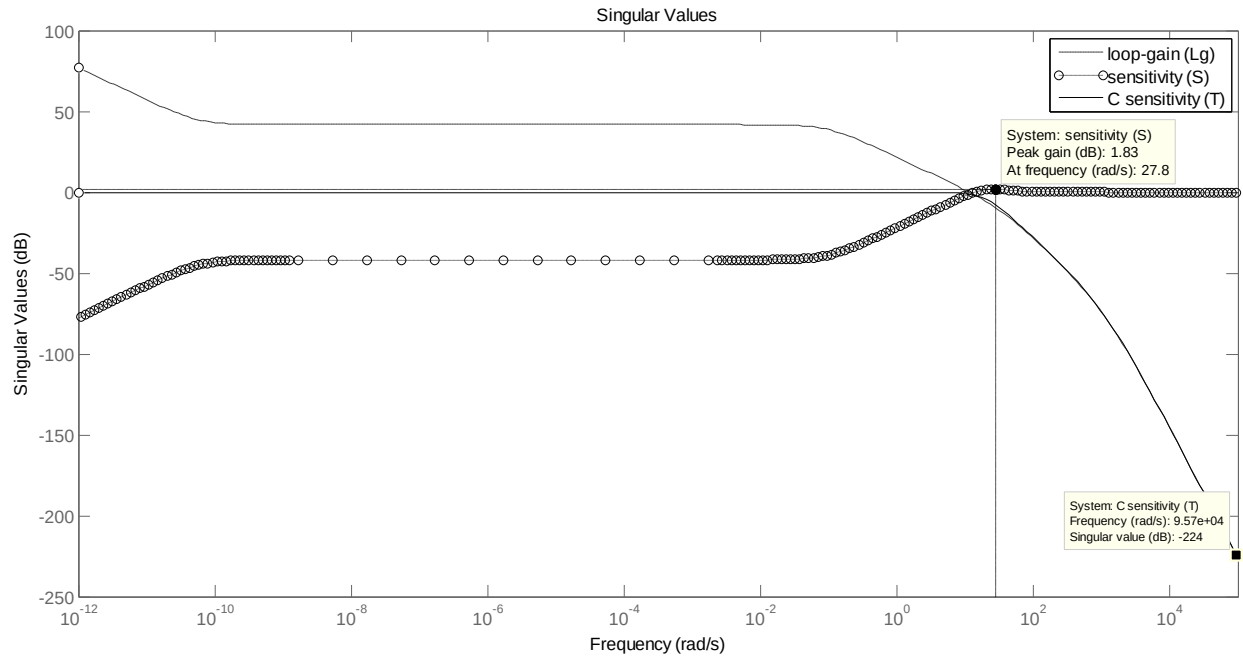


Figure 4.27: Sigma plot of L, S and T for H-Infinity controlled joint I for Exp.II

Experiment III

$$W1 = \frac{0.1(s + 100)}{s + 0.1}$$

$$W2 = 1/(s + 100)$$

$$W3 = 0$$

$$K = \frac{7627984s^4 + 5970009760s^3 + 676138569601s^2 + 42109050879999s + 2415781888001150}{s^5 + 11700.1s^4 + 26977169.18s^3 + 12323975711.74s^2 + 308047269564.02s + 30660460014.33}$$

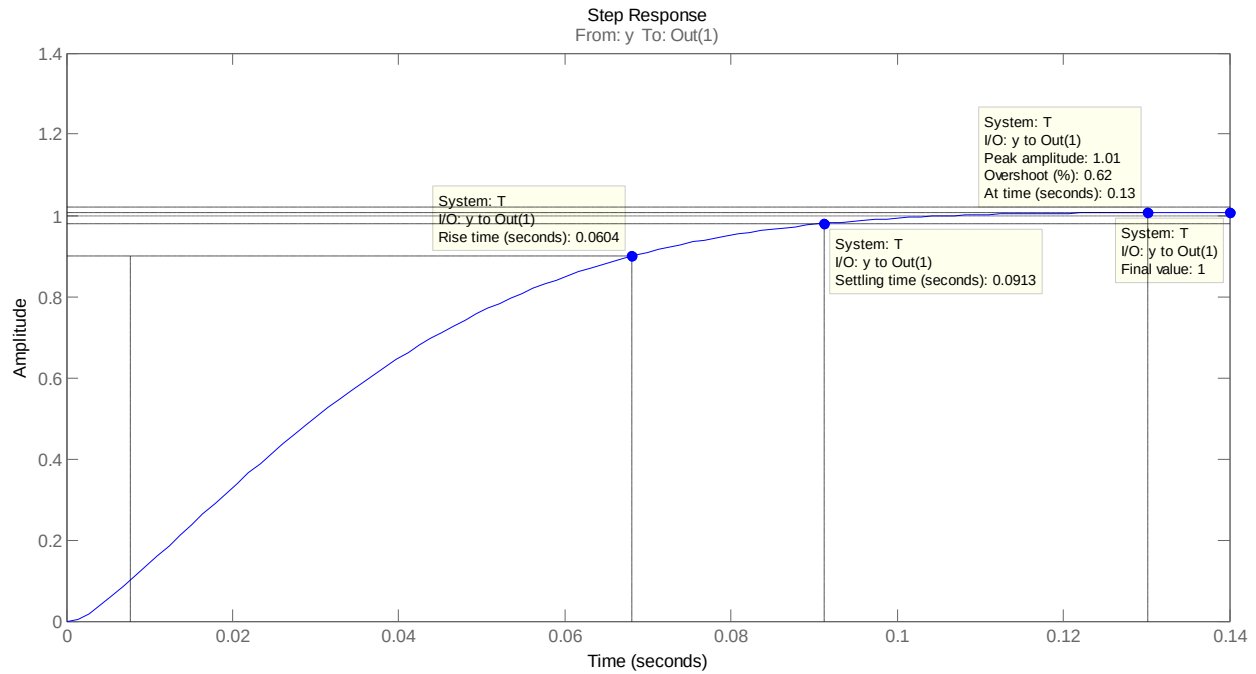


Figure 4.28: Reference tracking response plot of H-Infinity controlled Joint I for Exp.III

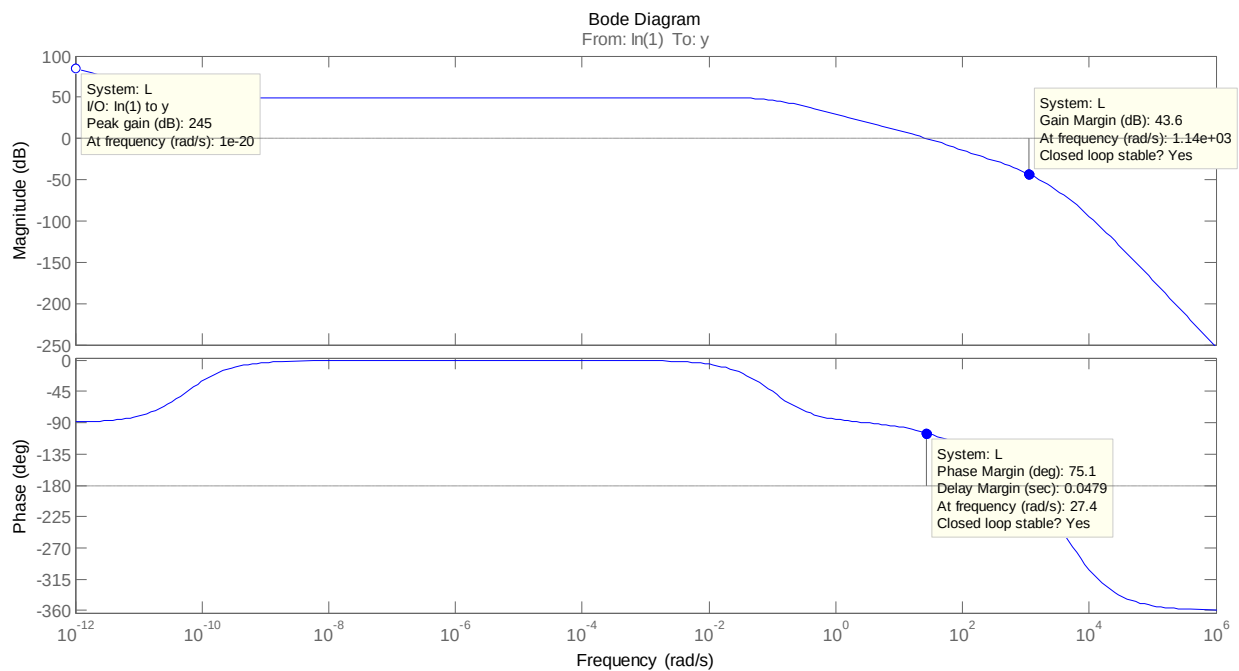


Figure 4.29: Bode plot of loop gain for H-Infinity controlled joint I for Exp.III

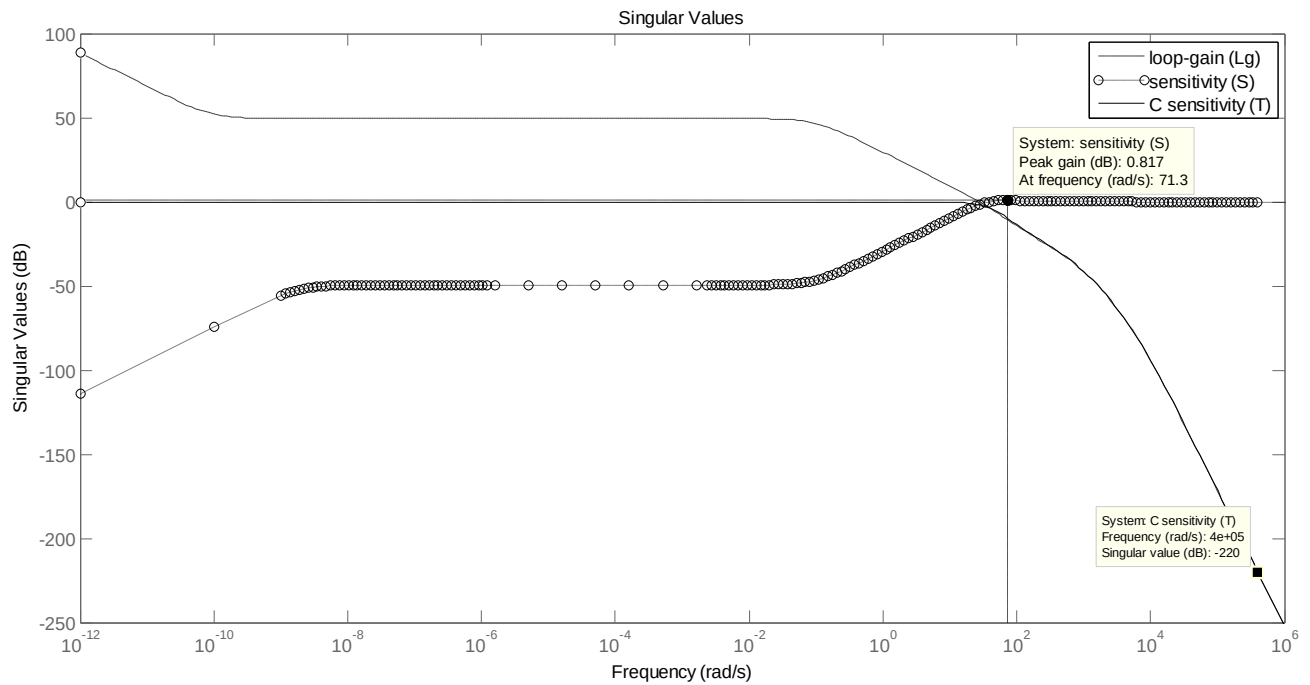


Figure 4.30: Sigma plot of L, S and T for H-Infinity controlled joint I for Exp.III

Table 4.4: Comparing the H-Infinity joint I results for the three experiments

Parameter	Exp.I	Exp.II	Exp.III
Loop gain at high frequency	-277	-224	-220
System sensitivity at low frequency	-86.9	-76.9	-114
Peak Sensitivity	1.82	1.83	0.817
Settling Time	0.189	0.197	0.0913
Overshoot	1.38	0.545	0.62
Steady state Error (SSE)	0	0	0
Gain Margin	34.7	32.6	43.6
Phase Margin	69.6	70.2	75.1

From table 4.4, experiment II recorded the best overshoot of 0.545% but has the highest value of peak sensitivity. However, one of the main goals of robust control is to reduce the level of sensitivity. The system achieved the best performance and robustness parameters values in the third experiment.

4.3.3 Comparing PID and H-Infinity Controller Results for Joint I

Table 4.5 summarizes the performance and robustness parameters results for the PID and H-Infinity controllers at joint I.

Table 4.5: Summary of PID and H-Infinity controller results for joint I

Parameter	PID	H-Infinity
Loop gain at high frequency	-134	-220
System sensitivity at low frequency	-107	-114
Peak Sensitivity	3	0.817
Settling Time	0.0518	0.0913
Overshoot	7.27	0.62
Steady state Error (SSE)	0.117	0
Gain Margin	20.4	43.6
Phase Margin	60.1	75.1

From the results in table 4.5, the H-Infinity controller design method recorded the best performance and robustness parameter values compared to PID controller design method for the manipulator joint I.

4.4. Controller Design Results for Joint II

The controller design methods applied to joint II control are the PID and H-Infinity methods.

4.4.1 PID Controller Design Results for Joint II

The PID controller gains selected for joint II control are as follows:

$P = 1.9765$, $I = 0.5066$, $D = 0.9177$ and $N = 92.0282$

Hence, substituting the values of P , I , D and N into the equation and simplifying, yields

$$K = 1.9765 + 0.5066 \frac{1}{s} + 0.9177 \frac{92.0282}{1 + 92.0282 \frac{1}{s}}$$
$$K = \frac{86.43s^2 + 182.4s + 46.62}{s^2 + 92.03s}$$

Figure 4.31 illustrates the reference tracking graph and figures 4.32 shows the bode plot of the open loop gain and figure 4.33 shows the sigma plot of the PID controlled for joint I.

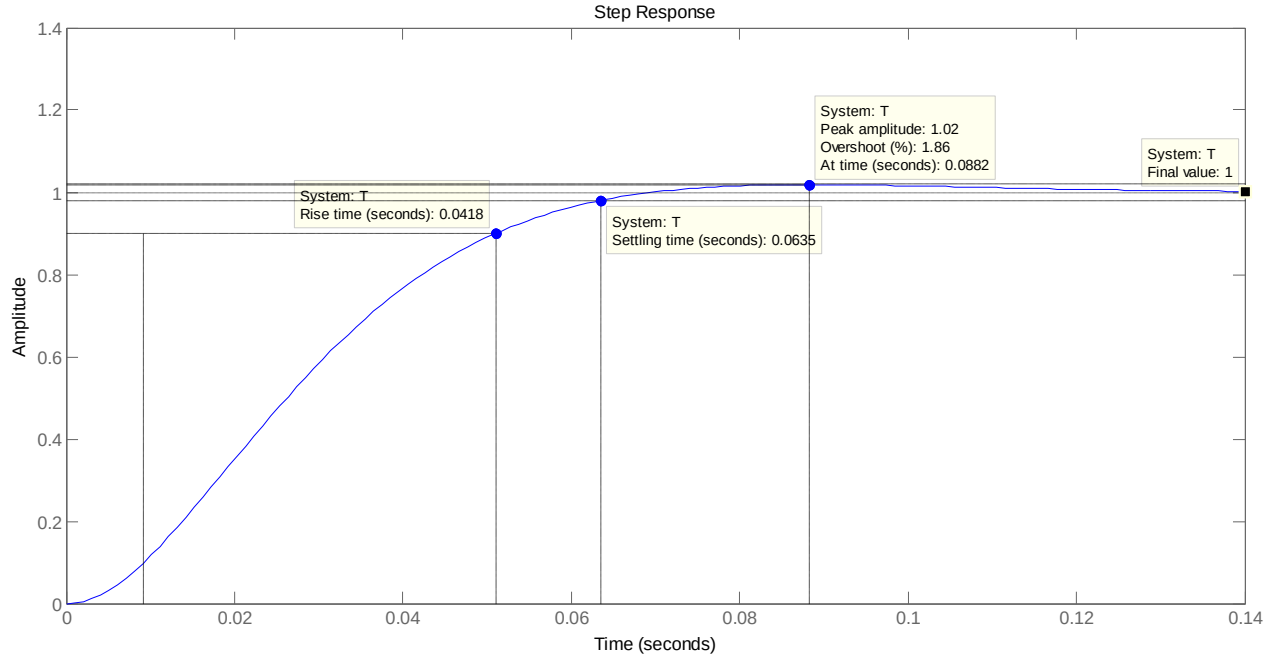


Figure 4.31: Reference tracking response plot of PID controlled Joint II

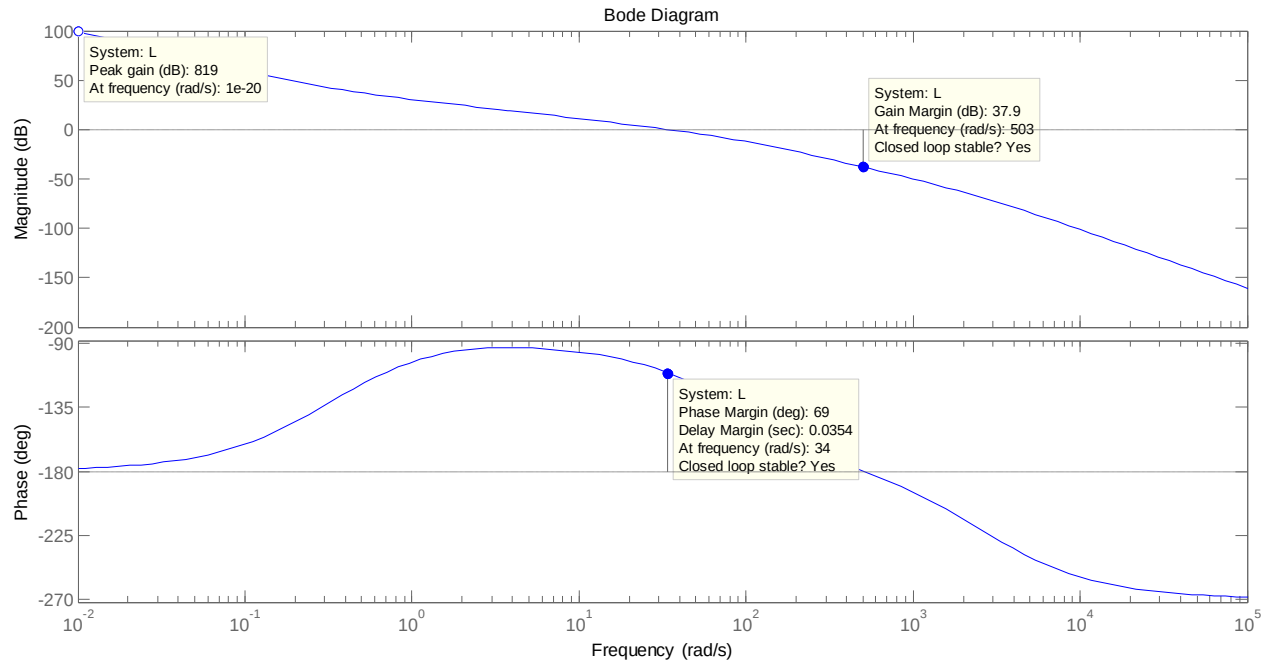


Figure 4.32: Bode plot of loop gain for PID controlled joint II

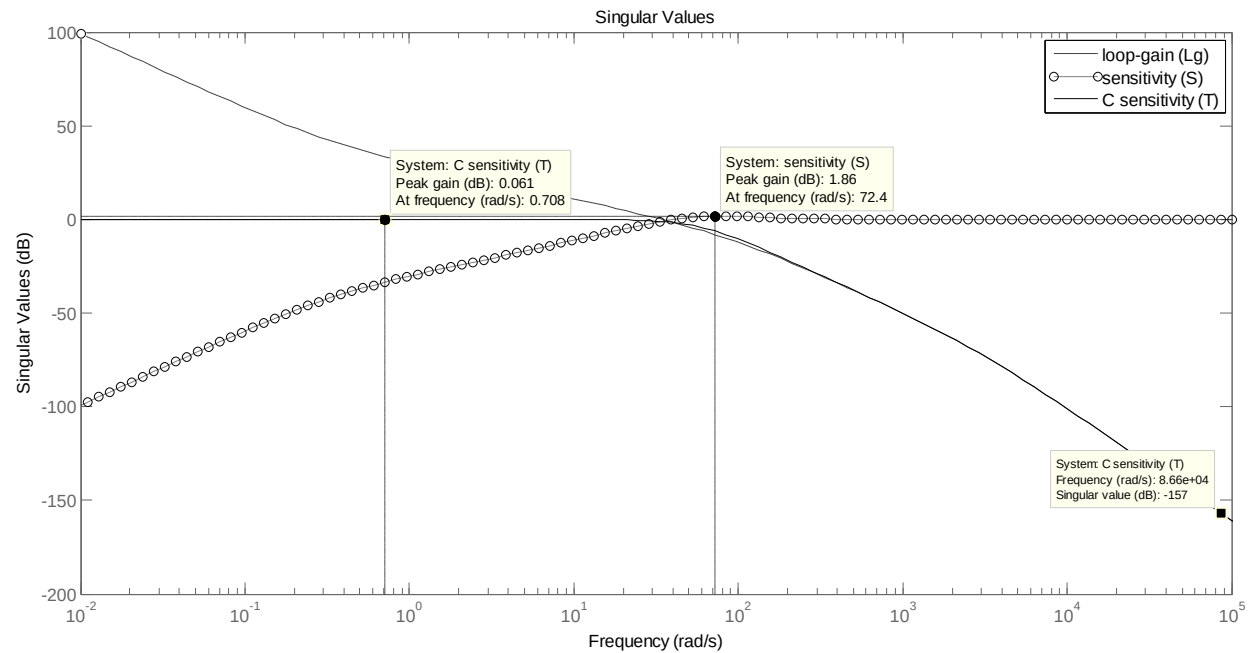


Figure 4.33: Sigma plot of L, S and T for PID controlled joint II

Table 4.6: Performance and robustness results for PID controlled joint II

Parameter	Value
Loop gain at high frequency for T	-153
System sensitivity at low frequency	-97.4
Sensitivity peak gain	1.86
Settling Time	0.0635
Overshoot	1.86
Steady State Error (SSE)	0.061
Gain Margin	37.9
Phase Margin	69

Table 4.2 shows the performance and robustness parameters of the joint II of the manipulator with PID controller.

4.4.2 H-Infinity Controller Design Results for Joint II

Figures 4.34 to 4.42 show the reference tracking response, loop gain bode plot and the sigma plots of the H-Infinity controller design for joint II.

The results are organized in three experiments as follows:

Experiment I:

$$W1 = \frac{0.1(s + 1000)}{100s + 1}$$

$$W2 = 0.1$$

$$W3 = 0$$

Applying the H-Infinity Synthesis yields:

$$K = \frac{205.42s^3 + 564945.24s^2 + 1204793.35s - 57473.34}{s^4 + 2869.01s^3 + 333784.44s^2 + 4170522.97s + 35522.12}$$

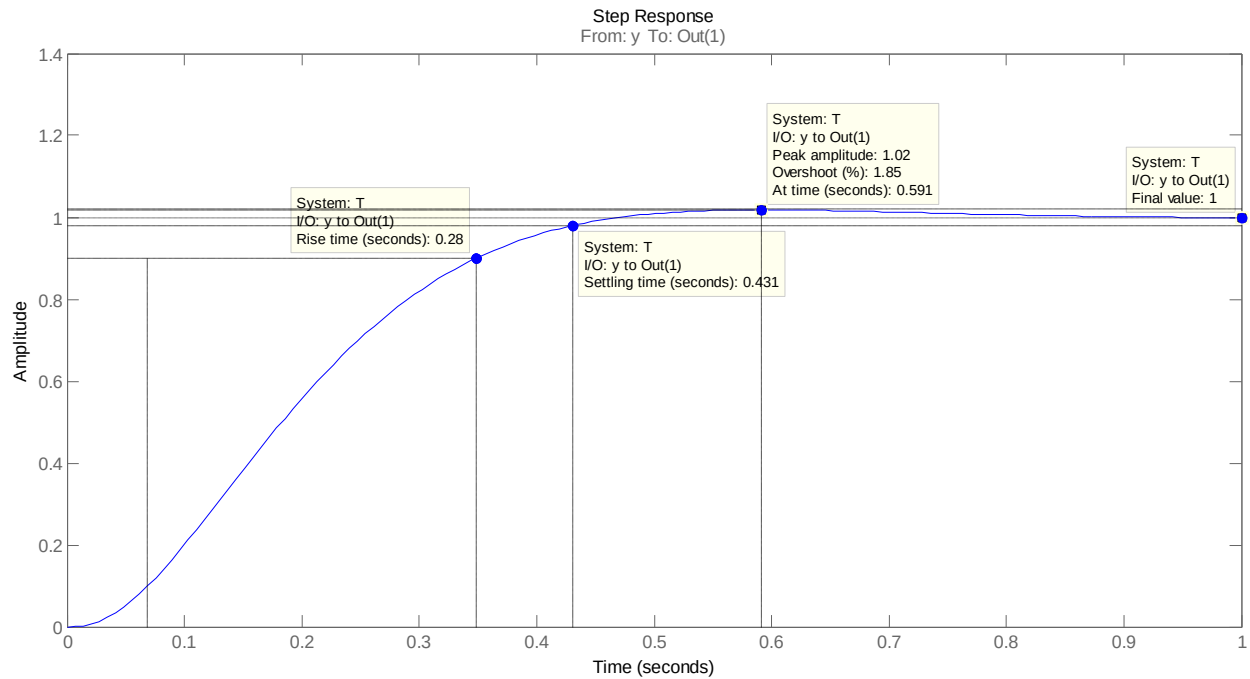


Figure 4.34: Reference tracking response plot of H-Infinity controlled Joint II for Exp.I

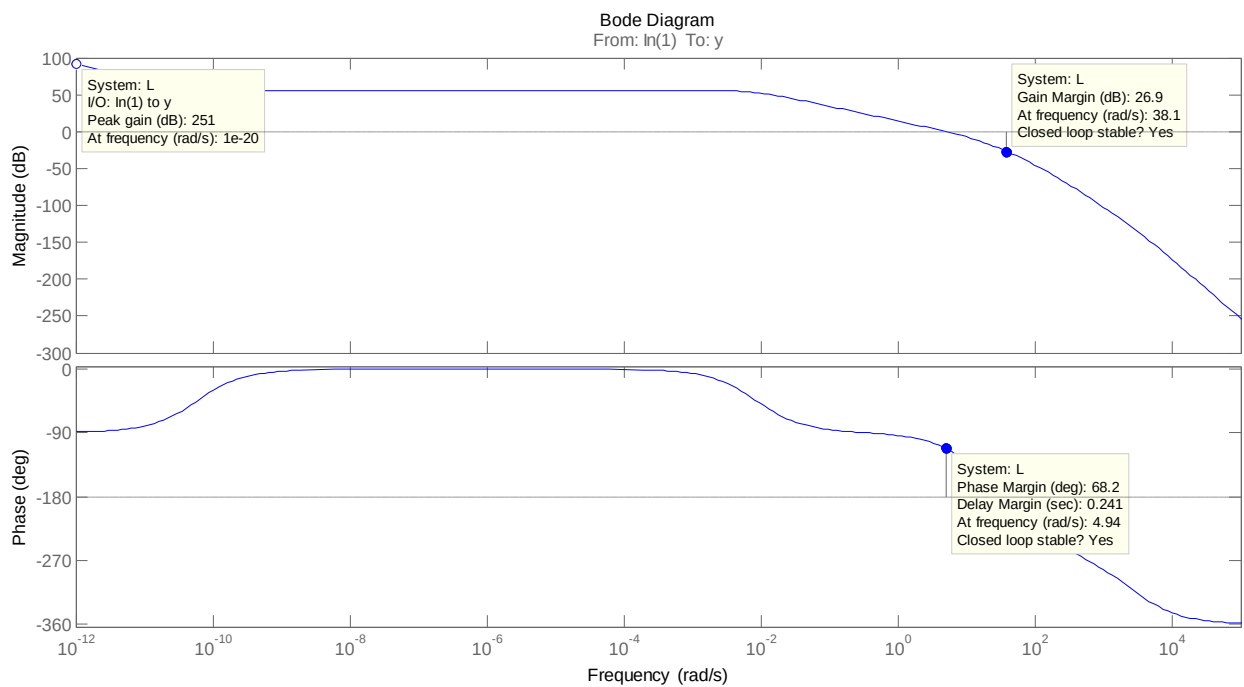


Figure 4.35: Bode plot of loop gain for H-Infinity controlled joint II for Exp.I

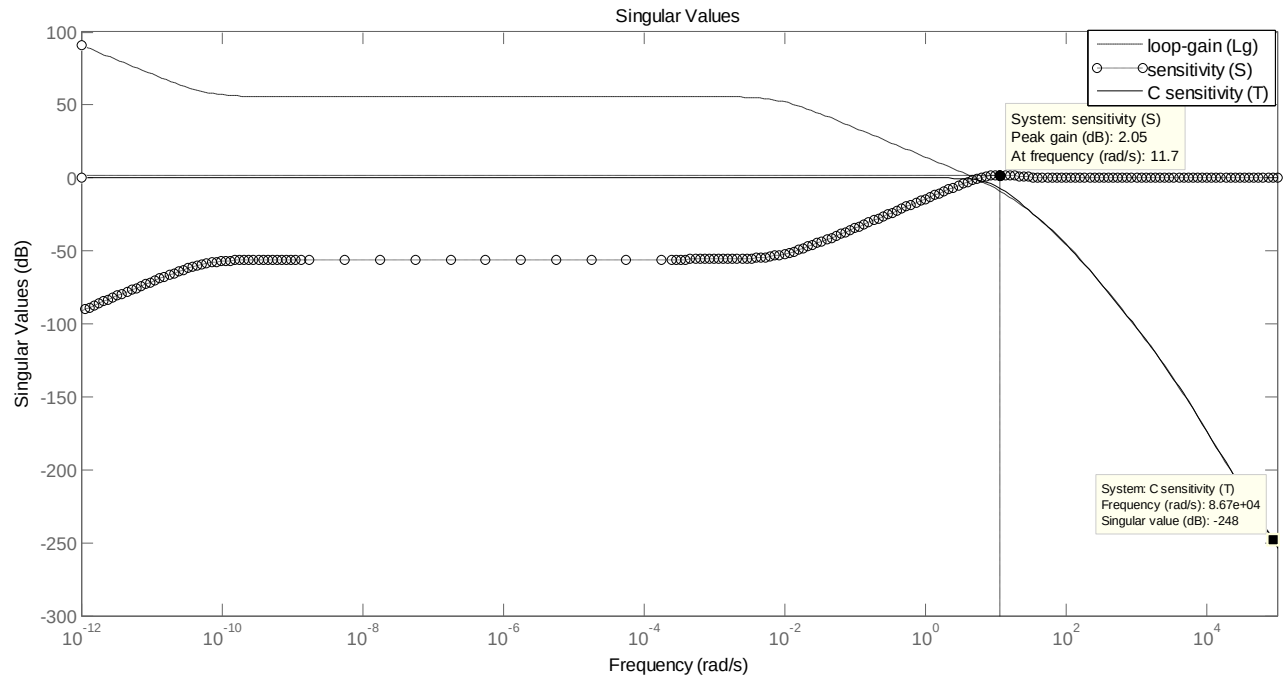


Figure 4.36: Sigma plot of L, S and T for H-Infinity controlled joint II for Exp.I

Experiment II

$$W1 = \frac{0.1(s + 100)}{s + 0.1}$$

$$W2 = 0.1$$

$$W3 = 0$$

$$K = \frac{127397.88s^3 + 349402057.2s^2 + 987942528.03s - 51613862399.94}{s^4 + 17690.1s^3 + 41499356.97s^2 + 1177241541.63s + 116527810.72}$$

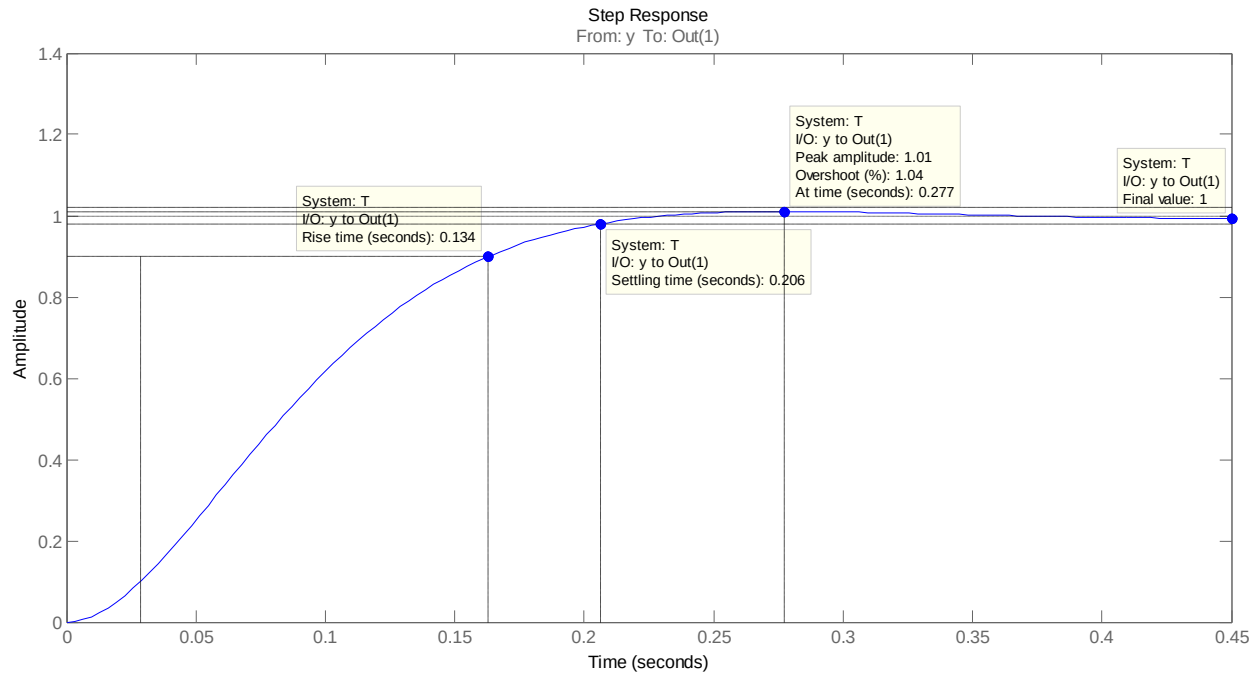


Figure 4.37: Reference tracking response plot of H-Infinity controlled Joint II for Exp.II

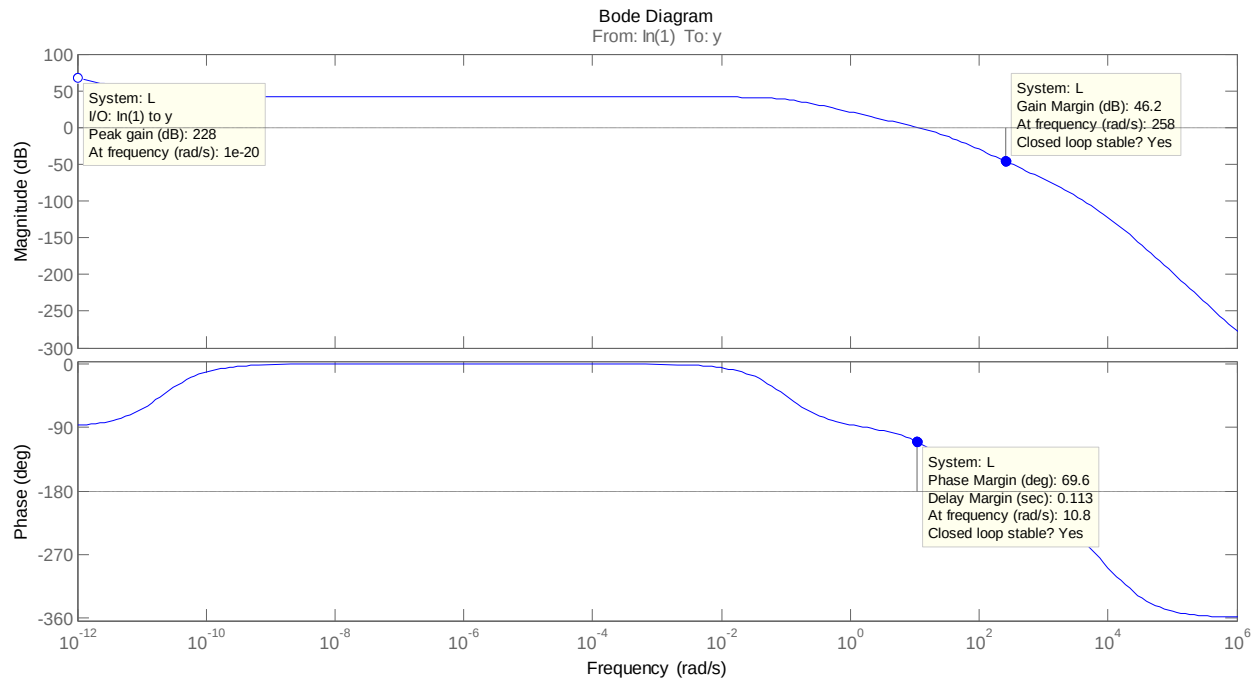


Figure 4.38: Bode plot of loop gain for H-Infinity controlled joint II for Exp.II

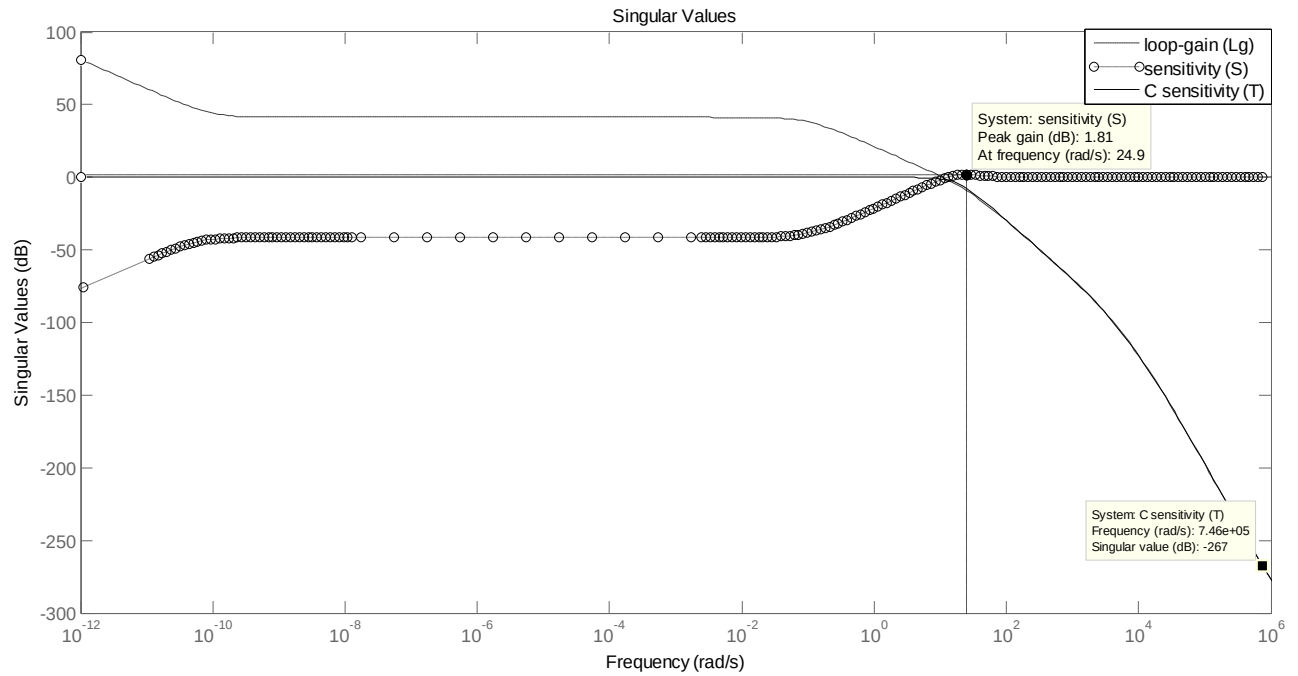


Figure 4.39: Sigma plot of L, S and T for H-Infinity controlled joint II for Exp.II

Experiment III

$$W1 = \frac{s + 100}{s + 0.1}$$

$$W2 = 0.1$$

$$W3 = 0$$

H-Infinity synthesis gives:

$$K = \frac{5141604s^3 + 14140164480.07s^2 - 327455231999.56s - 22228377599996.57}{s^4 + 132400.1s^3 + 364300838.83s^2 + 22492925007.38s + 2178259942.85}$$

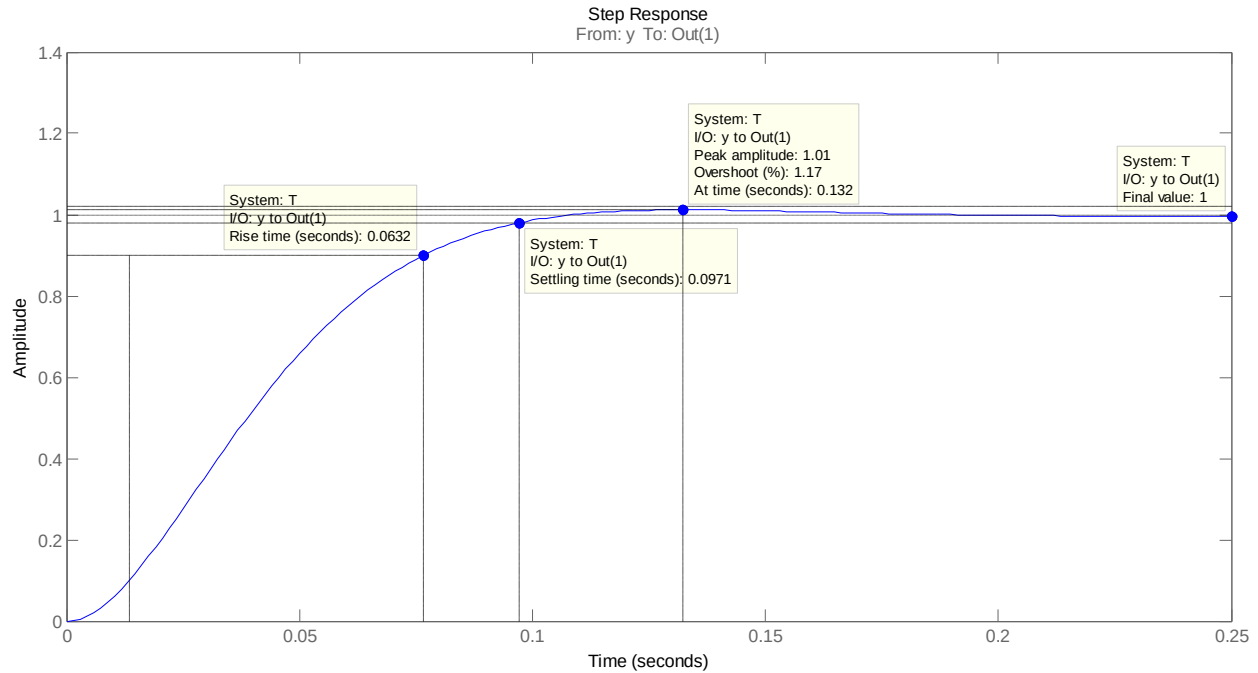


Figure 4.40: Reference tracking response plot of H-Infinity controlled Joint II for Exp.III

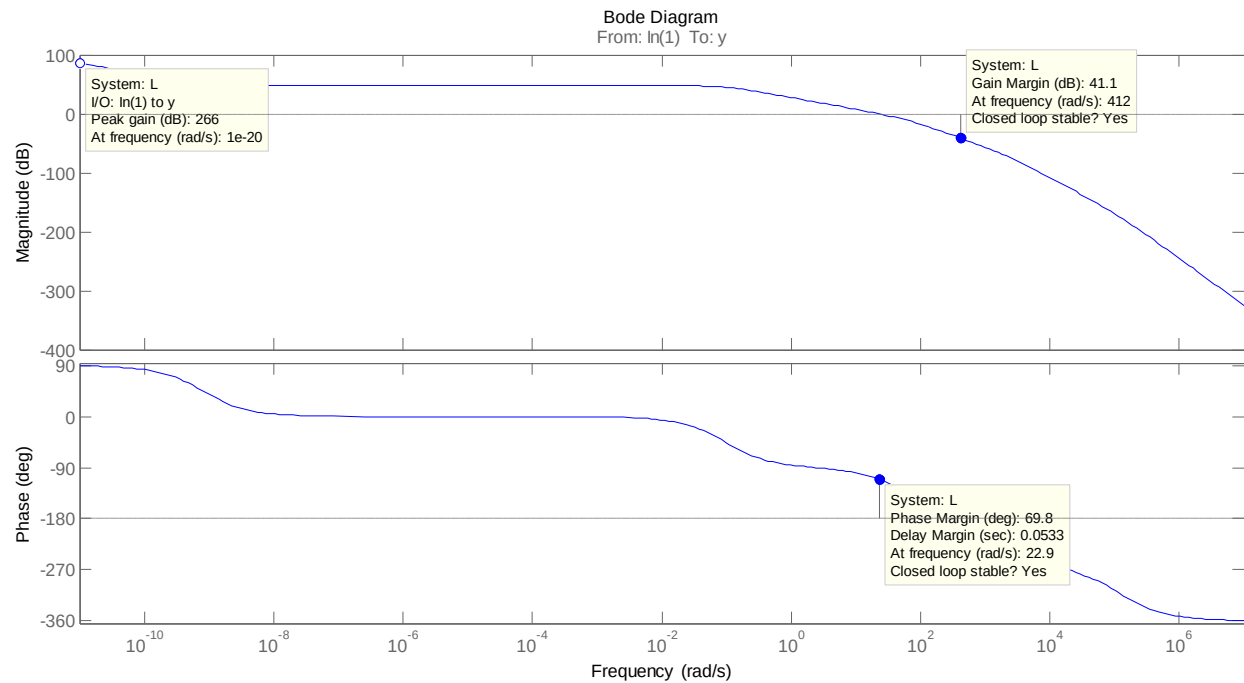


Figure 4.41: Bode plot of loop gain for H-Infinity controlled joint II for Exp.III

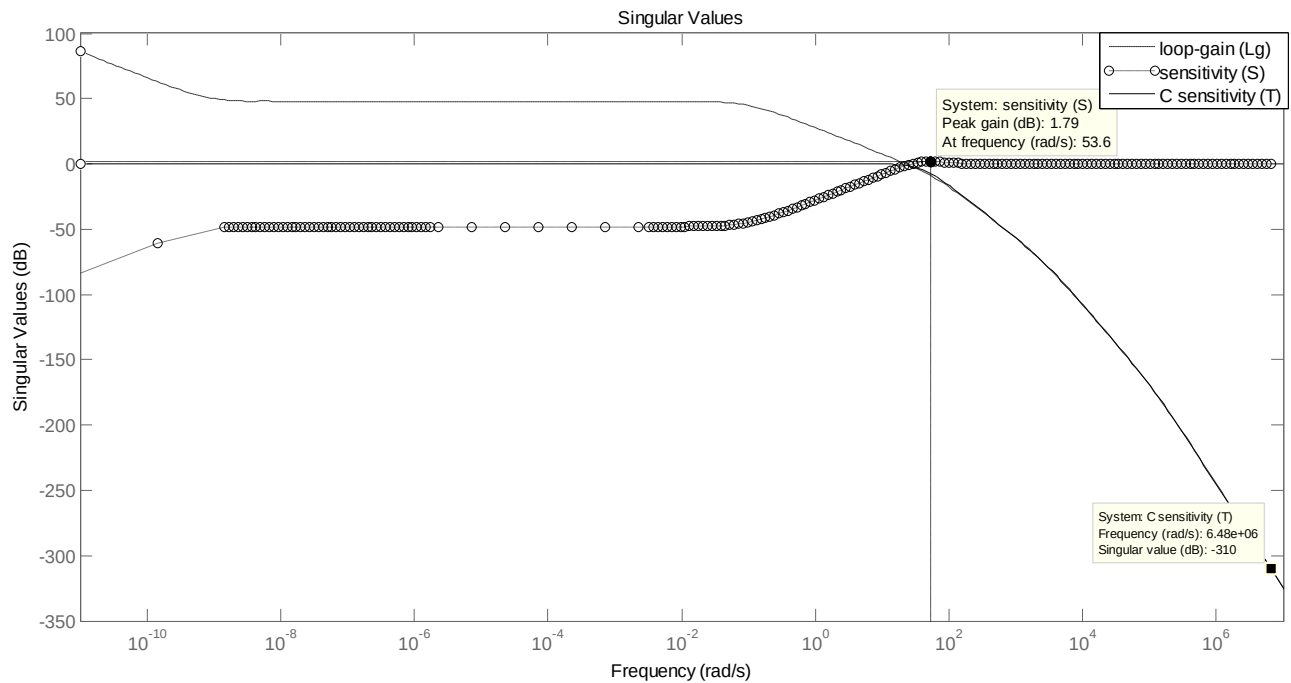


Figure 4.42: Sigma plot of L, S and T for H-Infinity controlled joint II for Exp.III

Table 4.7: Comparing the H-Infinity results for the three experiments for joint II

Parameter	Exp.I	Exp.II	Exp.III
Loop gain at high frequency for T	-248	-267	-310
System sensitivity at low frequency	-90.1	-76	-82.9
Peak Sensitivity	2.05	1.81	1.79
Settling Time	0.431	0.206	0.0971
Overshoot	1.85	1.04	1.17
Steady state Error (SSE)	0	0	0
Gain Margin	26.9	46.2	41.1
Phase Margin	68.2	69.6	69.8

From table 4.7, experiment II recorded the best overshoot of 1.04% but has a higher value of peak sensitivity compare to experiment III. However, considering peak sensitivity, settling time and other parameters, experiment III achieved the best results.

4.4.3 Comparing PID and H-Infinity Controller Results for Joint II

Table 4.8 summarizes the performance and robustness results for the PID and H-Infinity controllers.

Table 4.8: Summary of the PID and H-Infinity controllers results for joint II

Parameter	PID	H-Infinity
Loop gain at high frequency	-153	-310
System sensitivity at low frequency	-97.4	-82.9
Peak Sensitivity	1.86	1.79
Settling Time	0.0635	0.0971
Overshoot	1.86	1.17
Steady state Error (SSE)	0.061	0
Gain Margin	37.9	41.1
Phase Margin	69	69.8

From the results in table 4.8, the H-Infinity recorded the best performance and robustness results with lower values of peak sensitivity, settling time, overshoot, and steady state error; and higher values of gain and phase margins.

4.5. Summary of Discussions

In summary, at joint I, it was observed that the H-Infinity controller achieved better performance and robustness with peak sensitivity of

0.817dB, gain margin of 43.6dB, phase margin of 75.1degrees, steady state error of zero and overshoot of 0.62%, compared to the PID controller results that recorded peak sensitivity of 3dB, gain margin of 20.4dB, phase margin of 60.1degrees, steady state error of 0.117 and overshoot of 7.27%.

At joint II, it was observed that the H-Infinity controller achieved better performance and robustness with peak sensitivity of 1.79dB, gain margin of 41.1dB, phase margin of 69.8degrees, steady state error of zero and overshoot of 1.17%, compared to the PID controller results that recorded peak sensitivity of 1.86dB, gain margin of 37.9dB, phase margin of 69degrees, steady state error of 0.061 and overshoot of 1.86%.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1. Conclusion

The dynamic model of robotic manipulator for robust joint control was achieved based on decentralized control with multiple SISO configurations. This control scheme made it simpler to realize a coupled model of the actuator plus arm dynamics for the robust control. The joint torques due to the arm (link(s)) connected to the joints were determined using the link(s) parameters, and the actuators for the respective joints were selected based on the determined joint torques. Secondly, the actuators were selected to beat least 45g-cm higher than the calculated joint torques. This provides an extra input torque for unplanned torques produced due to uncertainties such as payloads. Other dynamic parameters of the joint actuator such as the torque and electromotive constants that were not given in the manufacturer's catalog but were needed in the formulation of the joint torque control model were determined.

The 2D and 3D models of the articulated 3DOF robotic manipulator were realized in Autodesk Inventor Professional software application and the equivalent SimMechanics model was achieved from the Autodesk inventor 3D model in Matlab environment. The SimMechanics model produced the masses of the links corresponding to the dimensions of the links used. The simulink model of the joint model was realized successfully for the controller design.

The viscous damping coefficient of the joint was derived from the simulation experiment and analysis of the results in the bode plot. It was

confirmed from the results that joint I and II performed best with damping coefficients of 0.00001 N.m/rad /sec and 0.01 N.m/rad /sec respectively.

The robust controller design was achieved using H-Infinity synthesis method and compared with PID controller. This robust controller for the joint torque control was designed based on the independent joint control method. The joint torque control model achieved based on the decentralized control method enabled a simpler way of designing the robust controllers and applying the torque control law. It also enabled a possible and easier way of implementing H-Infinity synthesis.

Good reference tracking means that the output of the manipulator joint tracks the input with little or zero difference. This determines the performance and stability of the system and it depends on the complementary sensitivity transfer function, T . From the results, the T graph at low frequencies for H-Infinity controller followed the zero dB line with zero difference at joint I and II, while the T graph at low frequencies for PID controller followed the line with a difference of 0.117 and 0.061 for joints I and II respectively. Therefore, H-Infinity controller recorded higher performance than the PID controller.

Disturbance rejection characteristic depends on the sensitivity transfer function, S . The sensitivity transfer function graph for H-Infinity recorded peak sensitivity of 0.817dB and 1.79dB at joints I and II respectively while it recorded 3dB and 1.86dB at joints I and II respectively for PID controller. Thus, the H-Infinity controller achieved better disturbance rejection characteristics than PID controller.

Noise attenuation or rejection characteristic depends on the complementary sensitivity transfer function. From the T graph, the H-Infinity recorded lower gains of -220dB and -310dB at joints I and II respectively at high frequencies than the PID which recorded -134dB and -153dB gains at joints I and II respectively at high frequencies. Therefore, H-Infinity controller achieved better noise rejection characteristics than the PID controller.

The controllers maintained good stability all through the experiment, however, the H-infinity showed better robust stability with higher stability margins (gain and phase) and also maintained zero tracking error than the PID controller at the joints I and II.

Finally, from the results, it was concluded that H-Infinity controller guarantees more robustness with better noise and disturbance rejection characteristics, good performance and robust stability characteristics than PID controller.

5.2 Contribution to Knowledge

In this work, reference tracking error of zero was achieved at low frequencies for the robotic manipulator joints I and II using H-infinity synthesis. Hence, this produces high performance stability, and good error management characteristics.

Reduced sensitivities to disturbance were achieved up to 0.817dB and 1.79dB respectively for the manipulator joints using H-infinity synthesis. Thus, this produces good disturbance rejection characteristics for the manipulator to work adequately in the presence of disturbances.

Very low gains up to -220dB and -310dB were achieved at high frequencies in the complementary sensitivity function using H-infinity synthesis for the

manipulator joints. This produces good noise attenuation characteristics for good error control and performance improvement.

Therefore, with the achievements of zero reference tracking error, good disturbance rejection and noise attenuation using H-infinity synthesis, a robust controller was developed for the robotic manipulator to perform effectively in the presence of significant uncertainties.

5.3 Recommendation

The robust H-infinity controller and the independent joint control scheme are recommended to be applied in the development of differential drives for the mobile robots and humanoids. This is because these robots can go through rough terrains and can work under high level of uncertainties therefore they are required to possess robust stability and good noise and disturbance rejection characteristics with very low sensitivity in order to perform adequately.

The robotic manipulator is a physical system which cannot avoid uncertainties, therefore every control technique such as the intelligent control in the form of Artificial Neural Network, Fuzzy Logic Control, Genetic Algorithm etc., to be applied for the torque control of the robotic manipulator should incorporate the robustness scheme so that the system will be able to perform satisfactorily in real environments.

Based on the performance and robustness characteristics of the H-Infinity controller, it is recommended for the control of industrial manipulators, remotely operated vehicles used in deep sea works, medical surgery robots, and space vehicles. The performance and robustness specifications are recommended to be strictly followed in the robust controller design.

The decentralized control model is recommended for the medical surgery robots and other robots used in the nuclear energy works etc. that require high precision and accuracy.

5.4. Suggestion for Further Study

The PID controller produced large loop-gains at low frequencies which can improve performance of the controller. The advantages of the two controllers can be utilized by hybridizing the H-Infinity and PID controllers to achieve a stronger system. Therefore, more research work should be carried out on the combination of the H-Infinity controller and the PID controller to form a hybrid controller.

The Autodesk inventor professional and the SimMechanics tool box are vital materials that help companies to design and test run physical systems such as manipulators. The SimMechanics demonstrates the practical way of applying the control law. However, both the inventor and SimMechanics application programs do not have any means of inputting or adjusting the parameters of the joint actuator and this makes it difficult to complete the robotic manipulator design using these platforms. Since the joint torque control model was achieved in Simulink therefore, further work should be carried out to combine the SimMechanics blocks and the Simulink blocks for the manipulator design by substituting the revolute blocks of the SimMechanics with the Simulink joint model subsystem.

A possible extension of this work would be implementation for the development of a 6 DOF articulated space robot.

REFERENCE

- Abdallah C., Dawson D., Dorato P. and Jamshidi M., (1991). Survey of robust control for rigid robots, *IEEE Control Systems Magazines*, pp.24-30
- Agrawal R., Kabiraj K., Singh R, (2012). Modeling a Controller for an Articulated Robotic Arm, Scientific Research, Intelligent Control and Automation, Vol. 3, pp.207-210
- Ahmad A. M., Mohammed M. K., and Farhan A. S., (2013). Modeling, Simulation and Dynamics Analysis Issues of Electric Motor, for Mechatronics Applications, Using Different Approaches and Verification by MATLAB/Simulink (I). *IJISA* Vol. 5, No. 5, pp.39-57
- Ahuja A., and Tandon B., (2013). Robust PID and Polynomial Controllers Tuning for DC Motor Speed Control using PSO and GA: a Comparative Study, *International Journal of Electrical and Electronics Engineering Research (IJEEER)*, Vol.3, Iss. 1, pp.273-286
- Ahuja A., and Tandon B., (2013). Robust PID and Polynomial Controllers Tuning for Dc Motor Speed Control Using PSO And GA: A Comparative Study, *International Journal of Electrical and Electronics Engineering Research (IJEEER)*, Vol. 3, Issue 1, 273-286
- Alashqar H.A.H., (2007). Modeling and High Precision Motion Control of 3 DOF Parallel Delta Robot Manipulator, The Islamic University of Gaza pp.1-98
- Alassar A.Z., (2010). Modeling and Control of 5DOF Robot Arm using Supervisory Control, University of Gaza, pp.1-105
- Al-Khedher M.A., Alshamasin M. S., (2012). Scara Robot Control using Neural network, 4th International Conference on Intelligent and Advanced Systems
- Alshamasin M.S., Ionescu F., Al-Kasasbeh R.T., (2012). Modelling and simulation of a SCARA robot using solid dynamics and verification by MATLAB/Simulink, *Int. J. Modelling, Identification and Control*, Vol. 15, No. 1
- Al-Tabey W.A., (2012). MATLAB – A Fundamental Tool for Scientific Computing and Engineering Applications: Micro-Robot Management, *InTech*. Vol.3, Pg.284
- An C., Atkeson C., and Hollerbach J., (1988). Model Based Control of a Robot Manipulator, MIT Press, Cambridge, MA
- Arimoto S., and Miyazaki F., (1984). Stability and robustness of PID feedback control for robot manipulators of sensory capability, in *Proceedings of the 1st International Symposium on Robotics Research*, pp. 783–799.

- Arkmann J., and Kaesbauer D., (2001). Design of Robust PID Controllers, Retrieved on March 26, 2019 from: <http://elib.dlr.de/12210/1/design-of-robust-pid-ackermann.pdf>
- Asada H. H., (1998). Dynamics, Massachusetts Institute of Technology, pp.1-16
- Atlas I.H., (2007). Dynamic Model of a Permanent Magnet DC Motor, Karadeniz Technical Universit
- Banga V. K., Kumar R., and Singh Y., (2011). Fuzzy-genetic optimal control for robotic systems, International Journal of the Physical Sciences Vol. 6(2), pp. 204-212
- Bansal A., and Sharma V., (2013). Design and Analysis of Robust H-infinity Controller, Control Theory and Informatics, Vol.3, No.2, pp.7-14
- Baslamish S.C., (2007). LPV Modeling and Robust Control of Yaw and Roll Modes of Road Vehicles, Retrieved on March 26, 2019 from: <http://yunus.hacettepe.edu.tr/~scaglarb/PHD/tezim6.pdf>
- Bellicoso C. D., Buonocore L. R., Lippiello V., and Siciliano B., Design, Modeling and Control of a 5-DoF Light-Weight Robot Arm for Aerial Manipulation, ARCAS large-scale integrating project
- Benson C., (2013). Robot Arm Torque Tutorial, Retrieved on March 26, 2019 from: <http://www.robotshop.com/blog/en/robot-arm-torque-tutorial-7152>
- Biradar R., and Kiran M.C., (2012). The Dynamics of Fixed Base and Free-Floating Robotic Manipulator, International Journal of Engineering Research & Technology (IJERT), Vol. 1 Iss. 5, pg. 1
- Bozma H.I., (2015). Independent Joint Control, Bogazici University, pp.12-14
- Carlson J., Murphy R.R., Nelson A., (2004). Follow-up Analysis of Mobile Robot Failures, University of South Florida, Tampa Florida, pp. 1-8
- Cheng L., Hou Z. G., Tan M., Liu D. and Zou A. M., (2008). Multi-agent based adaptive consensus control for multiple manipulators with kinematic uncertainties, pp.189-194
- Chiu C.S., Lian K.Y., and Wu T.C., (2004). Robust Adaptive Motion/Force Tracking Control Design for Uncertain Constrained Robot Manipulators, Automatica, Vol. 40, pp. 2111-2119
- Craig J.J., (1989). Introduction to Robotics, Mechanics and Control, 2nd Ed., Addison-Wesley
- Crowder R.M., (1998). The Industrial Robot. Retrieved on March 26, 2019 from: <http://www.southampton.ac.uk/~rmc1/robotics/arirobot.htm>
- Crowe S., (2019). Humanoid robots: Five to watch in 2019, WTWH Media, LLC
- Dawson D. M., Qu Z., Lewis F. L., and Dorsey J. F., (1990). Robust control for the tracking of robot motion, *Int. J. Control*, vol.52, no.3, pp.581-595

- D'Azzo J. J., and Houpis C. H., (1988). Linear control system analysis and design: conventional and modern
- De Wit C.C., Siciliano B., Bastin B., (1996). Theory of Robot Control, Springer-Verlag, London
- Dong G., and Zhu Z.H., (2013). Input Shaping-based Impedance Control for Flexible Joint Robotic Manipulator, American Institute of Aeronautics and Astronautics, Conference Paper, pg. 2
- Dorf R.C., and Bishop R.H., (2008). Modern Control Systems, Pearson Prentice Hall, 11th Ed.
- Elfasakhany A., Yanez E., Baylon K., Salgado R., (2011). Design and Development of a Competitive Low-Cost Robot Arm with Four Degrees of Freedom, Modern Mechanical Engineering, Vol. 1, pp. 47-55
- Emerich M., Design of a six Degree-of-Freedom Articulated Robotic Arm for Manufacturing Electrochromic Nanofilms, Retrieved on March 26, 2019 from: <http://oldweb.sbc.edu/sites/default/files/Honors/MEmerich.pdf>
- Eppinger S.D., and Seering W.P., (1992). Three Dynamic Problems in Robot Force Control, IEEE, Vol. 8, No. 6, pp. 751-758
- Fallahi M., and Azadi S., (2011). A Novel Adaptive Sliding Mode Control for DC Motor, International Journal of Advanced Engineering Applications, Vol.4, Iss.3, pp.19-27
- Faramarzi, A., & Sabahi, K., (2011). Recurrent fuzzy neural network for DC motor control, Paper presented at 5th International Conference on Genetic and Evolutionary Computing (ICGEC), IEEE, pp.93-96
- Farhan A. S., (2013). Mechatronics Design of Motion Systems; Modeling, Control and Verification, International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS, Vol: 13, No: 02, pp.1-17
- Fateh M.M., (2008). On the Voltage-Based Control of Robot Manipulators, International Journal of Control, Automation and Systems, Vol. 6, No. 5, pp. 702-712
- Filardi G., Sename O., Voda A.B., Schroeder H.J., (2003). Robust H-Infinity Control of A DVD Drive Under Parametric Uncertainties, Paper ID: 632, pp.1-6
- Filip V., (2008). Dynamic Modeling of Manipulators with Symbolic Computational Method, The Publishing House of the Romanian Academy, Proceedings of the Romanian Academy, Series A, Vol. 9, Num. 3, pg.1
- Fu K., Gonzalez R, Lee C.S.G, (1987). Robotics: Control, Sensing, Vision and Intelligence
- Garulli A., Tesi A., and Vicino A., Uncertainty Models for Robustness Analysis, Control Systems, Robotics, And Automation – Vol. IX
- Ghuffar S., Iqbal J., Mehmood U., And Zubair M., (2006). Design and Fabrication of a Programmable 5-DOF Autonomous Robotic Arm, Proceedings of the 6th WSEAS Int. Conf. on Systems Theory & Scientific Computation, Elounda, Greece, pp.167-173

- Goldstein H., (1980). Classical Mechanics, 2nd Ed., Addison-Wesley
- Grage J.J., (2005). Introduction to Robotics Mechanics and Control, 3rd Ed, Prentice Hall
- Gu D.W., Petkov P. H., and Konstantinov M. M., (2005). Robust Control Design with Matlab, Springer-Verlag London Limited
- Gupta A., Gupta M, Bajpai N, Gupta P, Singh P, (2013). Efficient Design and Implementation of 4-Degree of Freedom Robotic Arm, International Journal of Engineering and Advanced Technology (IJEAT), Volume-2, Issue-5
- Han J., Park J., and Chung W.K, (2011). Robust Coordinated Motion Control of an Underwater Vehicle-Manipulator System with Minimizing Restoring Moments, Elsevier, Ocean Engineering 38, 1197–1206
- Harry L.T., Anton A. S., and Malo H., (2002). Control theory for linear systems, *University of Groningen, Groningen, Netherland*
- Hedaya A., (2011). Control of DC Motor using Different Control Strategies, Global Journal of Technology and Optimization, Vol. 2, pp.21-28.
- Helal K. M., Atia M. R.A., Abu El-Sebah M. I., (2015). Gain Scheduling Control with Multi-Loop PID for 2-DOF Arm Robot Trajectory Control, IRES 23rd International Conference, Dubai, UAE, Pg.19
- Iqbal J., Hassan N., Waqar A., Bano S., Ilyas R., (2005). Fabrication and Control of 4-DOF, Autonomous Robotic Arm Using Low Cost AVR Controller, DARH2005 Conference, Session 5.3, pp1-6
- Iqbal and Author (1995). Dynamic Modeling and Simulation for Control of a Cylindrical Robotic Manipulator, Informatics Complex ICC, Islamabad, Pg. 38
- Izadbakhsh A., and Fateh M. M., (2007). A Model-Free Robust Control Approach for Robot Manipulator, World Academy of Science, Engineering and Technology
- Jaiswal N. K., Kumar V., (2014). 3- DOF Scara type Robot Manipulator using Mamdani Based Fuzzy Controller, International Journal of Scientific Research Engineering & Technology (IJSRET), Vol. 3, Iss. 3, pp. 659-663
- Jaiswal N.K., and Kumar V., (2014). 3- DOF Scara type Robot Manipulator using Mamdani Based Fuzzy Controller, International Journal of Scientific Research Engineering & Technology (IJSRET), Vol. 3, Issue 3, pp659-663
- Kafuko M., Singh, I., and Wanyama, T., 2015. Design of a Robotic Arm for Teaching Integrated Design, Canadian Engineering Education Association (CEEAI) Conference, pp.1-5
- Kambiz A. T., and Augustin M., (2012). Introduction to PID Controllers - Theory, Tuning and Application to Frontier Areas, Retrieved on March 26, 2019 from: <http://www.intechopen.com/books/introduction-to-pid-controllers-theory-tuning-and-application-to-frontier-areas/theory-of-pid-and-fractional-order-pid-fopid-controllers>

- Kelly R., (1998). Global positioning of robot manipulators via PD control plus a class of nonlinear integral actions, *IEEE Trans. on Automat. Contr.*, vol.43, no.7, pp.934-938
- Kim C.S., and Lee K.W., (2011). Robust Control of Robot Manipulators using Dynamic Compensators under Parametric Uncertainty, *International Journal of Innovative Computing, Information and Control*, Vol. 7, No. 7, pp.4129-4137
- Kucuk S., and Bingul Z., (2006). Robot Kinematics: Forward and Inverse Kinematics, *Industrial Robotics: Theory, Modelling and Control*, Sam Cubero (Ed.), Pro LiteraturVerlag, Germany / ARS, Austria, ISBN: 3-86611-285-8, pp117-149
- Kumar R.G.U.V., and Raja C.V.N., (2014). Comparison between FSC and PID Controller for 5DOF Robot Arm, *International Journal of Emerging Trends in Electrical and Electronics (IJETEE – ISSN: 2320-9569)*, Vol. 10, Issue. 2, pp.1-6
- Kumar V., (2011). Introduction to Robot Geometry and Kinematics, Retrieved on March 26, 2019 from: <https://www.google.com.ng/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiGx5j4xsDPAhWLA8AKHQSsC8UQFggaMAA&url=https%3A%2F%2Fwww.researchgate.net%2Ffile.PostFileLoader.html%3Fid%3D579d08bd48954c63a13e6c77%26assetKey%3DAS%253A389641501986816%25401469909181112&usq=AFQjCNHPmfJmgd3XpG-yYhJuPGCpSa2zTg&sig2=t2oIB-UeGkTGsXKcXTGJSA&bvm=bv.134495766,d.d24>
- Kurfess T. R., (2005). Robotics and automation handbook, CRC
- Lewis F.L., Dawson D.M., Abdallah C.T., (2004). Robotic Manipulator Control, Theory and Practice, Marcel Dekker Inc., New York, 2nd Ed. Pg. 152-158
- Lin F., and Brandt R.D., (1997). An Optimal Control Approach to Robust Control of Robot Manipulators, National Science Foundation, pp.1-19
- Liu C.S., and Peng H., (2000). Disturbance Observer Based Tracking Control, *Journal of Dynamic Systems, Measurement, and Control*, Vol. 122, pp. 332-335
- Liu G., and Goldenberg A.A., (1997). Robust Control of Robot Manipulators Based on dynamics decomposition, *IEEE Trans. on Robot. Autom.*, vol.13, no.5, pp.783-789
- Liu J., and Liu R., (2016). Simple Method to the Dynamic Modeling of Industrial Robot Subject to Constraint, *Advances in Mechanical Engineering*, Vol. 8(4) pg.4
- Liu T., (2011). Design And Prototyping of a ThreeDegrees of Freedom Robotic WristMechanism for a Robotic SurgerySystem, Case Western Reserve University, pp.1-83

- Lochan K., and Roy B.K., (2015). Control of Two-link 2-DOF Robot Manipulator Using Fuzzy Logic Techniques: A Review, Springer India, pp. 499-511
- Luo X., Wang J., Shpanin L., Jia N., Liu G., Zinober A.S.I., (2008). Development of a Mathematical Model for Vane-type Air Motors with Arbitrary N Vanes, Proceedings of the World Congress on Engineering, London, U.K, Vol. 1
- Madanipour S., (2014). Modeling and Control of a Servo Hydraulic motor, Indian J.Sci.Res. 1(2): 770-774
- Mahmoud N. A., and Khalil H. K., (1996). Asymptotic regulation of minimum phase nonlinear systems using output feedback, IEEE Trans. on Automat. Contr., vol.AC-41, pp.1402-1412
- Melchiorri C., (1995). Dynamic Model of Robotic Manipulators, University of Bologna
- Melchiorri C., (2006). Control of Robot Manipulators, University of Bologna
- Miro J.V., and White A.S., (2002). Modeling an Industrial Manipulator a Case Study, Simulation Practice and Theory, Vol.9, pp.293-319
- Mizrahi J., (2015). Mechanical Impedance and its Relations to Motor Control, Limb Dynamics and Motion Biomechanics, Journal of Medical and Biological Engineering, Springer 35(1):1-20
- Muhammad A., (2013). On replacing PID controller with ANN controller for DC motor position control, International Journal of Research Studies in Computing, Volume 2 Number 1, 21-29
- Nair S.S., (2011). Automatic Weight Selection Algorithm for Designing H Infinity controller for Active Magnetic Bearing, International Journal of Engineering Science and Technology (IJEST), Vol. 3, No. 1, pp. 122-138
- Nawash N., (2001). H-Infinity Control of an Autonomous Mobile robot, Cleveland State University, pp. 7-8
- Norman S. N., (2011). Control Systems Engineering, John Wiley and Sons, Inc, 6th Edition
- Ogata K., (2009). Modern control engineering: Prentice Hall
- Ovy E.G., Seeraji S., Ferdous S.M., Rokonzaman M., (2011). A Novel Design of an ATmega32L Microcontroller Based Controller Circuit for the Motion Control of a Robot Arm Actuated by DC Motors, Cyber Journals: Journal of Selected Areas in Robotics and Control (JSRC), April Edition, pp.1-8
- Pachaiyappan S., Micheal B.M., Sridhar T., (2014). Design And Analysis Of An Articulated Robot Arm For Various Industrial Applications, IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE), e- ISSN: 2278-1684, p-ISSN: 2320-334X, PP 42-53
- Panich S., (2015). Mathematic Model and Kinematic Analysis for Robotic Arm, Global Journals Inc., USA, Volume 15 Issue 6, Version 1.0, Print ISSN: 0975-5861

- Piltan F., Mirzaei M., Shahriari F., Nasari I., and Emamzadeh S., (2012). Design Baseline Computed Torque Controller, *International Journal of Engineering (IJE)*, Volume (6), Issue (3)
- Pires J.N., (2007). *Industrial Robots Programming: Building Applications for the Factories of the Future*, Springer, pp.75
- Quang H.N., (1998). Dynamic Modelling of Friction Clutches and Application of This Model in Simulation of Drive Systems, *Periodica Polytechnica Ser. Mech. Eng.*, Vol. 42, No.2, Pp. 169-181
- Ren X., Rad A., Lewis F., (2007). Neural network-based compensation control of robot manipulators with unknown dynamics, in: *American Control Conference, 2007. ACC'07*, pp. 13-18
- Reyes F., and Kelly R., (2001). Experimental Evaluation of Model-Based Controllers on a Direct Drive Robot Arm, *Mechatronics*, Vol. 11, pp. 267-282
- Richter H., *Robot Dynamics and Control*, MCE 503, pp.3-6
- Sage H. G., de Mathelin M. F. and Ostertag E., (1999). Robust control of robot manipulators: A survey, *Int. J. Control*, vol.72, no.16, pp.1498-1522.
- Salem F.A., (2013). Modeling, Simulation, Controller Selection and Design of Electric Motor for Mechatronics Motion Applications, using Different Control Strategies and Verification using MATLAB/SIMULINK, *European Scientific Journal*, Vol. 9, No. 27, pp.211-237
- Salem F.A., (2014). Modeling, Simulation and Control Issues for a Robot ARM; *Education and Research, I.J. Intelligent Systems and Applications*, 04, 26-39
- Sato K., Mukai H., Tsuruta K., (2008). An adaptive H_{∞} control for robotic manipulator with compensation of input torque uncertainty, *Proceedings of the 17th World Congress, The International Federation of Automatic Control Seoul, Korea*, pp.8919-8924
- Scherer C., (2001). *Theory of Robust Control*, Delft University of Technology, Netherlands
- Schiller Z., (1996). Time-Energy Optimal Control of Articulated Systems with Geometric Path Constraints, *Journal of Dynamic Systems, Measurement and Control*, Trans. ASME, Vol. 118, pp. 139-143
- Sciavicco L., and Siciliano B., (1996). *Modeling and Control of Robot Manipulators*, 2nd Ed., McGraw-Hill
- Shamrao A.K., and Viswanatha N., (2013). Predicting the Dynamic Behaviour of Hybrid Stepper Motor, *Proceedings of the 1st International and 16th National Conference on Machines and Mechanisms (iNaCoMM2013)*, IIT Roorkee, India
- Shuaib A. O., Ahmed M. M., (2014). Robust PID Control System Design Using ITAE Performance Index (DC Motor Model), *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 3, Iss. 8, pp. 15059-15067

- Shweta P., and Sanjay L., (2012). Position Control of Pick and Place Robotic Arm, EIE's 2nd Intl' Conf. Comp., Energy, Net., Robotics and Telecom
- Slotine J. J. E. and Li W., (1991). Applied nonlinear control, Prentice hall Englewood Cliffs, NJ, vol. 461
- Spong M. W., and Vidyasagar M., (1989). Robot Dynamics and Control, John Wiley & Sons, Inc., 1st Ed
- Spong M. W., Hutchinson S., and Vidyasagar M., (2006). Robot Modeling and Control, John Wiley & Sons, Inc., 1st Ed
- Spong M.W., (1992). On the Robust Control of Robot Manipulators, IEEE Trans. On Automatic Control, Vol. 37, No. 11, pp. 1782-1786
- Su C. Y. and Leung T. P., (1993). A sliding mode controller with bound estimation for robot manipulators, *IEEE Trans. on Robot. Autom.*, vol.9, no.2, pp.208-214
- Symon K.R., (1987). Mechanics, 3rd Ed., Addison-Wesley
- Talole S.E., Phadke S.B., and Singh R.K., Robust Feedback Linearization Based Control of Robot Manipulator, Institute of Armament Technology, Girinaga, Pune, India, pg. 89
- Tan W., Chen T., and Marquez H.J, (1999). Robust Controller Design and PID Tuning for Multivariable Processes, North China Electric Power University, pp.1-22
- Tarek A., Yixin D., Joseph L. H., Chenyang L, and Xiaoyun Z., Introduction to Control Theory And Its Application to Computing Systems, University of Illinois, Urbana-Champaign, USA
- Toivonen H.T., (1998). Robust Control Methods, Abo Akademi University Turku (Abo), Finland, pp. 1-8
- Torres S., Mendez J.A., Acosta L., and Becerra V.M, (2007). On Improving the Performance in the Robust Controllers for Robot Manipulators with Parametric Disturbances, Control Engineering Practice, Vol. 15, pp. 557-566
- Voung N.D., and Ang M.H., (2009). Dynamic Model Identification for Industrial Robots, Acta Polytechnica Hungarica, Vol. 6, No. 5, pp. 64
- Wang H, Feng Z, and Liu X, (2008). Robust Tracking Control of Robot Manipulator Using Dissipativity Theory, Modern Applied Science, Vol. 2, No. 4, Pg. 95
- Yadav P S., and Singh N., (2015). Robust Control of Two Link Rigid Manipulator, International Journal of Information and Electronics Engineering, Vol. 5, No. 3, pp.198-203
- Yim J., and Park J.H., (1999). Nonlinear H_∞ Control of Robotic Manipulator, IEEE, pp.866-871
- Zhu Y., and Fan L., (2012). On Robust Hybrid Force/Motion Control Strategies Based on Actuator Dynamics for Nonholonomic Mobile Manipulators, Hindawi Publishing Corporation Journal of Applied Mathematics, Volume 2012, pp.1-19