

4-2021

## **MODELLING AND CONTROL OF A TWO-LINK RIGID-FLEXIBLE MANIPULATOR**

Rahma Saeed Rasheed AlNajjar

Follow this and additional works at: [https://scholarworks.uaeu.ac.ae/all\\_theses](https://scholarworks.uaeu.ac.ae/all_theses)



Part of the [Mechanical Engineering Commons](#)

---

United Arab Emirates University

College of Engineering

Department of Mechanical Engineering

**MODELLING AND CONTROL OF A TWO-LINK RIGID-FLEXIBLE  
MANIPULATOR**

Rahma Saeed Rasheed AlNajjar

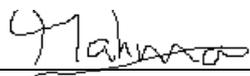
This thesis is submitted in partial fulfilment of the requirements for the degree of  
Master of Science in Mechanical Engineering

Under the Supervision of Dr. Tariq Darabseh

April 2021

### Declaration of Original Work

I, Rahma Saeed Rasheed AlNajjar the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this thesis entitled “*Modelling and Control of a Two-Link Rigid-Flexible Manipulator*”, hereby, solemnly declare that this thesis is my own original research work that has been done and prepared by me under the supervision of Dr. Tariq Darabseh, in the College of Engineering at UAEU. This work has not previously been presented or published or formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my thesis have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this thesis.

Student's Signature:  Date: 27/4/2021

Copyright © 2021 Rahma Saeed Rasheed AlNajjar  
All Rights Reserved

## **Advisory Committee**

1) Advisor: Dr. Tariq Darabseh

Title: Associate Professor

Department of Mechanical Engineering

College of Engineering

2) Co-advisor: Dr. Farag Khalifa Omar

Title: Associate Professor

Department of Mechanical Engineering

College of Engineering

## Approval of the Master Thesis

This Master Thesis is approved by the following Examining Committee Members:

- 1) Advisor (Committee Chair): Dr. Tariq Darabseh

Title: Associate Professor

Department of Mechanical Engineering

College of Engineering

Signature 

Date 25/4/2021

- 2) Member: Dr. Mohamed Okasha

Title: Assistant Professor

Department of Mechanical Engineering

College of Engineering

Signature 

Date 25/4/2021

- 3) Member (External Examiner): Dr. Daniil Yurchenko

Title: Associate Professor

Department of Mechanical Engineering

School of Engineering and Physical Sciences

Institution: Heriot-Watt University, UK

For Signature   
Program Coordinator.

Date 25/10/2021

This Master Thesis is accepted by:

Dean of the College of Engineering: Professor James Klausner

Signature James F. Klausner Date 26/10/2021

Dean of the College of Graduate Studies: Professor Ali Al-Marzouqi

Signature Ali Hassan Date 26/10/2021

Copy \_\_\_\_ of \_\_\_\_

## Abstract

Literature lacks data on the reliability of 3D models created by Autodesk Inventor software and imported to MATLAB Simulink software in comparison to mathematically generated models. In this contribution, a two-link rigid-flexible manipulator modelled in two different methods was demonstrated, one of which is using Lagrange equations and Finite Element Method to generate a mathematical model of the manipulator, and the other is creating a 3D model with the aid of Autodesk Inventor then import to MATLAB Simulink, both models were subsequently controlled by three types of controllers, conventional PID controller, LQR controller, and LQG controller. The research demonstrated the performance of the two models with response to the three types of controllers. Achieved results has proven that the Autodesk Inventor is considered as reliable tool for modelling mechanical systems. Results have also confirmed that modern controllers, i.e., LQR and LQG controllers perform much better than conventional PID controller with regards to the manipulator movement. The implementation of Autodesk Inventor along with MATLAB Simulink indicates that, the Autodesk Inventor can be considered as an instrumental tool for designers and engineers. The results enable future developments in the frontier area of robotics and mechanical systems, where sophisticated models could be generated by Autodesk Inventor instead of being modelled mathematically which will benefit engineers and designers by saving time and effort consumed in modelling using mathematical equations, and by reducing the potential errors associated with such modelling techniques.

**Keywords:** Dynamic model, Flexible manipulators, Finite Element method, Autodesk Inventor, LQR controller, LQG controller, PID controller.

## Title and Abstract (in Arabic)

### النمذجة الرياضية الديناميكية لمناور مرن مكون من حلقتين مترابطتين

#### الملخص

إن عدد البحوث التي تحتوي على بيانات حول موثوقية النماذج ثلاثية الأبعاد التي تم إنشاؤها بواسطة برنامج Autodesk Inventor والتي يتم تصديرها فيما بعد إلى برنامج MATLAB Simulink قليل مقارنةً بالنماذج التي تم إنشاؤها رياضياً. في هذه المساهمة، تم عرض مناورا صلباً مرناً ثنائي الارتباط تم تصميمه بطريقتين مختلفتين، إحداهما تستخدم معادلات لاجرانج وطريقة العناصر المحدودة لإنشاء نموذج رياضي للمعالج، أما الطريقة الأخرى فقد تمت من خلال إنشاء نموذج ثلاثي الأبعاد للمناور باستخدام برنامج Autodesk Inventor ثم تصديره إلى MATLAB Simulink، تم التحكم في كلا النموذجين لاحقاً بواسطة ثلاثة أنواع من وحدات التحكم، وحدة تحكم PID التقليدية، وحدة تحكم LQR، ووحدة تحكم LQG. أظهر البحث أداء النموذجين بالاستجابة لأنواع المتحكمات الثلاثة. كما أثبتت النتائج المحققة أن Autodesk Inventor يعتبر أداة موثوقة لنمذجة الأنظمة الميكانيكية. أكدت النتائج أيضاً أن وحدات التحكم الحديثة، أي وحدات التحكم LQR و LQG تعمل بشكل أفضل بكثير من وحدة التحكم PID التقليدية فيما يتعلق بحركة المناور. يشير استخدام برنامج Autodesk Inventor جنباً إلى جنب مع برنامج MATLAB Simulink إلى أنه يمكن اعتبار برنامج Autodesk Inventor أداة مفيدة للمصممين والمهندسين. تتيح النتائج فرصة للتطورات المستقبلية في كل من مجال الروبوتات ومجالات الأنظمة الميكانيكية بشكل عام، حيث يمكن إنشاء نماذج متطورة أو معقدة التركيب بواسطة برنامج Autodesk Inventor بدلاً من تكوين نموذج باستخدام الطرق الرياضية المتعارف عليها في نمذجة هذه التركيبات الميكانيكية، مما سيفيد المهندسين والمصممين من خلال توفير الوقت والجهد المبذولين في النمذجة باستخدام المعادلات الرياضية، ومن خلال تقليل الأخطاء المحتملة المرتبطة بتقنيات النمذجة هذه.

**مفاهيم البحث الرئيسية:** النموذج الديناميكي، المناور المرن، طريقة العناصر المحدودة، معادلات لاجرانج، المتحكم.

## **Acknowledgements**

I would like to thank Dr. Tariq Darabseh for his guidance, support, and assistance throughout my preparation of this thesis.

Special thanks go to my father, husband, brothers, and sisters who helped me along the way. I am sure they suspected it was endless. In addition, special thanks are extended to the UAEU for their intensive care of their students. I am so proud of being one of them.

## Dedication

*To my father and everyone who believed in me*

## Table of Contents

Title.....	i
Declaration of Original Work .....	ii
Copyright .....	iii
Advisory Committee .....	iv
Approval of the Master Thesis .....	v
Abstract .....	vii
Title and Abstract (in Arabic) .....	viii
Acknowledgements .....	ix
Dedication .....	x
Table of Contents .....	xi
List of Tables.....	xiii
List of Figures .....	xiv
List of Abbreviations.....	xvii
Chapter 1: Introduction and Literature Review .....	1
1.1 Introduction and State of Art.....	1
1.1.1 Applications of Rigid-Flexible Manipulators.....	1
1.1.2 Advantages of Rigid-Flexible Manipulators .....	2
1.2 Research Objectives .....	3
1.3 Expected Impact and Contribution of this Research.....	4
Chapter 2: Literature Review .....	5
2.1 Relative Background.....	5
2.2 Techniques Used in Analyzing and Modeling the Rigid-Flexible Manipulators.....	6
2.2.1 Assumed Mode Method (AMM).....	7
2.2.2 Lumped Parameter Method .....	8
2.2.3 Finite Elements Method (FEM).....	9
2.3 Previous Research .....	10
Chapter 3: Mathematical/Dynamic Modelling.....	24
3.1 Approach.....	24
3.2 Arm Geometry .....	24
3.3 Modeling of the Rigid-Flexible Manipulator.....	25
3.4 Damping Modeling .....	36
3.5 State Space Representation .....	37
Chapter 4: Control of Flexible Manipulator.....	40

4.1 Modeling and Simulation .....	40
4.1.1 Simulink Model .....	42
4.1.2 MIMO System .....	46
4.2 Simulation Results .....	46
4.2.1 Simulation Results Without Any Controller .....	47
4.2.2 Simulation Results Using PID Controller .....	52
4.2.3 Simulation Results Using LQR Controller.....	61
4.2.4 Simulation Results Using LQG Controller.....	69
4.3 Comparison Between Proposed PID Controller with the Literature.....	77
4.4 Discussion .....	80
4.5 Comparative Analysis of PID, LQR and LQG Controllers .....	80
Chapter 5: Modeling Based on Autodesk Inventor and MATLAB Simulink Software.....	87
5.1 PID Controller Implemented on Simulated Model of Manipulator .....	90
5.2 LQR Controller Implemented on SimMechanics Model of Manipulator .....	99
5.3 LQG Controller Implemented on SimMechanics Model of Manipulator .....	106
5.4 Comparative Analysis of PID, LQR and LQG Controllers .....	112
Chapter 6: Conclusion and Future Work .....	116
6.1 Conclusion .....	117
References .....	120
Appendices.....	123
Appendix A: Newton's Second Law Dynamic Modelling .....	123
Appendix B: Mathematical Description for Dynamic Model.....	124
Appendix C: Laplace Transformation.....	134
Appendix D: MATLAB Codes .....	141

## List of Tables

Table 1: Initial values for link 1 and link 2.....	39
Table 2: PID tuning results .....	54
Table 3: PID proposed results .....	79
Table 4: Comparative analysis of three controllers.....	84
Table 5: Dimension of the simulated model .....	88
Table 6: PID controller settings .....	90

## List of Figures

Figure 1: Schematic for three revolute joint manipulators.....	25
Figure 2: Rigid flexible manipulator.....	27
Figure 3: Beam elements for two links .....	28
Figure 4: Element nodal displacement schematic .....	28
Figure 5: Flow chart of modelling methodologies.....	41
Figure 6: The input signal for hub 1.....	42
Figure 7: The input signal torque of hub 2.....	43
Figure 8: Simulink diagram .....	45
Figure 9: Rotational angle ( $\theta$ ) of link 1 without a controller .....	47
Figure 10: Deflection of nodes of link 1 without a controller.....	48
Figure 11: Rotational displacement of link 1 without a controller .....	48
Figure 12: Rotational angle ( $\theta$ ) link 2 without a controller .....	49
Figure 13: Deflection of nodes of link 2 without a controller.....	50
Figure 14: Rotational displacement of link 2 without a controller .....	50
Figure 15: Model simulation using PID controller to track angular displacement ....	53
Figure 16: PID tuning block from MATLAB control system library .....	55
Figure 17: Angular displacement $\theta_2$ of link 1 with PID controller.....	56
Figure 18: Angular displacement $\theta_2$ of link 2 with PID controller.....	56
Figure 19: Deflection of nodes of link 1 and link 2 with PID controller .....	57
Figure 20: Angular velocity of link 1 and link 2 with PID controller.....	57
Figure 21: Angular acceleration of link 1 and link 2 with PID controller .....	59
Figure 22: Actuation torque for both joints of manipulator with PID controller.....	60
Figure 23: Simulation of LQR control scheme .....	63
Figure 24: Angular displacement of link 1 with LQR controller.....	64
Figure 25: Angular displacement of link 2 with LQR controller.....	65
Figure 26: Deflection in both joints from the desired reference with LQR controller.....	66
Figure 27: Actuation torque for both joints with LQR controller.....	67
Figure 28: Angular velocity of both joints with LQR controller .....	68
Figure 29: Angular acceleration of both joints with LQR controller.....	68
Figure 30: Simulation of LQG control scheme.....	71
Figure 31: Angular displacement of link 1 with LQG controller.....	72
Figure 32: Angular displacement link 2 with LQG controller.....	73
Figure 33: Angular deflection of both joints with LQG controller .....	74
Figure 34: Angular velocity of both joints with LQG controller .....	75
Figure 35: Angular acceleration of both joints with LQG controller.....	75
Figure 36: Actuation torque signal from the LQG controller with LQG controller ..	76
Figure 37: Trajectories of $\theta_1$ and $\theta_2$ of the literature.....	78
Figure 38: Angular displacement of first joint for the PID controller .....	78
Figure 39: Angular displacement of second joint for the PID controller.....	79

Figure 40: Comparison between angular displacement of link 1 for the three controllers .....	81
Figure 41: Comparison between angular displacement of link 2 for the three controllers .....	81
Figure 42: Comparison between actuation torque of link 1 for the three controllers .....	82
Figure 43: Comparison between actuation torque of link 2 for the three controllers .....	83
Figure 44: RMS graph of PID controller .....	85
Figure 45: RMS graph of LQG controller .....	85
Figure 46: RMS graph of LQR controller.....	86
Figure 47: Two- link manipulator model in Autodesk Inventor .....	87
Figure 48: Model exported in MATLA .....	89
Figure 49: PID controller settings in MATLAB .....	91
Figure 50: Complete Simulink model with PID controller.....	93
Figure 51: Angular displacement of link 1 of the simulated model with PID controller.....	94
Figure 52: Angular displacement of link 2 of the simulated model with PID controller.....	95
Figure 53: Angular deflection of both links of the simulated model with PID controller.....	96
Figure 54: Actuation torque to both links of the simulated model with PID controller.....	97
Figure 55 Angular velocity of both links of the simulated model with PID controller.....	98
Figure 56 Angular acceleration of both links of the simulated model with PID controller .....	98
Figure 57: Simulation of two-link manipulator with LQR controller.....	100
Figure 58: Angular displacement of link 1 of the simulated model with LQR controller.....	101
Figure 59: Angular displacement of link 2 of the simulated model with LQR controller.....	102
Figure 60: Angular deflection of both links of the simulated model with LQR controller.....	103
Figure 61: Actuation torque for both links of the simulated model with LQR controller.....	104
Figure 62: Angular velocity of both links of the simulated model with LQR controller.....	105
Figure 63: Angular acceleration of both joints of the simulated model with LQR controller.....	105
Figure 64: Simulink model of the LQG implemented controller in manipulator ....	107
Figure 65: Angular displacement of link 1 of the simulated model with LQG controller.....	108

Figure 66: Angular displacement of link 2 of the simulated model with LQG controller.....	108
Figure 67: Angular deflection of both links of the simulated model with LQR controller.....	109
Figure 68: Actuation torque feed to both link actuators of the simulated model with LQR controller .....	110
Figure 69: Angular velocity of both links of the simulated model with LQR controller.....	111
Figure 70: Angular acceleration of both links of the simulated model with LQR controller.....	111
Figure 71: Angular displacement of link 1 of the simulated model with the three controllers .....	112
Figure 72: Angular displacement of link 2 of the simulated model with the three controllers .....	113
Figure 73: Actuation torque of link 1 of the simulated model with the three controllers .....	114
Figure 74: Actuation torque of link 2 of the simulated model with the three controllers .....	114
Figure 75: Free body diagram schematic for two-link manipulator .....	123
Figure 76: Link 1 schematic diagram.....	124
Figure 77: Link 1 free body and kinetic diagrams Free-body diagram and Kinetic diagram .....	125
Figure 78: Equivalent simple spring mass system for link 1 .....	127
Figure 79: Link 2 schematic diagram.....	128
Figure 80: Link 2 free body and kinetic diagrams: Free-body diagram and Kinetic diagram .....	129
Figure 81: Equivalent simple spring mass system for link 2 .....	130
Figure 82 :Link 2 cross sectional area.....	131
Figure 83: Link 2 specific displacement schematic .....	132
Figure 84: Laplace transformation block diagram .....	134
Figure 85: Element nodal displacements and constraint.....	135

**List of Abbreviations**

AMM	Assumed Mode Method
FEM	Finite Element Method
LQG	Linear–Quadratic–Gaussian Control
LQR	Linear–Quadratic Regulator
MATAB	Mathematics Laboratory
PID	Proportional Integrator Derivative

## **Chapter 1: Introduction and Literature Review**

### **1.1 Introduction and State of Art**

#### **1.1.1 Applications of Rigid-Flexible Manipulators**

The implementation and use of robotic arms provide precise displacement that serves a wide range of sectors, such as the fields of medical operations, industrial manufacturing, military, and civil defense industries. All these sectors implement robotic arm technology for running moderate and critical tasks. Most of the robotic manipulators tend to reduce vibrations at the manipulator tip to achieve accurate positioning, accordingly, to reach the desired tip position, a detailed structure must be designed and modeled by maximizing the surface stiffness. The use of this high stiffness value causes great negative effects in the system associated with heavier material and large design dimensions. These heavy and rigid manipulators result in poor efficiency which in turn reduces the speed and increases the power consumption concerning the payload [1].

A high accuracy robotic arm needs a proper maintenance during operation to restrict dynamic deflection. Arm vibrations occur during high-speed motions in most industrial manipulators. Maintaining the manipulator's high productivity during the industrial manufacturing process will require reducing the manipulator arms weight, and increasing their operation speed, or both [2].

Generally, recent robots used in modern life can lift only about one-twentieth of their own weight, in comparison with the human arm which can lift about ten times of its own weight. The tip slew velocity of a robot arm is typically around 40 inches per second while that of the human arm during a task (such as throwing a baseball) 1

is around 1500 inches per second. But still there is opportunity to improve the existing performance of the robotic manipulators. The two major problems in robotic manipulators are accuracy and control. As far as the accuracy is concerned, the repeatability of present robots is of the order of 1 mm over a working space, the accuracy of absolute positioning, i.e. for the end effectors to reach the required point may be off as much as 1 inch. The solution to such a problem is to make robots very stiff and rigid.

It is preferable to build a lightweight robot arm which has a longer reach and capability of carrying heavy payloads during a rapid motion. The robot arm has to be flexible to achieve the above-mentioned requirement, accordingly static deflection of the robot arm has to be considered for positioning accuracy. Large scale inertial forces are generated by acting on a high-speed moving arm which in turn degrades the stability of the robotic arm, which necessitates the design of a more sophisticated control system.

Flexible manipulators are suitable for incorporation into micro devices. Therefore, the demand for lightweight manipulators requires further research focusing on new materials, more accurate dynamic equations, and design of advanced control systems for manipulators.

### **1.1.2 Advantages of Rigid-Flexible Manipulators**

The rigid-flexible manipulators have many advantages; here are some of them listed below:

- They have a lower cost.
- The work volume is larger in these systems.

- Their operational speed is quite high.
- They have a greater payload than the rigid manipulator.
- Their energy consumption is quite low.
- They have better maneuverability, and their transportability is also high.
- Their inertia is low, so they are safer.
- They have a smaller actuator in their systems. On the other hand, their greatest problem is the high vibrations which are due to stiffness at high values.

## 1.2 Research Objectives

The focus of this research is to model and control a two-link rigid-flexible manipulator without payload and damping effect. The manipulator is moving in the horizontal plane, which makes the effect of gravity neglected.

To make the manipulator model approachable by control techniques, the model was presented in the state space form. The model is then simulated by MATLAB Simulink, in which its behavior is observed, recorded, and investigated.

The objectives of this research are listed below:

1. Perform dynamic modeling of a two-link rigid-flexible manipulator using Lagrangian Equations of motion, and the Finite Elements Method as a discretization technique. The dynamic model is to be modeled using Simulink and MATLAB for the sake of simulation.
2. Implement various control systems applied on the dynamic model, using Simulink and MATLAB, the controllers to be used are conventional PID controller, LQR controller, and LQG controller. The behavior and response of each controller are to be compared and analyzed to decide which controller is best to be used in such manipulators.

3. Create a 3D model of the manipulator using Autodesk Inventor software, the model is then imported to MATLAB Simulink and controlled by the same types of controllers applied on the dynamic model generated mathematically. The dynamic behavior of the two models is to be analyzed and compared to validate the ability of Autodesk Inventor to be used as a reliable tool for generating 3D models of robots and importing them to MATLAB Simulink for the sake of simulation.

### **1.3 Expected Impact and Contribution of this Research**

Flexible manipulators and robotics, in general, are widely used in industrial filed, constructions, medical applications, and several other different areas. This work is expected to provide a better understanding of the dynamic modeling of flexible manipulators and their control methods. In particular, the prominent outcome of this research work is to analyze and compare between types of controllers used to control the general behavior of flexible manipulators; thus, to determine the type of controller that is most appropriate for a specific application, which will reduce time and effort when selecting the control system of robotic arms in any engineering field [3].

The main contribution of this research is the validation of Autodesk Inventor software as a proper tool to ease the modeling of 3D assemblies by linking them to MATLAB Simulink. If it was found out that the Autodesk Inventor serves as a good tool for this type of modeling, then this would make the design and control of such systems much easier than the numerical modeling techniques created either by MATLAB coding/programming or by Simulink blocks.

## Chapter 2: Literature Review

### 2.1 Relative Background

The exploration in the field of flexible manipulators goes from a single-link controller pivoting about a fixed hub to three-dimensional multi-link arms. Generally, most experimental works were restricted to single-link manipulators. This is because of the unpredictability of the system in multi-link manipulators, which came because of the increasing interactions of deformed and gross motions in one hand, and the increment of the number of degrees of freedom in the robotic arm in the other hand [4].

The geometric dimensions of robotic arms play a major role in the system in which the arm is to be used. For example, in the case of extra thin and long robotic arms, vibrations and mechanical deflections are strongly present, thus the geometric dimensions of such robotic arms have to be taken into consideration, especially when using these kinds of robotic arms in space applications and manipulation systems that require high-speed movements. To compensate for the geometric dimensions and for the sake of reducing mechanical deflections and generated vibration, a well-designed control system is to be integrated within the dynamic model of the manipulator to reduce any undesired deflections and/or vibrations .

The dynamic modeling of manipulators is generally generated either by forward-dynamics or inverse-dynamics to track or predict the position of the tip of the end-effector, along with each of these, a controlling system is to be designed and implemented to minimize vibration and deflection of the robotic arm.

## 2.2 Techniques Used in Analyzing and Modeling the Rigid-Flexible Manipulators

Several control strategies are analyzed and modeled by the researchers to solve the control problems of rigid-flexible manipulators. To model the manipulator, different types of progressing models can be used, the main objective of those models is to mathematically solve the equation of motion that represents the concerned model. The solution of the equation of motion depends on the energy principles that include the phenomenon of inertia stored as a type of kinetic energy and in form of gravitational forces elastic deflection, which is stored as potential energy.

There are two basic approaches to writing down the equations of motion, or modeling: first is the free body diagrams followed by force and moment balance; governed by Newton's second law which is described in the appendix, and the second is the energy method.

For mechanical systems and structures coupled to other physical fields, linear graph theory is also a useful modeling concept. For more complex systems, finite element modeling, boundary element modeling, and spectral elements are useful [5].

For the analysis of rigid-flexible manipulators, the following techniques are mostly used:

- Assumed Mode Method (AMM).
- Lumped Parameter Method.
- Finite Element Method (FEM).

In this part of the thesis, the methods listed above are to be briefly explained and discussed to provide a general overview of each of them to make the comparison

clearer to the reader. Both the Assumed Mode Method and the Finite Element Method rely on the spatial discretization technique to achieve a set of differential equations that are used to describe the motion of the mechanical object in general.

### **2.2.1 Assumed Mode Method (AMM)**

In the Assumed Modes Method, the deflection of continuous elastic structures is modeled by a finite series of space-dependent functions that are multiplied by specified time-dependent amplitude functions. The amplitudes become a set of generalized configuration coordinates, in the usual Lagrangian interpretation of generalized coordinates. The space-dependent functions are typically chosen as a complete set of linearly independent functions, selected to satisfy at least the geometric boundary conditions, and differentiate at least half as many times as the order of the system. Admissible functions are the most widely encountered choice for the basic functions in the Assumed Modes Method. However, if the problem formulation and a priori insight permit, the basic functions may be selected to satisfy; also, some or all of the physical boundary conditions, and are differentiated as many times as the order of the system. More restricted subset of admissible functions as comparison functions were referred to. That is, for a given number of functions, expect better accuracy in approximating the true system dynamics when using comparison functions instead of admissible functions is expected. While better accuracy is expected, a judicious admissible functions. While the best space-dependent functions are usually the system's Eigenfunctions, which satisfy not only all boundary conditions but also the rigorous spatial differential equations, these are not always available. This is because the rigorous partial differential equations must usually be formulated and solved

exactly to obtain the Eigen functions and it is impossible to carry these developments to completion for complicated structures.

### **2.2.2 Lumped Parameter Method**

Many physical systems cannot be modeled successfully by the single-degree-of-freedom model that is, to describe the motion of the structure or machine, several coordinates may be required. An approximation of distributed parameter systems is considered in the lumped parameters method [5].

To simply compare between the lumped parameter system and the distributed parameters system, it could be stated that the respected dependent variables in the lumped system are described as functions of time alone; generally, this requires a set of ordinary differential equations describing the system to be solved. While in the case of distributed systems, the dependent variables are described as functions of time and one or more spatial variables. This will be represented by solving a set of partial differential equations.

Distributed parameter systems are modeled by sets of partial differential equations, boundary conditions, and initial conditions, which describe the evolution of the state variables in several independent coordinates, e.g. space and time. Most distributed parameter models are derived from first principles, i.e. conservation of mass, energy, and momentum.

Whereas first principles allow a model structure to be defined, some degrees of freedom are usually left for model parameterization, and some unknown parameters have to be estimated from experimental data by minimizing an output error criterion measuring the deviation between real system and model outputs.

### 2.2.3 Finite Elements Method (FEM)

The Finite Element Method is a numerical technique for solving problems which are described by partial differential equations or can be formulated as functional minimization. Approximating functions in finite elements are determined in terms of nodal values of a physical field which is sought. The elements are assumed to be connected to one another, but only at interconnected joints, known as nodes. It is important to note that the elements are notionally small regions, not separate entities like bricks, and there are no cracks or surfaces between them.

In Finite Element Method, there are two important features to be considered while solving the partial differential equations that represent the physical quantity or model:

- 1- Number of elements discretizing the physical quantity play a major role in the precision of this method even with simple quantities.
- 2- Locality of approximation play a great rule in solving systems with great number of nodal functions, by leading to meager the system equations of a discretized problem.

The procedure of solving physical parameters or models using the Finite Element Method could be summarized in the steps below:

- 1- Discretization of continuous quantity, this is achieved by dividing the continuum into separate divisions which are defining the finite elements. Those elements are defined by matrices representing the elements and nodal functions connectivity.
- 2- Definition of interpolation functions, in this step interpolation function are being defined to describe the field variable over the element. Interpolation

function are generally polynomial function with a degree depending on the number of nodes assigned to the function.

- 3- Definition of element properties, a finite element matrix is being generated in this step to relate the nodal functions of the unknown functions to the parameters. This step could be completed by several techniques such as Galerkin method and the vibrational approach.
- 4- Finding the global equation for the whole system region. This is done by assembling and combining all the equation for the local elements used in discretizing the system with the aid of nodal functions that serve as connections. An important consideration before solving the general equation in this step is imposing the boundary conditions that were not accounted before in the definition of element properties (equations).
- 5- Solving the global equation of the system, this is done either by direct or iterative methods. As a result of this solution, the values of the nodal values of the requested functions are obtained in this step.
- 6- Computation of additional results. After solving the global equation system, some additional parameters are needed to be calculated in many cases. For example, calculating the stress and strain values in mechanical systems is required after finding the solution to the general equation of the system. This should be achieved in this step.

### **2.3 Previous Research**

Most of the dynamic models are formulated based on the consideration of forward and inverse dynamics. The computational efficiency is given the most consideration because of the accuracy and the lack of sophistication in the model being

formed. Here, predicting changes in the system dynamics of the manipulator, resulting from loading and configuration changes, has been proposed, where expectations of variations in frequencies and nodal functions or mode shapes can be completed without the necessity for a solution to the complete final determinant equation of the system. Taking into consideration that the flexible nature of the manipulator system is important for control purposes, and for the system modeling.

Various methods have been either used or combined to develop the model of flexible manipulators. These approaches differ in their characteristics, which are generally describing their complexity and accuracy. These models can be described and explained as follows:

- The method of Lagrange's equation and modal expansion which is also known as Ritz-Kantorovich.
- The Finite Element Method and Lagrange's equation.
- The method of Euler Newton equation and the method of modal expansion.
- The combination of Euler Newton equation and Finite Element Methods.
- The techniques of frequency domain and the use of single perturbation method.

For the sake of simplicity, models of flexible manipulators consisting of one link were created and presented by several researchers. Understanding the nature and behavior of such models will help in generating a better understanding of other complex models consisting of many more links.

Single link Flexible manipulator study was discussed using Lagrange's Principle, which Hashting and Book presented in their work [6]. Similarly, Tokhi and Azad [7] presented single link study with Finite Element Method.

In the methodology of Lagrange's equation and modal expansion method, summation of nodes is used to represent the deflection in the manipulator. In this method the product of two functions in each node was used, one of these functions depends on the distance of the manipulator length and the other one is a generalized co-ordinate which depends on time. To solve the problems which are related to the single link flexible control of the manipulator, Cannon and Schmitz [8] has used the Assumed Modes Method especially where the sensor is not actually collocated with the sensors of system.

The systems where the link deflection is supposed to be of a small size, models of flexible manipulators links are formulated by use of the recursive Lagrangian formulation and the transformation is usually represented by additional modal shapes. The closed form of the equations of motion of a planar robot arm were presented by Luca and Siciliano [9]. These equations were formed by using the Assumed Modes Method to get the results in the form of a linear model of a definite number of dimensions.

The accuracy and validity of the modal approximation was studied by Cetinkunt and Yu [10], in which they used a suitable feedback controller to control a single link flexible robot arm consisting of an infinitely large number of modes.

A Finite Element Model combined with Lagrange's approach was used to create a mathematical model for lightweight flexible manipulator in [11]. The links of the manipulator were treated as an assembly of a number of finite elements for which the potential and kinetic energies were derived using the Lagrange's approach. Afterwards, the dynamic model of the manipulator was derived by combining the elemental potential and kinetic energies. It was found out that the reasonable modelling

and analysis of the dynamic model of the manipulator will result to the use of the enhanced control systems to find a solution to some problems linked to the flexion nature of the manipulators arms being lightweight. A detailed model development of two-link manipulators system along with its simulation results were presented in this research [11].

The Assumed Modes Method is usually quite like the Lagrange's equation and finite element method. The displacement and/or slope at specific points (nodes) are the generalized coordinates along the manipulator [11]. However, the Lagrange's formulation is mostly used since its matrix structure readily allows the dynamic model to be reformed in a state-space form that is convenient for control purposes. The (4 x 4) transformation matrix along with the Lagrang's approach was used by Sunada and Dubowsky [12] to model manipulators with elastic nature.

The available and normally used Finite-Element programs are mainly used to compute the mass and flexibility properties of the links. One efficient method used for the study of the dynamics of elastic robots was developed by Bayo et al. [13]. This method was based on the principles of Lagrangian formulation and on the finite-element method. De Smet et al. [14] and Pereira and Proenca [15] presented the analysis of the flexible structure with the help of the Finite-Element Model which was effectively reduced by applying the Component Mode Synthesis techniques.

Raksha and Goldenberg [16] used the Euler-Newton's method, which is another method of calculating system dynamics. In this method, Newton's second law is used to formulate the equations of applied forces and balancing them with rate of change, in both linear and angular momentums, rather than using Lagrange's equations.

In the study of forward dynamics, angular and linear momentum are unknown variables of the system, while the actuator forces are known. Representing the former terms using assumed modes or finite elements will lead to a dynamic model in which these modes/elements are related to external forces in terms of time dependency.

The basic technique in the Euler-Newton and the assumed mode method, is to divide the manipulator into several elements and separately perform a dynamic balance on each element. This is a very difficult process especially for great number of elements, however, it would be pretty much easier to include non-linear effects without affecting the basic model.

In the method of singular perturbation, the system's characteristic modes are divided into two clear groups: a string of fast or high-frequency modes and a group of slow or low-frequency modes. In the scene of flexible manipulators, the fast modes which are flexible ones, while the slow modes are the rigid body modes. Accordingly, the dynamics of the main system can be divided into two subsystems, a slow sub system which is equivalent to the rigid sub system, and a fast sub system in which the slow variables are set to be constant parameters.

Frequency-domain analysis of the model could be used as another technique for modelling the manipulator in the time-domain. This method uses Bernoulli-Euler beam equation used for the analysis of uniform beam to develop a transfer matrix to represent the model. The disadvantage of this method is that it does not allow for the interaction between the dynamics and the gross motion of the flexible manipulator to take place, nor can these effects be easily indulged into the model. Because of this, the model can only be called as approximate [17].

A model of ordinary differential equations was created by Tomei et al. [18] based on Ritz method and the Hamiltonian approach. The Ritz method was also applied by Park and Park [19] who presented an approach, which is developed on the basis of Fourier series in order to generate a path for the flexible manipulator by the careful selection of the coefficients of each harmonic term to reduce the remaining vibrations.

Hohenbichler et al. [20] discussed the case of the two-link flexible planar arm, where he compared between the Finite Element Method in the time domain. Bremer and Pfeiffer [21] tested and experimented some proposed algorithms to study the control problem and behavior of a flexible manipulator.

Mustafa and Al-Saif [22] represented three PID Controller Techniques, which were applied on the model of 2-R Robot that was designed using Autodesk Inventor then imported to the Simulink. Researchers in this paper explored the movement control of a 2R manipulator through examination and computer simulation. Nonlinear equations of motion were created utilizing the assumed mode strategy. Three different models were presented, one of which was created based on the Autodesk Inventor and MATLAB Simulink, the two others were created based on the linearization technique around the point of equilibrium of the manipulator and the other one was created based on the linearization feedback of the robot. A comparison between the behavior of the three models was made to evaluate the effectiveness of each of them.

A survey of two-link flexible manipulators was presented by Lochan et al. due to its various advantages and uses. The researchers discussed the various features of flexible manipulators that were reported in the literature. The presented work focused

on the dynamic analysis, modelling methods and the difficulties faced in the control system used in each manipulator [2].

Complexities in both modelling and control techniques were also discussed by Lochan et al. Researchers classified the two-link flexible manipulators based on the modelling techniques and the nature of the controller used. Another classification was made based on whether the reported work was made based on experimental work or model simulation [2].

Another review study of flexible manipulators was presented by Alandoli et al. [23]. Researchers presented the mathematical modeling and control techniques of various types of flexible manipulators starting from single-link, going through the two-link manipulators, and ending with multi-link manipulators. The effectiveness of some types of controllers used to control flexible manipulators was also presented in this review. Some of which was a control scheme based on the Model Predictive Control (MPC) alone and another technique was demonstrated with MPC combined with Piezoelectric Actuators (PZT). Another controller consisting of a combination of robust nonlinear and fuzzy compensator was also presented, in which a piezoelectric actuator was employed by considering their position along the link. Other controllers such as Linear Quadratic Regulator (LQR), and fuzzy logic controllers were also presented. Researchers in this paper concluded that the latest two types of control approaches had better performance in terms of controlling the flexible manipulator and in vibration suppression compared to the other control schemes [3].

Major research and development contribution in investigating the behavior and modelling of flexible manipulators has been taking place, without considering the flexibility characteristic of both the link and the joint in most cases. However,

neglecting the effect of flexibility will reduce the complexity of the system; this results in neglecting the behavior of the elastic manipulators. In such cases, bending, torsion, and damping that take place in the links will not be addressed, which can reduce the reliability in the model. This emerging area invited further research to be conducted while considering the importance of adhering to the concept of flexible manipulators. Since then, researchers have been studying the effect of vibration and damping commencing in the links that form the flexible manipulator [3], [24], [25].

The most conventional types of controllers that have been widely used with flexible manipulators are the PID and fuzzy PID controllers. Both controllers have advantages and disadvantages, a major disadvantage is the requirement to perform the tuning process of the  $K_p$ ,  $K_i$ , and  $K_d$  gains of the PID controller. This is considered a time-consuming iterative process, which cannot be applied on non-linear systems and systems with noises and disturbances. Modified types of controllers were created and modified in order to overcome this problem, these controllers include interval 2 type fuzzy compensated PID [26], adaptive PD-PID controller [27], self-tuning fuzzy logic PID controller [28], gain-schedule PID controller [29], intelligent PI controller [30], fuzzy fractional order PID controller [31].

In almost all research, a comparison was made between one controller of the conventional type: mainly PID, with a new proposed controlling technique. In nearly all cases, the new proposed techniques have outperformed the conventional controller [24], [32]. Thus, new research area can emerge depending on previously conducted research on new generation controllers.

In Mamahood and Pedro [27], a controller type called adaptive hybrid PD\_PID was used to control the movement of a two-link flexible manipulator. The PD gains in

the adaptive PD-PID controller were constantly changed by using a simple law, then the control system was applied on a two-link flexible manipulator. Then the whole model was simulated using MATLAB Simulink. The results obtained from the simulation were compared to a hybrid PD\_PID controller without the adaption mode, and results obtained concluded that the adaptive controller has proven its ability to suppress vibrations in the model and proved its ability to maintain the robustness of the flexible manipulators in case of white noise, disturbances, and variation in the input signal.

Dehghani and Khodadadi [28] addressed in their study another type of controller that includes a combination of a fuzzy logic controller and PID controller to control a single-link flexible joint robot, which is called a self-tuning fuzzy logic PID controller. It is known that the flexibility of the link and joint in flexible manipulators is a major cause of system uncertainty and nonlinearity. Regular PID controllers could be used to control flexible manipulators with flexible joints, in such a case the  $K_p$ ,  $K_i$ , and  $K_d$  gains are to be determined using various methods, such as the Ziegler-Nichols method (taken into consideration that they must remain constant). However, this will produce high link movement speed with great overshoot, which makes this method undesired in trajectory tracking. In the self-tuning Fuzzy logic PID controller, the controller will keep producing fuzzy  $K_p$ ,  $K_i$ , and  $K_d$  gains values throughout the tuning process, causing a high-speed response of the flexible link, with a minimum value of the overshoot to become close to zero. A comparison between the regular PID controller and the STFL-PID controller was made, and it was concluded that the STFL-PID controller has a better performance.

A single link flexible controller was presented using Lagrange's model by Narayana and Ramesh [1]. The model was controlled using three types of controllers: classical PID controller, Pole Placement controller, in which the poles of the closed loop system are found and determined in the plane, (poles are the gains which are set to make the non-linear dynamic model as a linear one), and LQR (Linear Quadratic Regulator, which is a special case of pole placement controller, in which the poles of the closed loop system are set to their optimal values). A comparison was made between the three types of controllers. With the aid of MATLAB and Simulink M-Files and S Function blocks, it was found that the LQR has the better results in obtaining the most accurate rotational angle and the tip movement.

Vesely's and Ilka [29] reported another comparison between Classical PID controller and gain scheduled PID controller. A conclusion was made that the gain-scheduled PID controller has improved the performance of the closed-loop system dramatically compared to the classical controller, and it also has proven that it minimizes the cost of the system.

Other researchers studied various types of controllers with flexible link manipulators, an intelligent PI controller was presented by Agee et al. [30] and compared results with conventional PI and PID controllers. Results have shown that the PI outperforms the conventional PI and PID controllers in tip trajectory for sinusoidal input signals; it also has better performance in eliminating noises and disturbances and noise in the system. This indicates that PI controller has great potential in improving the tracking accuracy and reducing disturbances and noises in the system compared to classical PI and PID controllers, which indicates that the PI controller can also be successfully used in multi-links flexible manipulators.

For further comparisons, a single-link flexible manipulator was controlled in two methods, conventional PID, and State feedback controller, which is also called pole placement control. Simulation and experimental results were compared for the two controllers and results obtained have shown that the state feedback has a slight outperformance over the conventional PID controller in achieving less deflection in the end effector through the tip trajectory, and in keeping the rotation angle of the link at the desired value [32].

Lee S. and Lee H. [33] implemented composite PID controller, which was created by adding disturbance observer to the slow subsystem, and a modal feedback to the fast subsystem, these two added components have made up the composite controller. Which has been tested on a flexible single-link manipulator. The obtained results were satisfactory and have led to the conclusion that they can be used in more complex flexible manipulators. Keeping in mind that the PID controller was suppressed to a special tuning process, in which the three gains were changed and the effect of this change on the performance of the closed-loop control system was observed and proven to be reliable compared to the classical PID controller.

Current research has dealt with new areas of research in the field of flexible manipulators controlling systems, some of which are presented by Gaidhane [26], Kumar [31], and Talebi [34]. All of them proposed different controlling systems that are applicable in non-linear and much more complex dynamic models. One is the interval 2 type fuzzy compensated PID controller with footprint of uncertainties in membership functions. This controlling method has proven its ability to overcome the uncertainties in a 2DOF robotic arm caused by variable loading conditions, damping forces, disturbance to signals, and random noises in the system. Results were compared

to interval 1 type fuzzy controller compensated PID controller, and they were comparable. Due to the highly constrained dimensional model proposed by Gaidhane [26], a recently proposed optimization technique called a hybridized GWO-ABC algorithm has been implemented. This algorithm has proven its ability to be used in solving optimization problems in both low and high-constrained high-dimensional models.

Another type of new fuzzy fractional order PID controller was used to control a redundant six DOF SCARA and reported by Gaidhane [26] and it was compared to the conventional Fuzzy-PID and PID controllers. The new FO Fuzzy PID controller showed its ability in achieving better tip trajectory tracking and overcoming the associated system disturbances. The system is well known for its high non-linearity, therefore, optimizing the gains in the proposed controller required following ABC optimizing algorithm. Their study concluded that the performance of the new fuzzy PID controller was robust and effective when compared with conventional PID and Fuzzy-PID controllers.

Talebi et al. [34] and Kumar et al. [31] investigated the effect of using neural network to control flexible manipulators. In their study the performances of two controllers were investigated, both consisted of a neural network tracking controller and a stabilizing joint PD controller. The first model was developed to achieve better tip tracking and minimize link deflections, and the second model included tracking a point along the arm of the flexible manipulator to eliminate difficulties associated with the output feedback control of the manipulator itself.

Research was also reported by Wie and Soffker [35] addressing the effect of closed-loop PID controllers on rotor movement using Active Magnetic Bearing

(ABM) to control the rotation of a rotor, which are usually controlled by conventional PID. This methodology appeared time-consuming due to the tuning process that takes place through many iterations. To optimize the performance of the PID controller controlling an AMB connected with a rotor, a new optimization technique was introduced. This technique includes multi-objective genetic algorithm that concentrates on frequency and time-domain-base to achieve optimization. Enhanced results were reported after comparing the new optimized system with the conventional one. This indicates that this is a very useful controlling method. Abe [24] and Jiang et al. [36] presented other optimization methods.

It is known that a time-delay in the transformation of movement could occur due to the flexibility of the joint in the manipulator. Research was conducted by Jiang et al. [36] using a slow-fast response system that uses an optimization method called Geometric Singular Perturbation method, in which two quantitative forces were derived from the negative feedback of the link and used to determine the intersection between the slow and fast time-delay responses. It was concluded that for slow small delay response, the oscillation of the link was successfully suppressed, while on the other hand, for large delay response, the oscillation has increased to an unsatisfactory amount.

Moreover, a mandatory aspect when modelling the flexible manipulators is vibration displacement. Accuracy may be reduced due to system vibrations which leads to a decrease in engineering system efficiency and reliability. This aspect was addressed by Khot et al. [25] using cantilever beam system. In their study a cantilever beam mathematical model has been imported from ANSYS model to a MATLAB model, in which a piezo-electric sensor/actuator has been used along with a PID

controller to simulate and control the effect of vibration to the system. The obtained results from the two models in both ANSYS and MATLAB were comparable, which indicates that the conversion process happened to be successful and reliable. In conclusion, vibration control in this system, and many more complex real-life systems could be derived and simulated in the same manner, which will in turn, reduce the time consumed in mathematical calculations.

A new optimization technique was used to minimize the vibration and deflection caused by the bending of the flexible link in the model created by Abe [24]. This method served vibration suppression in a flexible manipulator consisting of two rigid-flexible links. The author implemented a technique called Particle Swarm Optimization, by which a great efficiency in the tip trajectory and in vibration suppression of the flexible link of the manipulator was presented.

The main advantages of the rigid-flexible manipulators are lower cost, larger work volume, higher operational speed, greater payload to manipulator, lower energy consumption, better maneuverability, better transportability, and safer operation due to reduced inertia and smaller actuator [37]. But the greatest problem of these flexible manipulators is vibration due to low stiffness [38].

## Chapter 3: Mathematical/Dynamic Modelling

### 3.1 Approach

Flexible manipulators are of great importance where robotics is required in different industrial fields. Controllers of flexible manipulators are also important due to their ability to reduce vibrations and minimize the undesired movements of the end effectors and therefore can achieve better accuracies. Some of these applications can be as simple as pick-and-place robotics, ranging to micro-surgery robotics and robots used in applications with space air craft [23], [39].

The methods of dynamic modelling and control of flexible manipulators could be classified as many types. One of them is the flexibility of links used in the design, which can be addressed via dynamic modelling of the manipulators. Another classification is the type of controller used to control the manipulator [23].

### 3.2 Arm Geometry

The robotic manipulator parts specified to demonstrate the geometry and classification for this manipulator type will be used to develop the following kinematic and dynamic analysis. Figure 1 illustrates schematics for three revolute joint manipulators, such that axes  $z_0$  and  $z_1$  are orthogonal for joint 1 and 2, respectively, and for joints 2 and 3 axes  $z_1$  and  $z_2$  are parallel, respectively. The input torque will be applied at the revolute joint at  $z_2$  axes shown in Figure 1 [40].

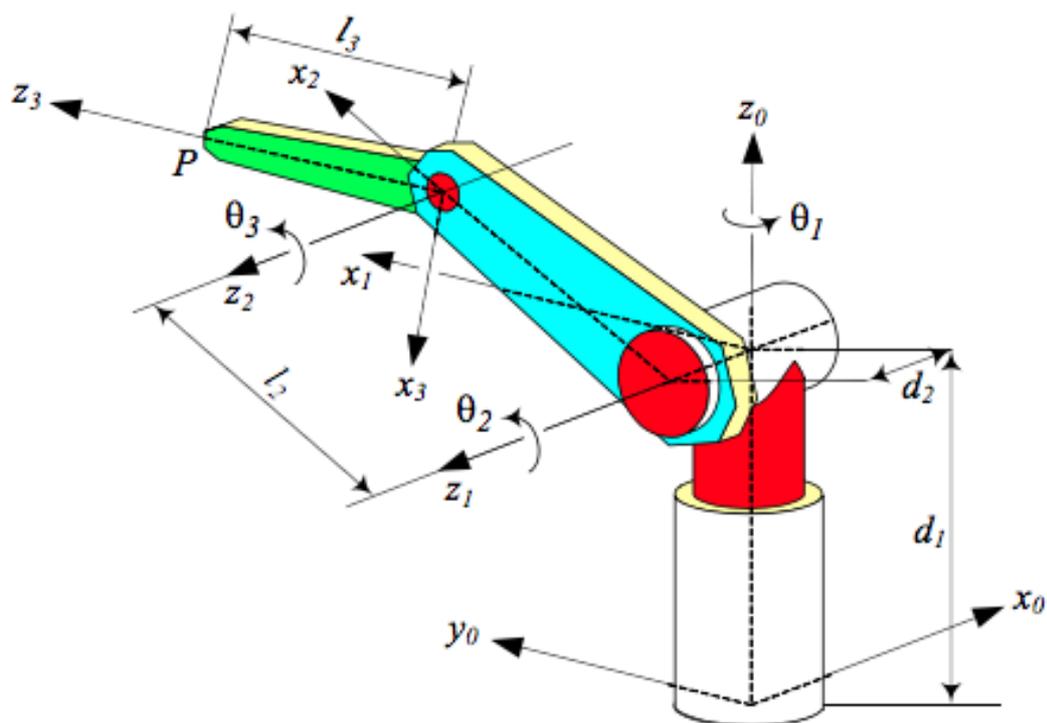


Figure 1: Schematic for three revolute joint manipulators

### 3.3 Modeling of the Rigid-Flexible Manipulator

In this research Finite Element Method will be implemented to conduct numerical prediction for flexible link displacement and rotation. The equations of motion will be defined accordingly along with associated boundary and initial conditions for the two-link manipulator system based on Lagrange approach. The PID controller system will provide data showing the enhanced system response for torque signal input that is applied on the hub of the first joint. The first and second links will be considered as rigid-flexible links, respectively, and the input torque signal is applied at the hub of the first link.

Various techniques can be used for rigid flexible manipulator analysis. The equation of motion for this manipulator can be performed using different progressing

models that depend on energy principles associated with the inertia stored as kinetic energy, and gravitational forces elastic deflection stored as potential energy.

Three techniques are used in analysis of rigid flexible manipulators namely, Assumed Mode Method (AMM), Finite Element Method (FEM) and lumped parameter method.

A schematic diagram of the rigid-flexible manipulator model addressed in this research is shown in Figure 2. The rigid link has a length  $L_1$ , with flexure rigidity  $EI_1$  and mass moment of inertia about the hub 1 labeled  $J_1$ . The flexible link has length  $L_2$  and flexural rigidity  $EI_2$ . There is a hub between the ground and the fixed link with mass  $m_{h1}$  and mass moment of inertia  $J_{h1}$ . The second hub connected between rigid link and flexible link has mass  $m_{h2}$  and mass moment of inertia  $J_{h2}$ . A tip mass load with mass  $M_p$  represents the payload of the manipulator and is attached to the end of the flexible link. The Bernoulli equation is satisfied, with small shear and small rotary inertia.

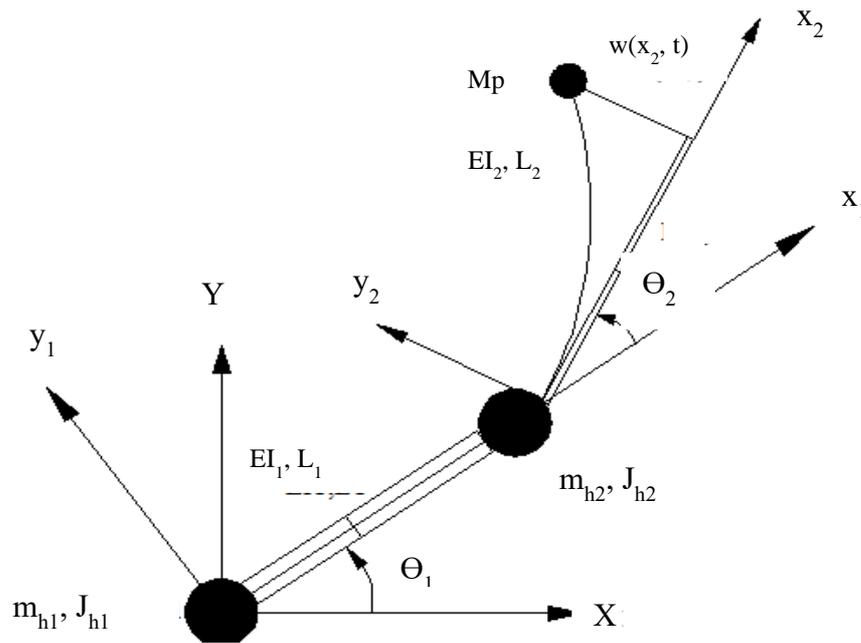


Figure 2: Rigid flexible manipulator

The hub<sub>1</sub> and hub<sub>2</sub> are assumed to be lumped masses with negligible radii.

The linear transverse displacement of an arbitrary point on the flexible link can be written as:

$$y(x_2, t) = L_1\theta_1(t) + x_2(\theta_1(t) + \theta_2(t)) + w(x_2, t) \quad (1)$$

Given that the angles are small, so the sin function is ignored.

The finite element method solution provides reliability in addressing complexity of such flexible manipulator systems. This is associated with the reason that the system is divided into finite elements in which simpler linear displacements at nodes and nodal rotations are addressed in the equation of motion, which represents the schematic for the two-link manipulator with finite element distribution. A schematic for nodal displacement is presented in which displacement and rotation are

required to be addressed in this system and the elements that will be used for this purpose are beam elements. The beam elements for the two links of the manipulator along with the element nodal displacement schematic are shown in Figures 3 and 4 respectively.

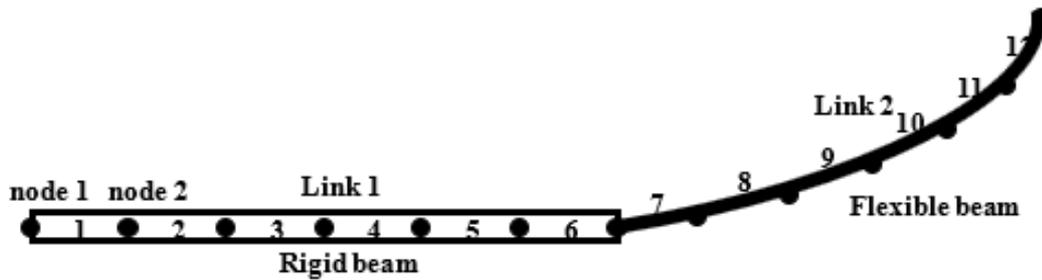


Figure 3: Beam elements for two links

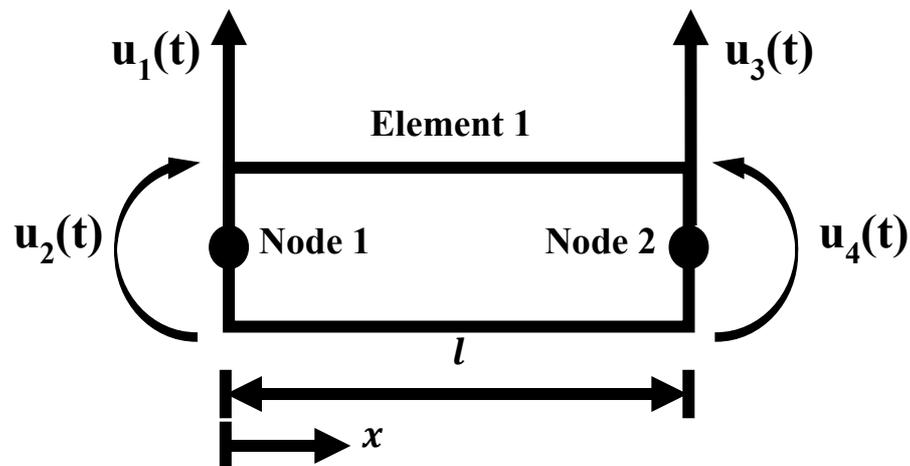


Figure 4: Element nodal displacement schematic

The transverse static displacement equation is defined via Euler-Bernoulli's Equation which addresses the relation between the load applied on the beam and corresponding displacement.

$$\frac{\partial^2}{\partial x^2} \left( EI \frac{\partial^2 u}{\partial x^2} \right) = 0 \quad (2)$$

Given that I indicate the moment of inertia term in Pa, and E is the flexural modulus in m<sup>4</sup>. The corresponding solution is shown in the Equation below:

$$u(x, t) = C_1(t)x^3 + C_2(t)x^2 + C_3(t)x + C_4(t) \quad (3)$$

Implementing the boundary conditions for the beam element in Equation (3) yields:

$$\begin{aligned} u(0, t) &= u_1(t) \\ u_x(0, t) &= u_2(t) \\ u(l, t) &= u_3(t) \\ u_x(l, t) &= u_4(t) \end{aligned} \quad (4)$$

correspondingly,

$$\begin{aligned} C_1 &= \frac{1}{l^3} (2(u_1 - u_3) + l(u_2 + u_4)) \\ C_2 &= \frac{1}{l^2} (3(u_3 - u_1) - l(2u_2 + u_4)) \\ C_3 &= u_2(t) \\ C_4 &= u_1(t) \end{aligned} \quad (5)$$

Substituting into the equation and rearranging it yields to

$$\begin{aligned}
u(x, t) = & u_1(t) \left( \frac{2x^3}{l^3} - \frac{3x^2}{l^2} + 1 \right) + u_2(t) \left( \frac{lx^3}{l^3} - \frac{2lx^2}{l^2} \right) \\
& + u_3(t) \left( \frac{3x^2}{l^2} - \frac{2x^3}{l^3} \right) + u_4(t) \left( -\frac{x^2}{l} + \frac{x^3}{l^2} \right)
\end{aligned} \tag{6}$$

Using the Finite Element Method, Equation (6) can be written in matrix form as shown in the below Equation (7):

$$u(x_2, t) = N_k(x_2)Q_k(t) \tag{7}$$

Where  $N_k(x_2)$  and  $Q_k(t)$  represent the shape function and nodal displacement of link 2 respectively. The shape functions of the beam element can be obtained as shown in Equation (8).

$$N_k(x_2) = [L_1 + x_2 \quad x_2 \quad N_{k1}(x_2) \quad N_{k2}(x_2) \quad N_{k3}(x_2) \quad N_{k4}(x_2)] \tag{8}$$

Where,

$$\begin{aligned}
N_{k1}(x_2) &= 1 - \frac{3x_2^2}{l_2^2} + \frac{2x_2^3}{l_2^3} \\
N_{k2}(x_2) &= x_2 - \frac{2x_2^2}{l_2} + \frac{x_2^3}{l_2^2} \\
N_{k3}(x_2) &= \frac{3x_2^2}{l_2^2} - \frac{2x_2^3}{l_2^3} \\
N_{k4}(x_2) &= \frac{x_2^3}{l_2^2} - \frac{x_2^2}{l_2}
\end{aligned} \tag{9}$$

For element  $k$  the nodal displacement vector is given as shown in Equation (10).

Andres San-Millan et al. (2015) defined the tip angle of a single link manipulator with a payload by the Equation 10 [41].

$$Q_k(t) = [\theta_1(t) \quad \theta_2(t) \quad d_{k1}(t) \quad \varphi_{k1}(t) \quad d_{k2}(t) \quad \varphi_{k2}(t)] \quad (10)$$

Where  $d_{k1}(t)$  and  $d_{k2}(t)$  are transverse deflections of the element and  $\varphi_{k1}(t)$  and  $\varphi_{k2}(t)$  are the corresponding angular rotations.

By defining  $s = x_2 - l(n - 1)$ , where  $l$  is the length of the element ( $l = L_2/n$ ), and  $n$  is the number of elements. The kinetic and strain energies associated with the beam element motion are described below in equations (11) and (13), respectively.

$$\begin{aligned} T(t) &= \frac{1}{2} \int_0^l A\rho(x) \left( \frac{\partial u}{\partial t} \right)^2 dx \\ &= \frac{1}{2} \dot{u}^T M \dot{u} \end{aligned} \quad (11)$$

The kinetic energy of an element 'e' can be expressed as shown in Equation (11).

$$\begin{aligned} T_e &= \frac{1}{2} \int_0^l \rho_2 A_2 \left[ \frac{\partial y(s, t)}{\partial t} \right]^2 ds \\ &= \frac{1}{2} \dot{Q}_k^T \left[ \int_0^l \rho_2 A_2 N_k^T N_k ds \right] \dot{Q}_k \end{aligned} \quad (12)$$

The potential energy of the element can be obtained as shown in Equation (13).

$$\begin{aligned}
 V(t) &= \frac{1}{2} \int_0^l EI \left( \frac{\partial^2 u}{\partial x^2} \right)^2 dx \\
 &= \frac{1}{2} u^T K u
 \end{aligned} \tag{13}$$

The element mass matrix can be obtained from Equation (11) as shown in Equation (14).

$$M_e = \int_0^l \rho_2 A_2 N_k^T N_k ds \tag{14}$$

where  $B_k = \frac{d^2 N_k}{ds^2}$

The element stiffness matrix can be obtained from Equation (13) as shown in Equation (15).

$$K_e = \int_0^l E_2 I_2 B_k^T B_k ds \tag{15}$$

Solving Equations (14) and (15) the element mass and stiffness matrices can be obtained as shown in Equations (16) and (17), respectively.

$$M = \frac{\rho A l}{420} \begin{bmatrix} 156 & 22l & 54 & -13l \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ 6l & -3l^2 & -22l & 4l^2 \end{bmatrix} \tag{16}$$

$$K = \frac{EI}{l^3} \begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \tag{17}$$

The mass and stiffness matrices for elements are presented below in Equations (18) and (19), respectively.

$$M_e = \frac{\rho_2 A_2 l}{420} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & 156 & 22l & 54 & -13l \\ m_{41} & m_{42} & 22l & 4l^2 & 13l & -3l^2 \\ m_{51} & m_{52} & 54 & 13l & 156 & -22l \\ m_{61} & m_{62} & -13l & -3l^2 & -22l & 4l^2 \end{bmatrix} \quad (18)$$

$$K_e = \frac{E_2 I_2}{l^3} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 6l & -12 & 6l \\ 0 & 0 & 6l & 4l^2 & -6l & 2l^2 \\ 0 & 0 & -12 & -6l & 12 & -6l \\ 0 & 0 & 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \quad (19)$$

Where,

$$\begin{aligned}
m_{11} &= 140(3L_1^2 + 6L_1 ln - 3L_1 l + 3l^2 n^2 - 3l^2 n + l^2) \\
m_{12} &= m_{21} = 70l(2l - 6ln - 3L_1 + 6L_1 n + 6ln^2) \\
m_{13} &= m_{31} = 21(10ln + 10L_1 - 7l) \\
m_{14} &= m_{41} = 7l(5ln + 5L_1 - 3l) \\
m_{15} &= m_{51} = 21(10ln + 10L_1 - 3l) \\
m_{16} &= m_{61} = 140l^2(3n^2 - 3n + 1) \tag{20} \\
m_{22} &= 140l^2(3n^2 - 3n + 1) \\
m_{23} &= m_{32} = 21l(10n - 7) \\
m_{24} &= m_{42} = 7l^2(5n - 3) \\
m_{25} &= m_{52} = 21l(10n - 3) \\
m_{26} &= m_{62} = -7l^2(5n - 2)
\end{aligned}$$

The size of either element i.e. mass and stiffness matrices is  $6 \times 6$ . The total kinetic and potential energies of the rigid-flexible manipulator by incorporating the payload and hubs inertias into the system are given by the Equations (21) and (22) below, respectively.

$$\begin{aligned}
T &= \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}J_{h1}\dot{\theta}_1^2 + \frac{1}{2}m_{h2}L_1^2\dot{\theta}_1^2 + \frac{1}{2}J_{h2}\dot{\theta}^2 \\
&\quad + \frac{1}{2}M_p \left( (L_1 + L_2)\dot{\theta}_1 + L_2\dot{\theta} + \dot{w} \right)^2 + \sum_{e=1}^n T_e = \frac{1}{2}\dot{Q}^T M \dot{Q} \tag{21}
\end{aligned}$$

$$P = \sum_{e=1}^n P_e = \frac{1}{2} Q^T K Q \quad (22)$$

Where  $Q(t) = [\theta_1 \ \theta \ d_{11} \ \phi_{11} \ d_{12} \ \phi_{12} \ \dots \ d_{n1} \ \phi_{n1} \ d_{n2} \ \phi_{n2}]$ .

The  $M$  and  $K$  matrices are the global mass and stiffness matrices of the manipulator. The  $M$  and  $K$  matrices are of size  $2(n+2) \times 2(n+2)$ .

As an example, using a single element,  $n=1$ , the mass and stiffness matrices size will be  $4 \times 4$ . The dynamic equations of motion of the rigid-flexible manipulator can be derived using the Lagrange Equation (23).

$$\frac{d}{dt} \left\{ \frac{\partial L}{\partial \dot{Q}} \right\} - \left\{ \frac{\partial L}{\partial Q} \right\} = F(t) \quad (23)$$

Where  $L = T - P$  is the Lagrangian and  $F$  is the external forces vector. Considering the damping, the dynamic equations of motion of the system can be obtained as shown in Equation (24).

$$M\ddot{Q}(t) + D\dot{Q}(t) + KQ(t) = F(t) \quad (24)$$

Where  $D$  is the global damping matrix.

For the rigid-flexible manipulator under consideration, the global mass matrix can be represented as shown in Equation (25).

$$M = \begin{bmatrix} M_{\theta\theta} & M_{\theta w} \\ M_{\theta w} & M_{ww} \end{bmatrix} \quad (25)$$

Where,

$$M_{\theta\theta} = \begin{bmatrix} m_{11} + I_{h_1} + \frac{1}{3}\rho_1 A_1 L_1^3 + m_{h_2} L_1^2 + I_{h_2} & m_{12} + I_{h_2} \\ m_{21} + I_{h_2} & m_{22} + I_{h_2} \end{bmatrix} \quad (26)$$

Where  $M_{ww}$  is the residual motion,  $M_{\theta w}$  is representing the coupling between the elastic degree of freedom and hub angle  $\theta$  and  $M_{\theta\theta}$  is associated with the inertia of the system about the motor axis.

The global stiffness matrix can be written as shown in Equation (27).

$$[K] = \begin{bmatrix} 0 & 0 \\ 0 & K_{ww} \end{bmatrix} \quad (27)$$

Where  $K_{ww}$  is associated with the elastic degree of freedom (residual motion), and the elastic degree of freedom do not couple with the hub angle through the stiffness matrix.

### 3.4 Damping Modeling

The damping coefficient plays a vital role in dynamic analysis of the system; however, the most effective way to treat the damping on the dynamic system is to treat damping values as viscous damping and frictions of the hubs, but we considered in this section the structural damping only called a Rayleigh Damping that have a form shown in Equation (28).

$$[D_{damping}] = \alpha[M] + \beta[K] \quad (28)$$

Where,

$$[D_{damping}] = \begin{bmatrix} 0 & 0 \\ 0 & D_{ww} \end{bmatrix} \quad (29)$$

Where,

$$[D_{ww}] = \alpha[M_{ww}] + \beta[K_{ww}] \quad (30)$$

Where  $D_{damping}$  is the damping matrix of the physical system,  $M$  is mass matrix of the dynamic system,  $K$  is the stiffness matrix of the system and  $D_{ww}$  is the sub matrix with the material damping. The Rayleigh damping coefficient given by Equations (31) and (32).

$$\alpha = \frac{2f_1f_2(\zeta_1f_2 - \zeta_2f_1)}{f_2^2 - f_1^2} \quad (31)$$

$$\beta = \frac{2(\zeta_2f_2 - \zeta_1f_1)}{f_2^2 - f_1^2} \quad (32)$$

Where  $\zeta_1$ ,  $\zeta_2$ ,  $f_1$  and  $f_2$  represent the damping ratios and natural frequencies of first two modes, respectively. The main advantage of using the Rayleigh damping is to reduce the vibration of the system.

### 3.5 State Space Representation

The mass matrix ( $M$ ) and the stiffness matrix ( $K$ ) are given by Equations (25) and (27), respectively. The state space representation is implemented with the equation of motion under applied force  $f_t$  as shown in Equation (33).

$$[M]\ddot{y} + [D_{damp}]\dot{y} + [K]y = f_t \quad (33)$$

$$x_1 = y$$

$$x_2 = \dot{y} = \dot{x}_1 \quad (34)$$

$$x_3 = \ddot{y} = \dot{x}_2$$

Substituting Equation (34) into Equation (33) to obtain Equation (35) as presented below:

$$[M]\dot{x}_2 + [D_{damp}]x_2 + [K]x_1 = f_t$$

$$\dot{x}_2 = -\frac{[K]}{[M]}x_1 - \frac{[D_{damp}]}{[M]}x_2 + f_t \quad (35)$$

$$\dot{x}_2 = -[M]^{-1}[K]x_1 - [M]^{-1}[D_{damp}]x_2 + f_t$$

The linear state space model is presented in Equations (36) and (37).

$$\dot{x} = Ax + Bu \quad (36)$$

$$Y = Cx + Du \quad (37)$$

Where,

$$A = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}D_{damp} \end{bmatrix} \quad (38)$$

$$u = [\tau_1 \quad \tau_2 \quad 0 \quad \dots \quad 0]^T \quad (39)$$

$$x = [\theta \quad \theta_1 \quad w_1 \quad \varphi_1 \quad \dots \quad w_n \quad \varphi_n \quad \dot{\theta} \quad \dot{\theta}_1 \quad \dot{w}_1 \quad \dot{\varphi}_1 \quad \dots \quad \dot{w}_n \quad \dot{\varphi}_n] \quad (40)$$

$$B = \begin{bmatrix} 0 \\ -M_{(n,1)}^{-1} \end{bmatrix} \quad (41)$$

$$C = [0 \quad I] \quad (42)$$

$$D = 0 \quad (43)$$

The parameters of link 1 and links 2 used for representing the mathematical model are shown in Table 1.

Table 1: Initial values for link 1 and link 2

Parameters of link 1		Parameters of link 2	
Length, $L_1$	0.9 m	Length, $L_2$	1.1 m
Mass density per unit volume, $\rho_1$	2710 kg/m <sup>3</sup>	Mass density per unit volume, $\rho_2$	2710 kg/m <sup>3</sup>
Cross sectional area, $A_1$	6.0833 x10 <sup>-5</sup> m <sup>2</sup>	Cross sectional area, $A_2$	6.0833 x10 <sup>-5</sup> m <sup>2</sup>
Young's modulus, $E_1$	7.11 x10 <sup>-4</sup> N/m <sup>2</sup>	Young's modulus, $E_2$	7.11 x10 <sup>-4</sup> N/m <sup>2</sup>
Mass (second) moment of inertia, $I_1$	52.530 x10 <sup>-4</sup> m <sup>4</sup>	Second moment of inertia, $I_2$	52.530 x10 <sup>-4</sup> m <sup>4</sup>

## Chapter 4: Control of Flexible Manipulator

The control of flexible structures has been an active field of research for several decades because of advancement in the space applications. Aerospace structures are usually constructed with lightly damped materials of low mass density. These structures are susceptible to the external disturbances which cause undesirable high frequency vibrations.

### 4.1 Modeling and Simulation

Dynamic and mathematical modelling of a two-link rigid-flexible manipulator will be performed in this research by using the technique of Finite Element method. In robot arms there are two type of movements. One type of movement is the shoulder movement of the arm and the second type of movement is the elbow movement of the arm. Firstly, in this project the dynamic model is explained by using the techniques of Finite Element Method and Lagrange equations.

Finite Element Method is the method which is used to convert the functions present in dimensional function (which is infinite space to functions in dimensional function which are in finite space), and at the end, ordinary vectors that are found with the help of numerical methods. The modelling of this manipulator is based on segmenting each link into a specific number of elements and then we study these elements individually. To find local maxima and minima values of each parameter in the equation of motion by using the Lagrange's method.

By solving the Lagrange equations, the dynamic model will be represented in state space form preparing it to be controlled by various control schemes. The final model is simulated by MATLAB and Simulink to investigate the behavior of the

system and its response to each type of controller scheme. So, in this chapter, the modelling of the system by using MATLAB Simulink will be discussed.

The flow chart represented in Figure 5 demonstrates the steps which are followed to model and control the flexible manipulator during the current and coming chapters.

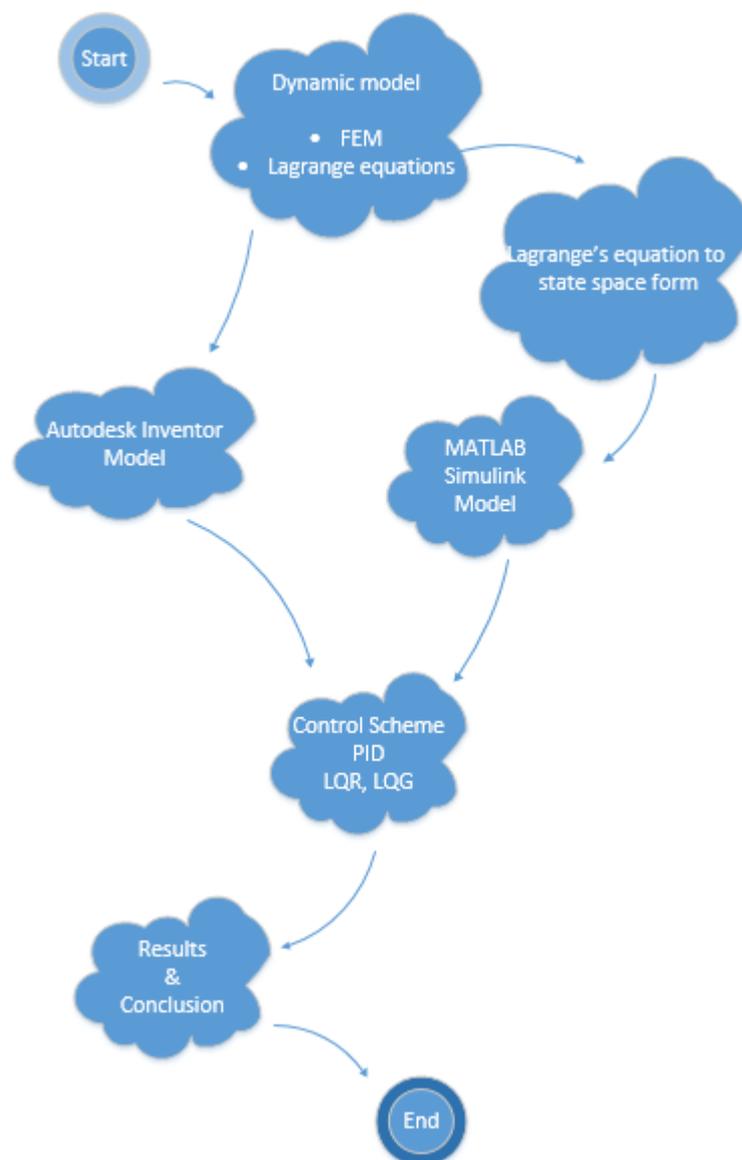


Figure 5: Flow chart of modelling methodologies

#### 4.1.1 Simulink Model

As a first step, a bang-bang signal (2 step or on–off signal) of amplitude 0.1 N.m was used as an input torque signal that was applied at the hub of link 1. However, For link 2, a bang-bang signal of amplitude 0.3 N.m, was used as an input torque signal that was applied at the hub of link 2, those bang-bang signals are shown in Figures 6 and 7 respectively. The Signal Builder is sending the torque signal as an input to the state space model of link 1 and of link 2. As a result of these applied input signals, the values of the parameters of manipulator are achieved accordingly.

As it can be seen from Figure 6 the torque input signal applied to link 1 will first rise until it reaches a value of 0.1 N.m, then it drops again to -0.1 N.m in a time of 0.75 seconds. The applied signal has a square type of signal, with a constant duration and peak values.

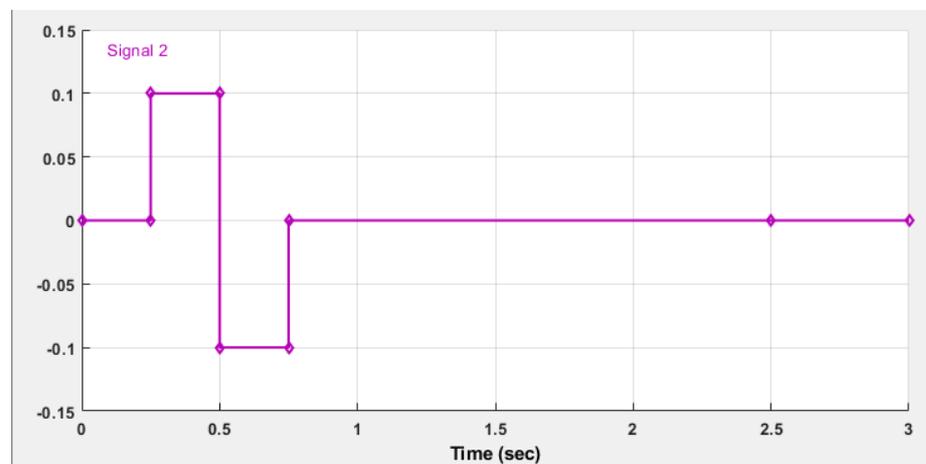


Figure 6: The input signal for hub 1

Figure 7 also represents the input signal of torque applied to link 2. It is clearly shown that the two-input signal are of the same square type, but with different peak

values, where the input torque at link 1 has a maximum value of 0.1 N.m, while the applied torque at link 2 has a maximum value of 0.3 N.m.

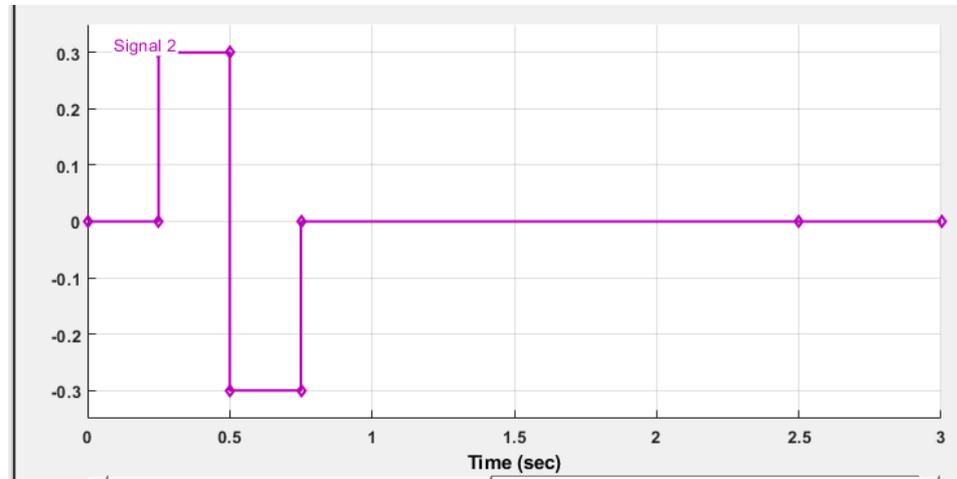


Figure 7: The input signal torque of hub 2

The simulink diagram of the model developed in this research is represented in Figure 8. It is a State Space model which was implemented and simulated on MATLAB/Simulink environment.

A bang-bang torque has a positive (acceleration) and negative (deceleration) period allowing the manipulator to, initially, accelerate and then decelerate and eventually stop at a target location. System responses were monitored for a duration of 1.5 seconds with a sampling time of 4 milli seconds. System responses and the sampling time was measured and elaborated in the MATLAB Simulink Model.

All the simulation parameters were built and implemented in the Simulink model, where the initial values were substituted in the matrices generated by the mathematical model, which are matrices A1, B1, C1 and D1 for link 1 and matrices

A2, B2, C2 and D2 of link 2. Completed code in M file “Link\_1.m” and “Link\_2.m”  
(See Appendix A).

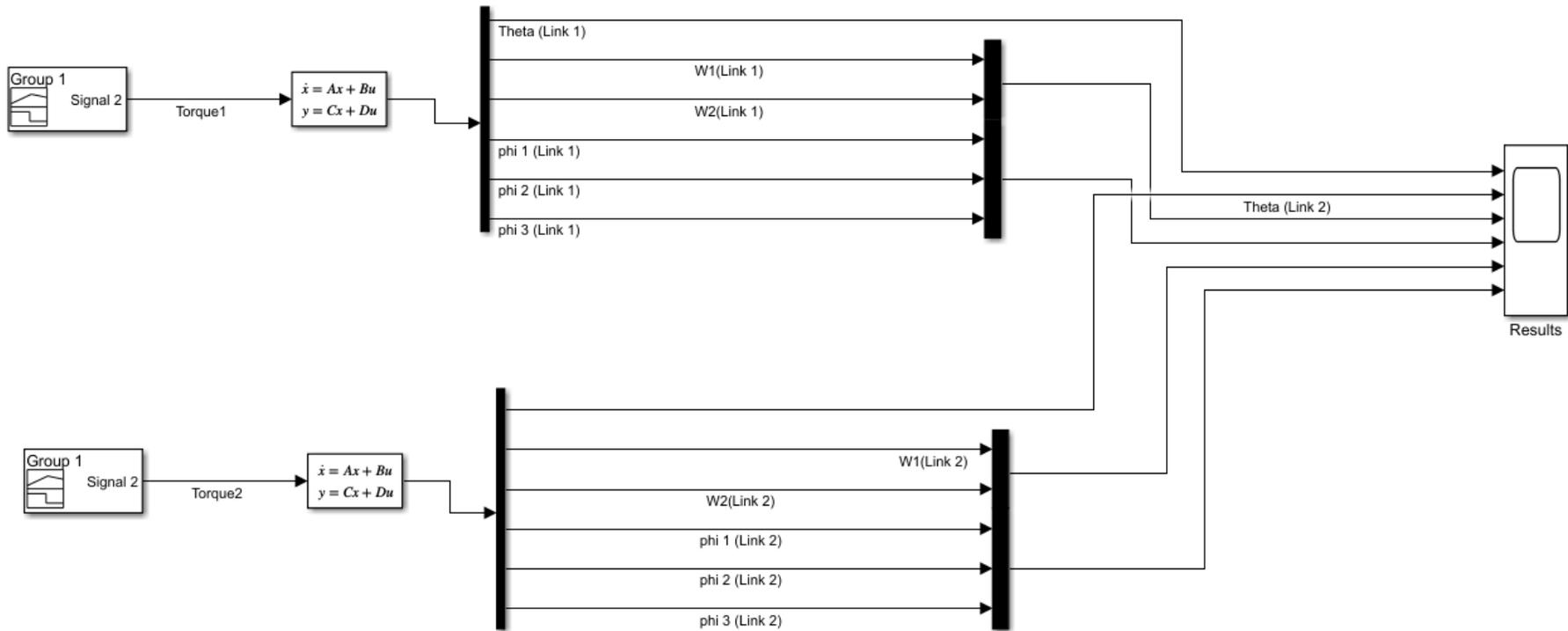


Figure 8: Simulink diagram

### **4.1.2 MIMO System**

The proposed system has two input torques and multiple output controlling variable. so, the system which have multiple inputs and multiple outputs are MIMO system. The inputs to the proposed MIMO system are the deflecting torque which activate the manipulator to deflect in desired angle  $\theta$  which is the output of the manipulator. In the control algorithm the controller tries to minimize the damping and overshoot in the response of the manipulator and minimizing the error signal which deteriorates the expected output response to achieve a specific job. The movement of the manipulator is not only in one direction i.e., rotational but also some translational displacement which is considered to be the output variable for the manipulator. So, it is concluded that the proposed system is MIMO with multiple inputs and multiple output variables.

### **4.2 Simulation Results**

Results are the primary objective of scientific research: data is collected, and statistical analysis of the data is done. The observations and measurements are recorded while simulating the model which are described in Chapter 3 which answers and predicts the questions raised in the introduction and any hypotheses formulated there. Flexible link manipulator results are reported in the diagrams presented in this section.

In this section, the simulation results of the MATLAB model will be discussed, starting from the results achieved without any controller scheme being used on the simulator, followed by the results achieved when using the three proposed controller schemes.

### 4.2.1 Simulation Results Without Any Controller

Based on mathematical modelling and the simulation described above, the following results are obtained. For rotational angle of link 1, the simulation result is shown in Figure 9. While the simulation result of deflection of nodes in link 1, which is  $w$ , is shown in Figure 10.

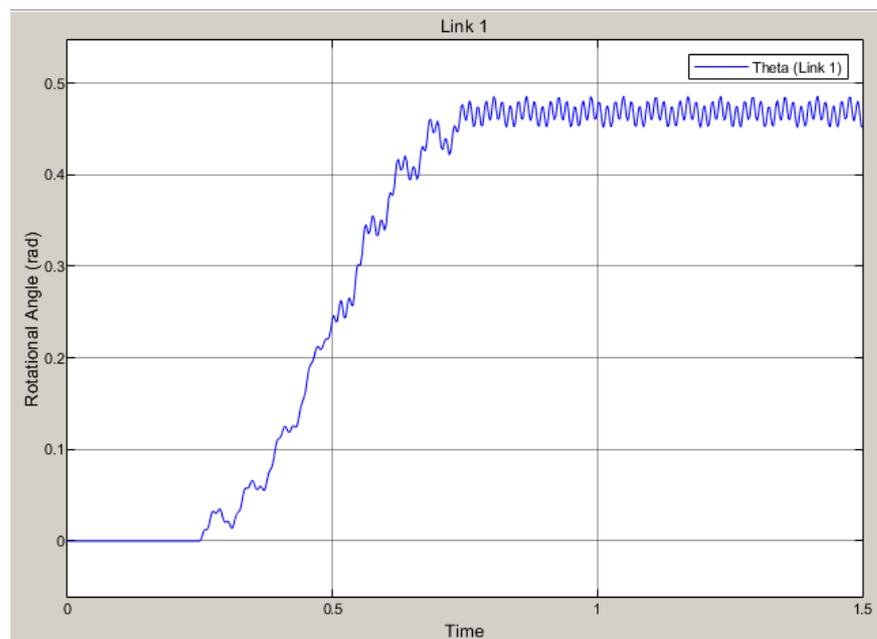


Figure 9: Rotational angle ( $\theta$ ) of link 1 without a controller

Figures 10 and 11 represent the parameters of link 1. In these graphs the amplitude varies indirectly with respect to time. Afterwards, a drop in the value will occur as the wave oscillates down and then rises after some time to a certain amplitude and then remains constant and continues oscillating with the increment of time.

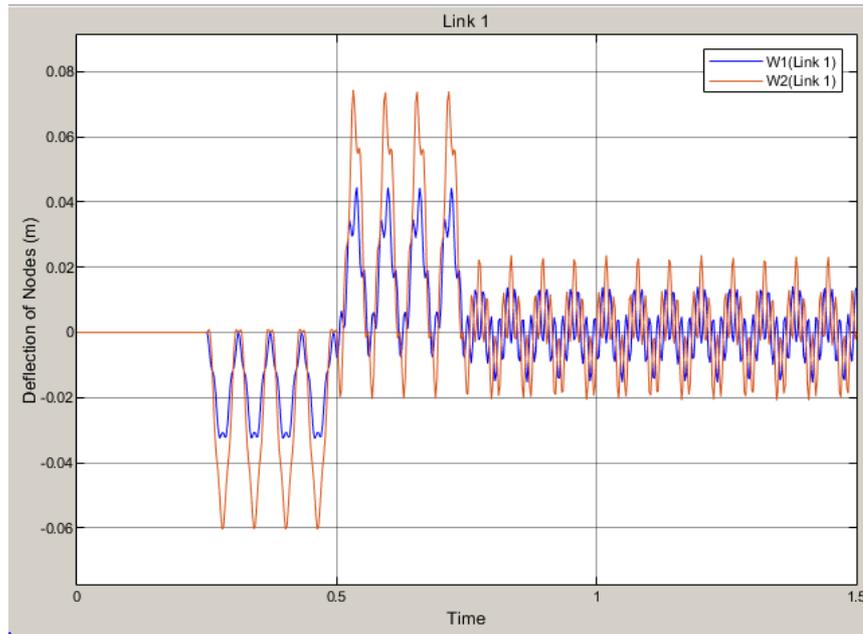


Figure 10: Deflection of nodes of link 1 without a controller

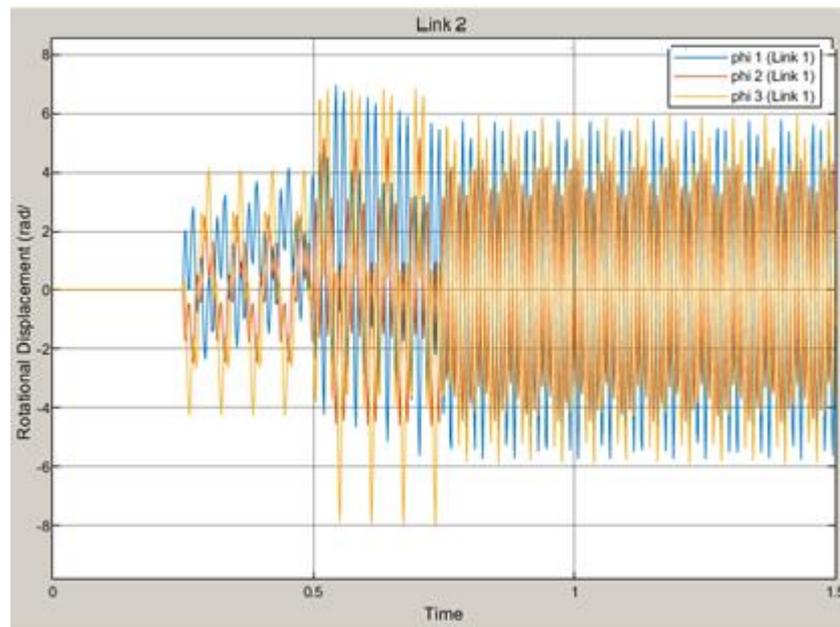


Figure 11: Rotational displacement of link 1 without a controller

Figure 12 represents the rotational angle of link 2. It can be observed from the graph that the value of the theta starts to increase abruptly and then reaches to a certain value and then remains unstable.

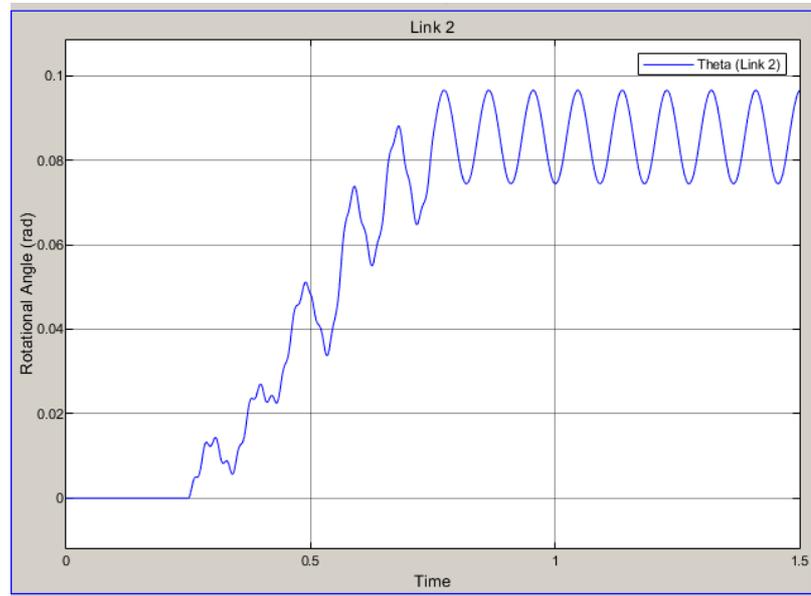


Figure 12: Rotational angle ( $\theta$ ) link 2 without a controller

Deflection of Nodes in Link 2 is represented in Figure 13. It can be observed from the graph that the values of deflection are unstable and continuously oscillating. The same case applied for the rotational displacement in link 2 which is presented by Figure 14.

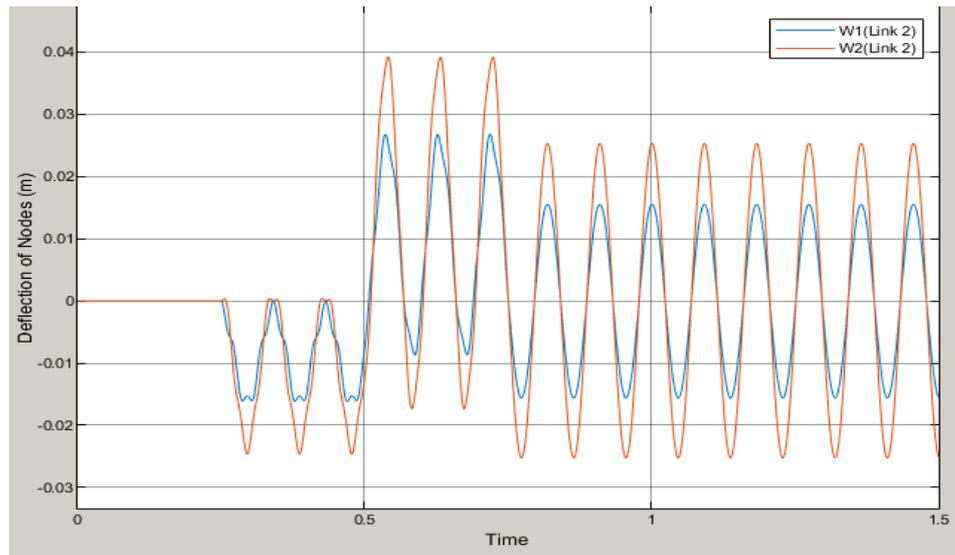


Figure 13: Deflection of nodes of link 2 without a controller

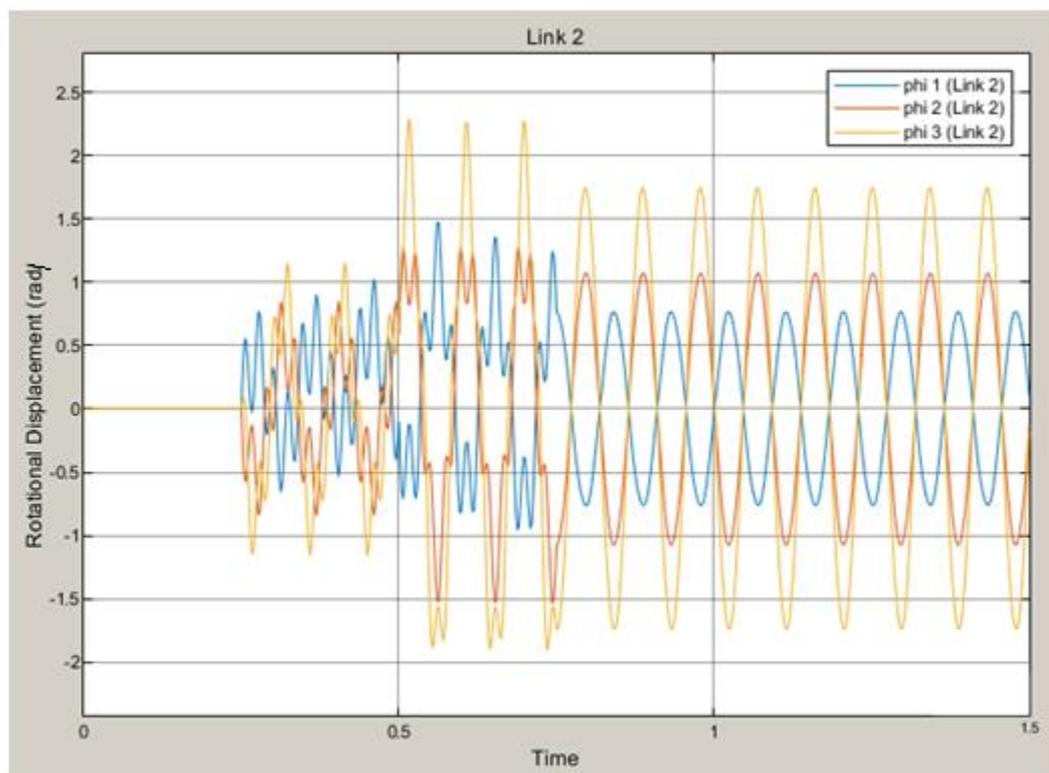


Figure 14: Rotational displacement of link 2 without a controller

In Figures 9-14, it can be seen the waveform simulation of various signals that are used in the circuit diagram shown in Figure 8. The waveform of theta signal first rises abruptly to a maximum value than it remains there, and oscillation is present in this signal. The waveform for rotational angle shows that the signal first goes down below zero and then increases and then the oscillations become stable at a certain value and remains constant. Deflection of nodes is oscillating in the start at low amplitude, but then increases gradually which shows that there is chattering in the behavior of the manipulator. So rotational angle does show oscillations arounds its steady state equilibrium points which depicts the performance of the manipulator in Figure 9.

Based on the equations illustrated earlier, the total displacement of link 1, is shown in Figure 11, while Figures 12, 13 and 14 represent the parameters for link 2.

It is clearly shown in Figure 14 that the behavior of the manipulator without the use of a controller system was completely sporadic and unreliable, however, it is shown that the transient response settles after some time, and it becomes harmonic, which emphasizes the importance of using and implementing a controlling system to control the movement of the manipulator.

In the coming sections of this research, the implementation of a controlling system will be discussed. Three types of controllers will be used to control the manipulator model; these controllers are PID, LQR and LQG controllers. Afterwards, the behavior of each system will be tested and observed, and a comparison between these controllers will be conducted to decide the best behavior achieved by each controller.

#### 4.2.2 Simulation Results Using PID Controller

The first type of controller to be implemented into the model is the PID controller, which stands for Proportional Integral Differential (PID) controller. This type of controller is widely used in many applications including the industrial control systems that require continuous modulation data control. It can be stated that, PID controller is a conventional type of controller which has a wide variety of uses and applications in the industrial field.

The proportional part of this controller gives an indication of how the error signal is propagated, the differential part indicates the rate of change in the error signal, which will also predict the value of the error signal in the future. Too much increase in proportional gain can also lead to deteriorated stability. Differential gain helps to reduce the overshoot and settling time of the states to desired reference. However, the integral part indicates the previous history of the error signal till the current moment. Integral gain helps reduce the rise time of states to the desired reference and helps eliminate the steady state error.

PID controllers are used to control the behavior of the manipulator model described earlier in Figure 6. Angular displacement of both joints has been tracked to a desired reference. The implementation of this controller into the manipulator state space model has been illustrated in Figure 15 .

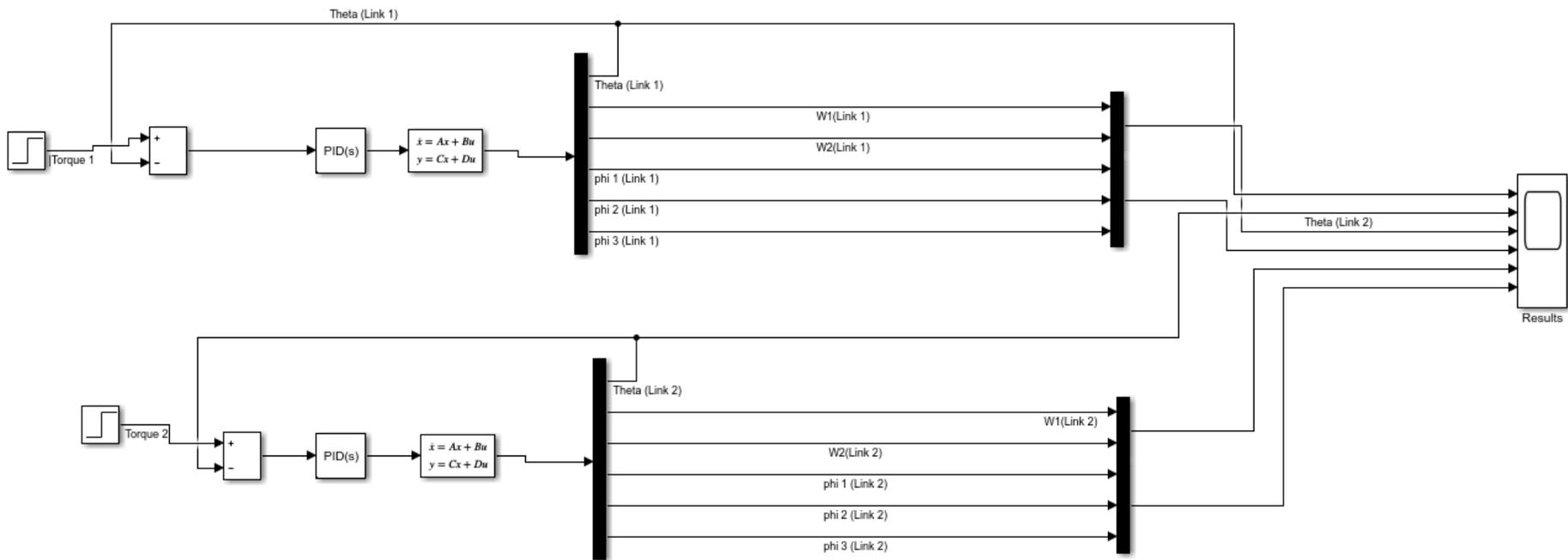


Figure 15: Model simulation using PID controller to track angular displacement

By tuning the gain parameters of the PID controller, states of the controlled system will be tracking the desired references. The tuning of PID gains has been done using Nicholas Ziegler's second method. Careful selection and turning of PID parameters/gains was done precisely to achieve greater accuracy and better behavior of the system. The gain values were entered into the PID block in a Simulink diagram. The values of P, I, and D gains used to model this diagram are entered in Table 2, while the tuning block of the PID controller is illustrated in Figure 16.

Table 2: PID tuning results

<b>PID Tuning Results</b>	
P	0.9136
I	0.4278
D	0.3757
N	88

Figure 16 represents the block parameters of the PID controller used to control the dynamic model of the two joint rigid flexible manipulator.

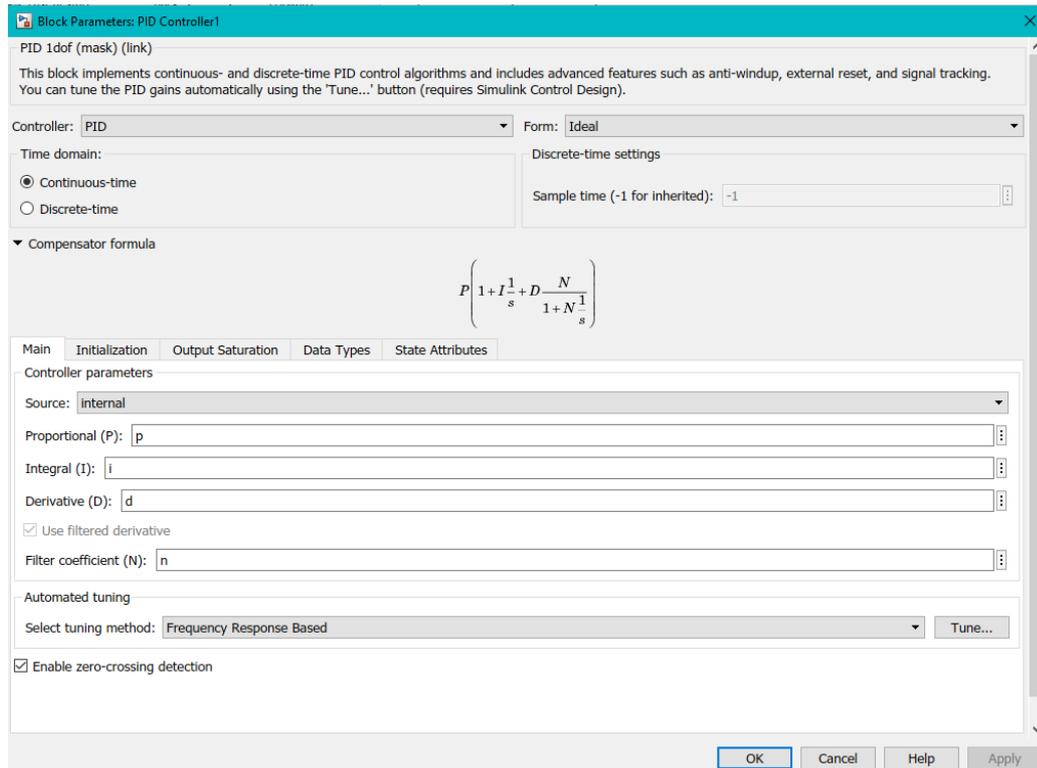
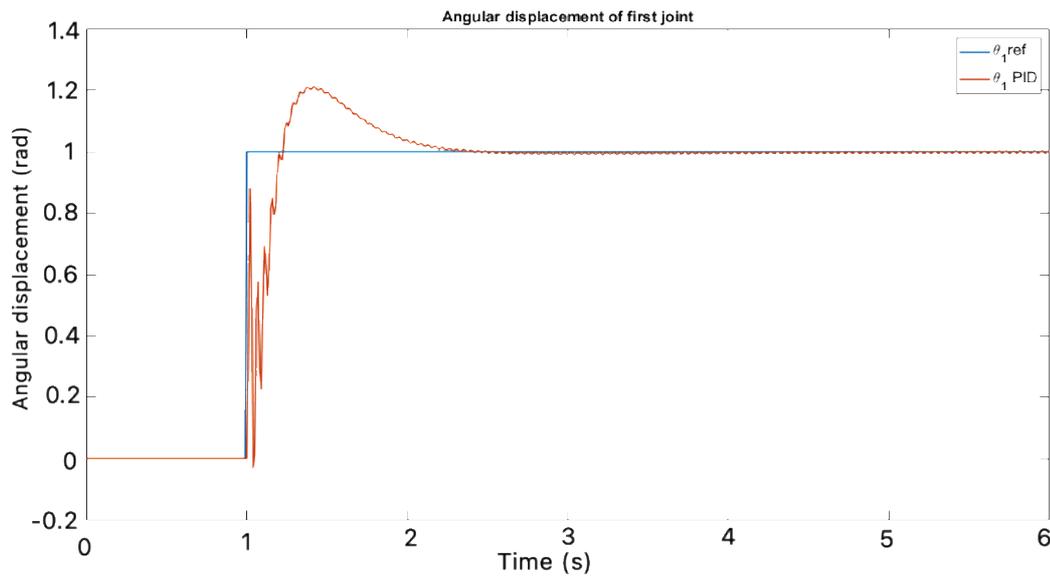
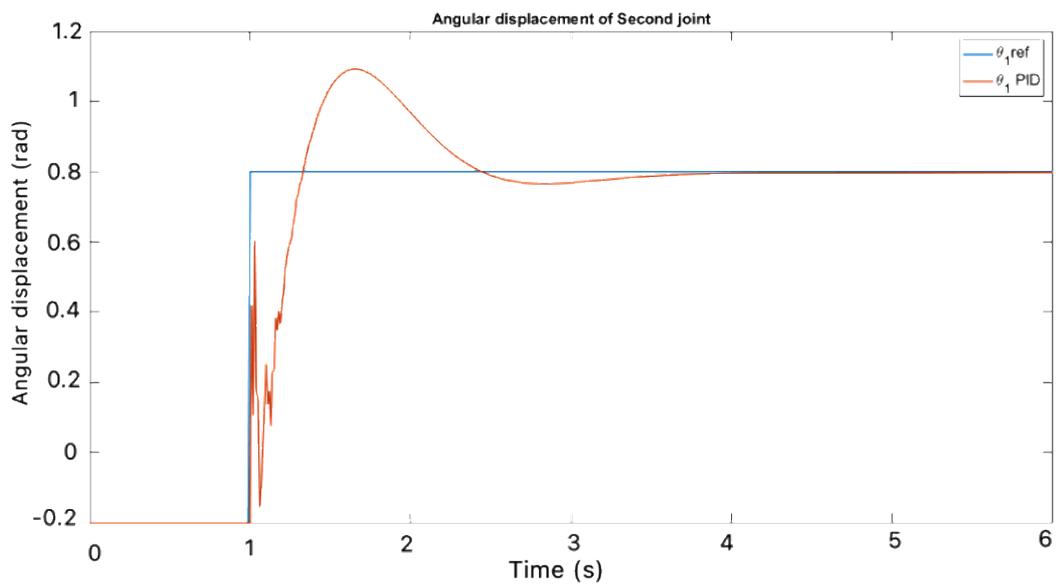


Figure 16: PID tuning block from MATLAB control system library

To simulate the behaviour of the two joint manipulator initially the state space matrix was loaded in the MATLAB workspace. Behavior of the system states after implementing PID controller into the model are presented in Figures 17 - 20. A dynamic desired reference has been considered to better estimate the performance of the implemented manipulator model. Desired reference for the angular displacement of both actuators has been taken 0 rad for a time span of 1 second and then changed to 1 rad till the end of simulation. Dynamical response of the angular position of joint 1 and joint 2 has been presented in the Figure 17 and Figure 18.

Figure 17: Angular displacement  $\theta_2$  of link 1 with PID controllerFigure 18: Angular displacement  $\theta_2$  of link 2 with PID controller

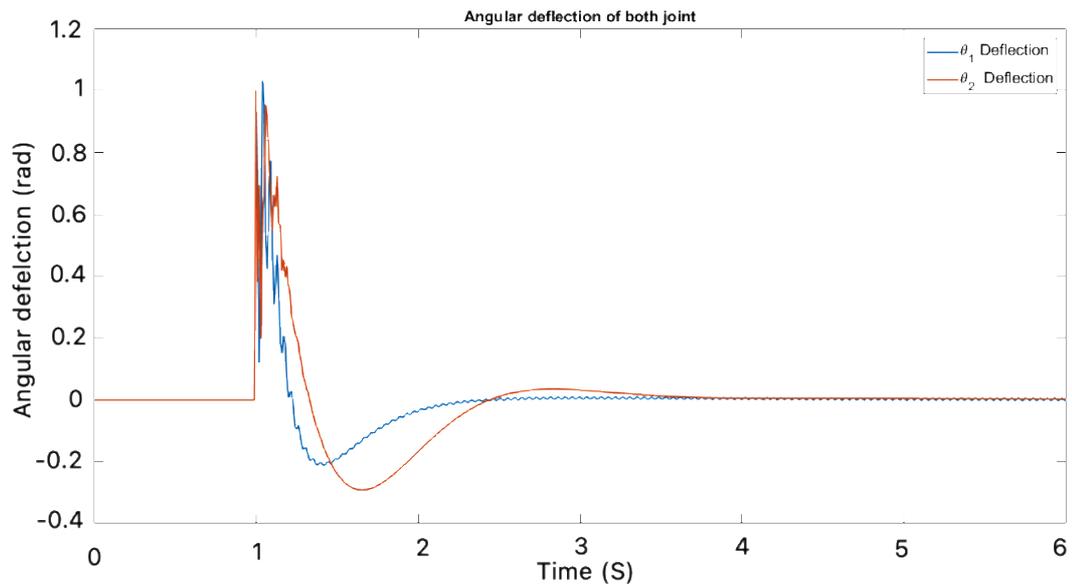


Figure 19: Deflection of nodes of link 1 and link 2 with PID controller

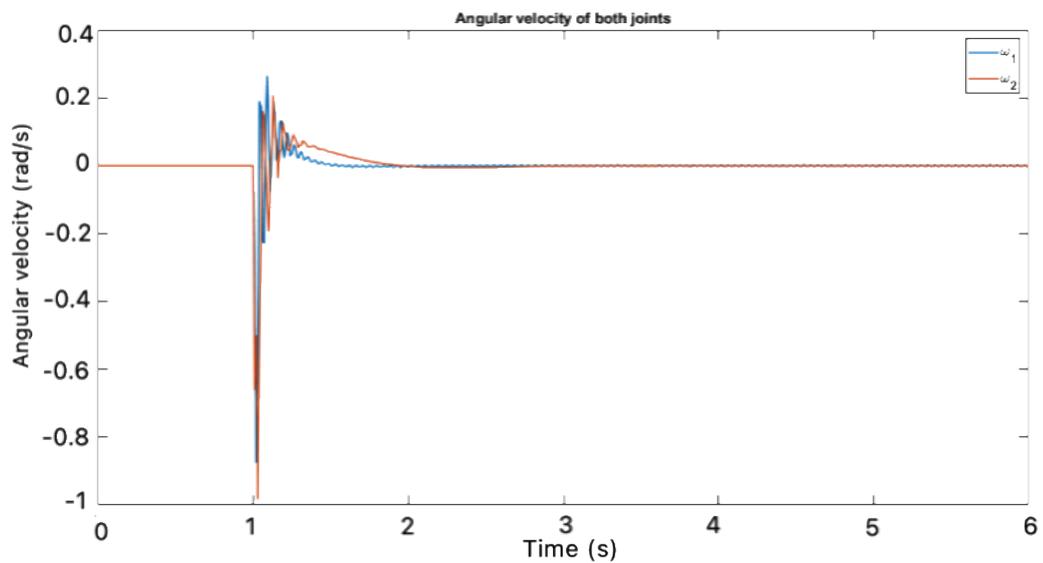


Figure 20: Angular velocity of link 1 and link 2 with PID controller

By implementing a PID feedback controller to track the desired reference for both joints angular displacement, the performance of the system is better than a bang-bang actuation signal to achieve a desired point on planner ground. Based on its error

minimization architecture PID controller is gradually decreasing the steady state error. Initially due to large error value at 1 sec integral gain tries to reduce the error by a large actuation signal at the joint and shows small overshoots at that phase of the dynamical response. Gradually the system states reach within its settling state when the steady state error is less than 2%. After 8 sec the link 1 angular displacement has been decreased to zero. Similar behavior has been observed in link 2 angular displacement as well. The angular deflection of joint 1 and joint 2 has been presented in the Figure 19.

The angular deflection of both joints from the desired reference can be observed in Figure 19. During the first second of simulation, there is no motion as the desired reference of the angular displacement, the same as the initial conditions. No actuation signal was generated to keep the joint at the desired position. Once a change in desired reference has been observed, the joint tries to track the reference position by the actuation signal generated at the PID output which has been fed to the manipulator state space model. Angular deflection of both joints from their desired reference with some overshoot and chattering decreased down to zero.

Rate of change in angular displacement of both joints can be observed from the angular velocity of both joints, which presents how fast the joints are moving with respect to time. Angular acceleration presents the variation in angular velocity with time. Angular velocity and angular acceleration help us analyze the smoothness of the response in case of planner robot. The angular velocity and angular acceleration of both joints have been presented in Figure 20 and Figure 21 respectively.

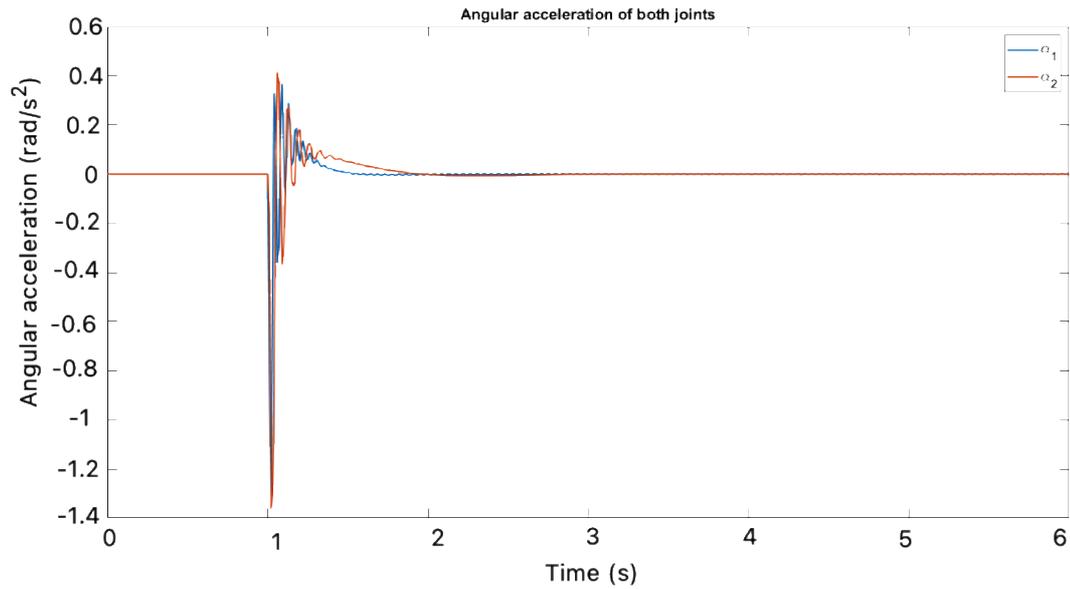


Figure 21: Angular acceleration of link 1 and link 2 with PID controller

It can be observed from Figures 17, 18, 19, 20 and 21, that the response of the system is fast, as velocity and accelerations are changing very rapidly during the transition stage of the desired reference value. Due to large actuation signal, higher velocities and accelerations have been observed in the transition phase causing overshoot from the desired reference. The actuation signal generated by the PID controller has been presented in the Figure 22.

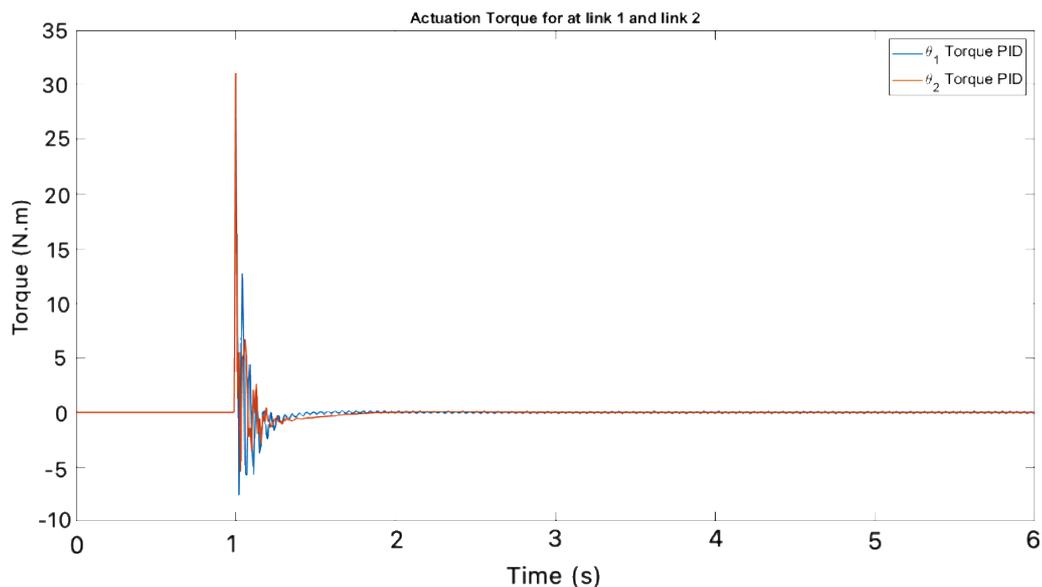


Figure 22: Actuation torque for both joints of manipulator with PID controller

It has been observed in Figures 17,18,19,20, and 21 where all states of the system were showing overshoot at the time of change in the desired reference. A major reason for the overshoots can be observed from the large spike of actuation signal generated by the PID controller. This type of controller with overshoots cannot be implemented with sensitive instruments where accuracy is required, but with less sensitive system this will work fine as PID has a smaller amount of computational effort. A small amount of vibration has been observed in each figure, however the stability of the model has been significantly increased compared to the model response without implementation of a feedback controller.

All of the Figures describing the movement of link 2 clearly indicates that there is a noticeable suppression in vibration of the link, and a better accuracy in the link motion; the duration needed for the stability of the link was decreased by implementing the PID controller into the model.

### 4.2.3 Simulation Results Using LQR Controller

In this section the implementation of the LQR controller into the manipulator model has been presented. LQR Stands for Linear Quadratic Regulator. The actuation signal generated by the LQR decreases with a decrease in error signal. A group of differential equations are generated to describe the controller dynamics, however, the mathematics behind these equations was not discussed as the LQR cost function was intuited and tweaked by setting the penalty factors for the performance and effort of the actuators using the MATLAB command  $[K,S,e] = \text{lqr}(\text{SYS},Q,R,N)$  and performance and the formula used for the LQR controller and the results and figures that describe the overall behavior of the system have been presented and discussed afterwards.

For a continuous time system, the state-feedback law  $u = -Kx$  minimizes the quadratic cost function:

$$J(u) = \int_0^{\infty} (x^T Q_x + u^T R u + 2x^T N u) dt \quad (44)$$

Subject to system dynamics:

$$\dot{x} = Ax + Bu \quad (45)$$

In addition to the state-feedback gain  $K$ ,  $\text{lqr}$  returns the solution "S" of the associated Riccati Equation:

$$A^T S + SA - (SB + N)R^{-1}(B^T S + N^T) + Q = 0 \quad (46)$$

The closed-loop eigenvalues  $e = \text{eig}(A-B*K)$ .  $K$  is derived from  $S$  using the following Equation:

$$K = R^{-1}(B^T S + N^T) \quad (47)$$

However, the MATLAB command  $[K,S,e] = \text{lqr}(\text{SYS},Q,R,N)$  was used to represent the function of the LQR controller to spare the time of going through the mathematical derivation of the LQR Equations 44, 45, 46, and 47.

It is worth mentioning that no actuation noise has been considered in the simulated system. The State weight matrix and input weight matrix has been presented below:

$$Q = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{6 \times 6} \quad (48)$$

$$R = 1$$

The dynamical response of the manipulator was improved by implementing the LQR controller. It is expected that the implementation of this controller into the system will increase the efficiency and stability of the system. MATLAB code for LQR parameter calculation is given in Appendix D. The Simulink state space model of the manipulator with feedback LQR Control scheme has been presented in the Figure 23.

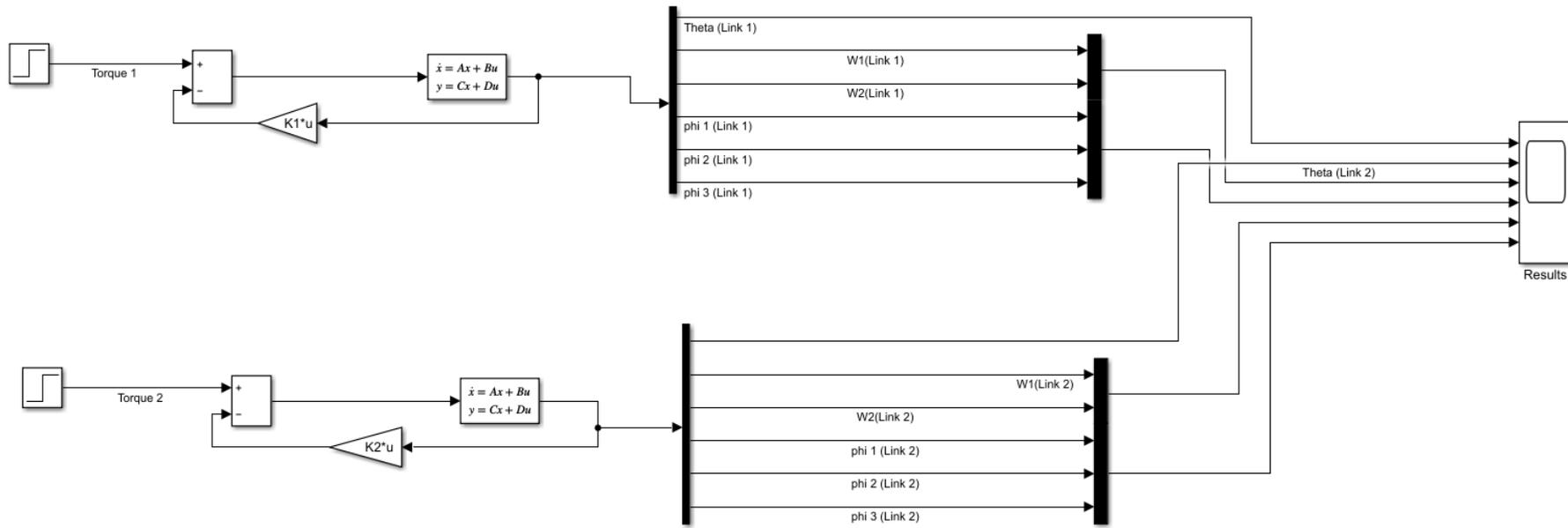


Figure 23: Simulation of LQR control scheme

A state feedback LQR controller has been implemented on the two-link state space dynamical model of the manipulator. The system has been subjected to a dynamical desired reference like the PID feedback controller scenario to observe the performance of the LQR state feedback controller. The time dynamic response of the manipulator against dynamic reference position for both the joints angular displacement has been monitored.

Angular displacement in both joints has been presented against desired reference has been presented in Figure 24 and Figure 25.

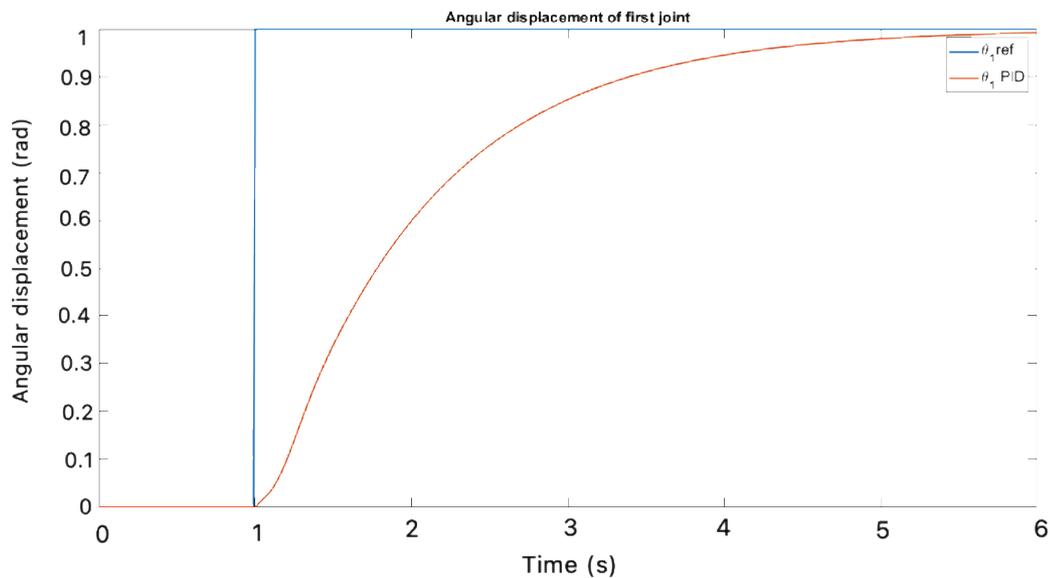


Figure 24: Angular displacement of link 1 with LQR controller

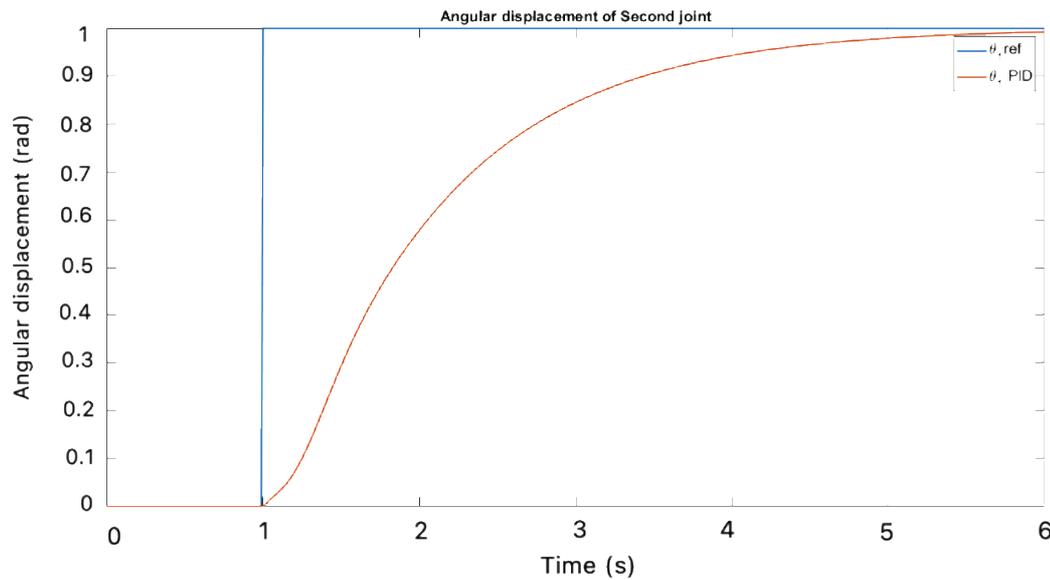


Figure 25: Angular displacement of link 2 with LQR controller

From Figure 23, it can be observed that the LQR state feedback controller is tracking the desired reference of the dynamical desired reference with zero overshoots. This behavior of state feedback controller is due to its regulator architecture. Whereas in Figures 24 and 25, this can be observed that the approach time for the angular displacement of both joints of manipulator has been increased compared to the PID controller. Rise time of the system dynamical response has also been increased. With slow dynamical response of the system stability of the manipulator has been increased. Angular deflection of both joints' angular displacement from the desired reference has been presented in Figure 26.

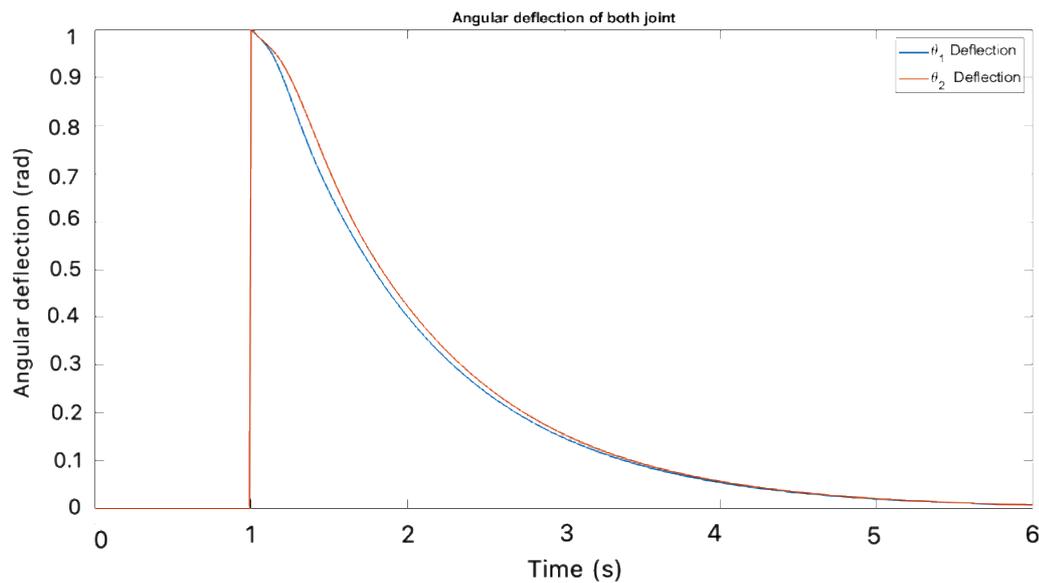


Figure 26: Deflection in both joints from the desired reference with LQR controller

Slow dynamical response has been observed in the deflection graph of the states as well. A sudden rise in deflection waveform can be observed in both joints at the point of 1 second when desired reference for the angular displacement has been shifted from 0 rad to 1 rad. LQR slow dynamical response gradually decreased the error between the desired reference and the system states. As the error between both states gradually decreases resulting into a decrease in actuation signal which leads to a slow dynamical response. The actuation signal for both joints has been presented in Figure 27.

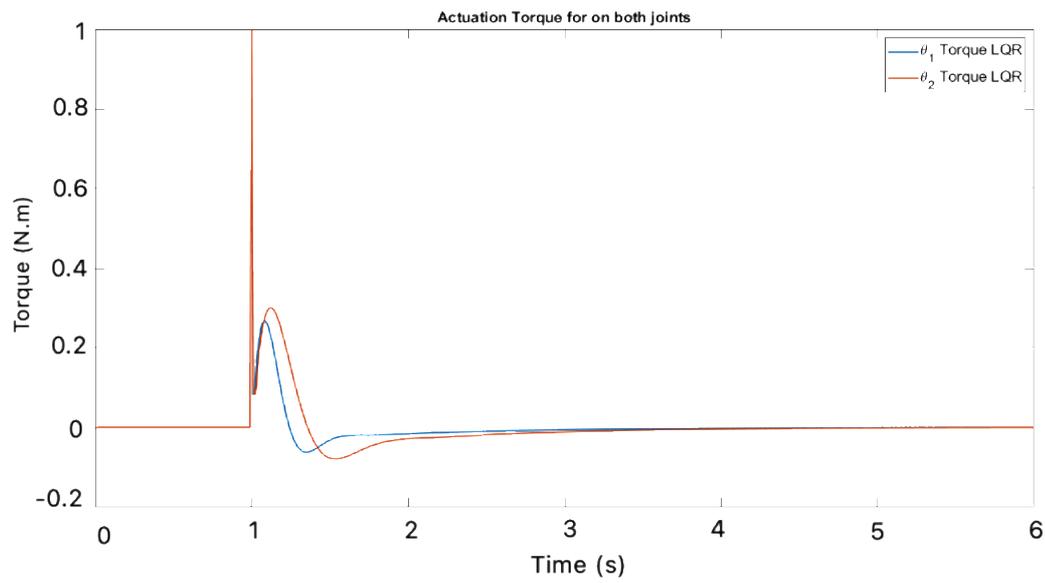


Figure 27: Actuation torque for both joints with LQR controller

It can be observed that the slow dynamics of the LQR came as a result of the small actuation value, which made the dynamical response of the system free of overshoots or steady state error. Angular velocity and angular acceleration of the two joint state space model has been presented in the Figure 28 and Figure 29, respectively.

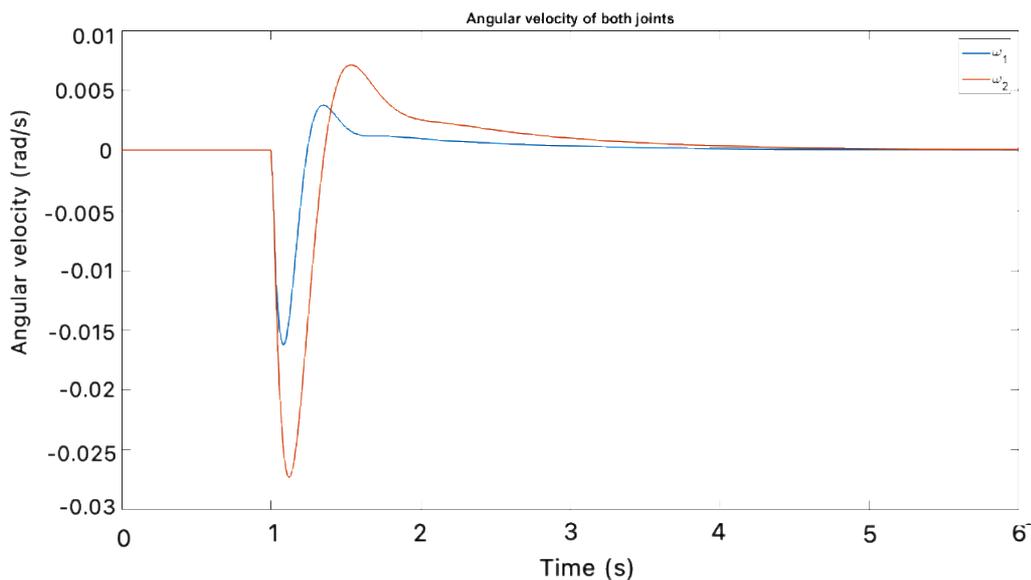


Figure 28: Angular velocity of both joints with LQR controller

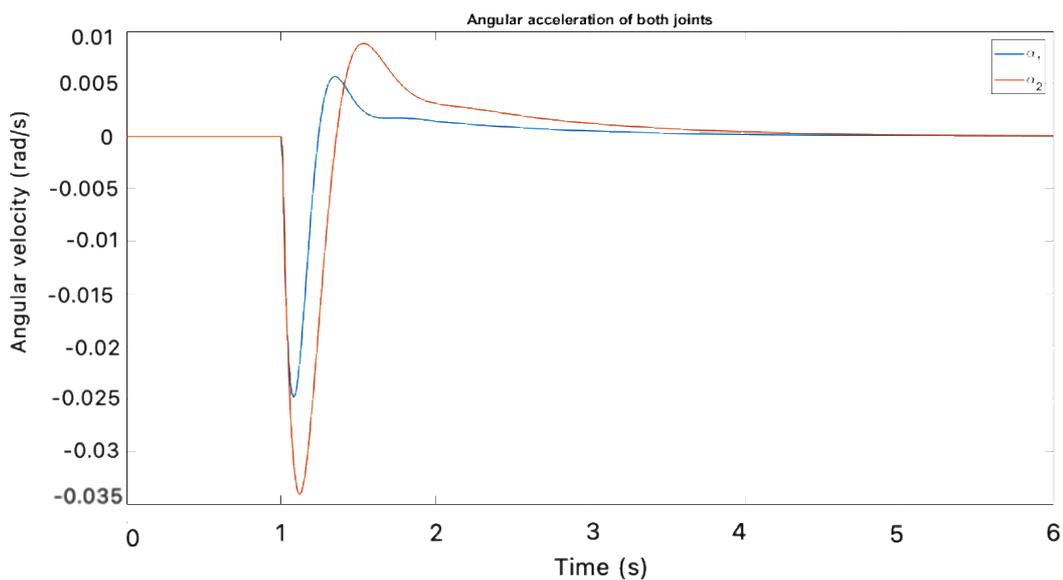


Figure 29: Angular acceleration of both joints with LQR controller

It can be observed from Figures 24-29 that, when implementing a state feedback LQR the angular velocity and angular acceleration of the manipulator has been decreased significantly due to small actuation values generated by the LQR state

feedback controller. Due to the slow dynamics system rise, the time has been increased but system steady state error and settling time has been decreased. Overshoot and chattering have been removed from the response of targeted system states. LQR feedback controller is the best option compared to PID when high accuracy of the system response is required. Whereas if a fast response is required and small overshoots will not affect the desired output then the PID controller is the best choice.

#### 4.2.4 Simulation Results Using LQG Controller

The third type of controller that is being implemented into the model is the LQG controller. LQG stands for the linear–quadratic–Gaussian controller. It studies the linear systems which are normally driven by Gaussian White Noise. Its primary purpose is to minimize the expected value of the quadratic cost in the systems; it can be considered as a combination of Linear–Quadratic State Estimator (LQE) and Linear–Quadratic Regulator (LQR).

The LQG is a modern state space modeling technique for optimal dynamic regulators or servo motor with set point tracker. To design the LQG filter we need to follow the following steps:

1. Construct the LQ-optimal gain.
2. Construct a Kalman filter.
3. Form the LQG design by connecting the LQ-optimal filter and Kalman filter.

To construct the Kalman estimator type the following command.

```
[kest, L, P] = kalman(sys,Qn,Rn,Nn);
```

The software generated this state estimate using the Kalman filter.

$$\frac{d}{dt} \hat{x} = A\hat{x} + Bu + L(y - c\hat{x} - Du)$$

For a continuous time, system, the state-feedback law  $u = -Kx$  minimizes the quadratic cost function:

$$J(u) = \int_0^{\infty} (x^T Q_x + u^T R u + 2x^T N u) dt$$

To construct the optimal gain the following MATLAB command is used:

$$K = \text{lqr}(A, B, Q, R, N)$$

To form the LQG filter, connect the LQ-optimal gain  $k$  and Kalman filter  $kest$  by using the following command in MATLAB:

$$\text{regulator} = \text{lqgreg}(kest, K);$$

Figure 30 represents the LQG controller scheme which has been developed using the MATLAB Simulink Library. The design parameters are implemented by using the derivations in chapter 4 and then calculated in the M file (see Appendix D). The implemented state space model of the state feedback LQG controller has been presented in the Figure 30 .

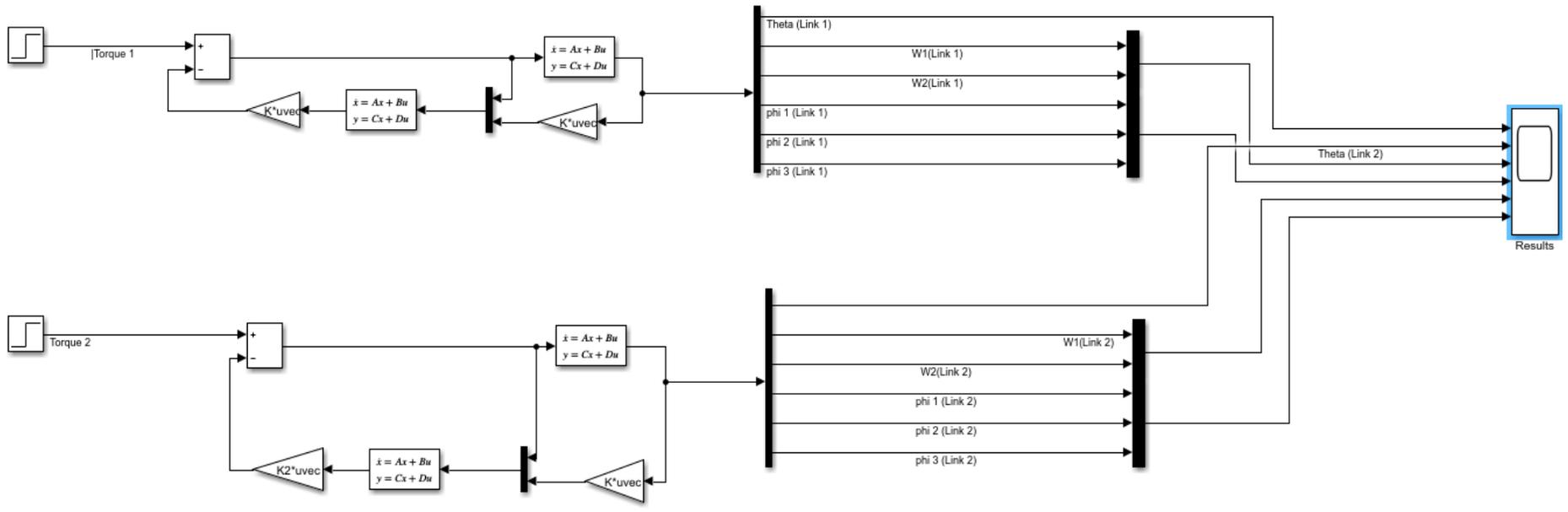


Figure 30: Simulation of LQG control scheme

It is worth mentioning that the noise intensity for the LQG controller was 0.001 rad, this value was used when designing the observer for all simulations done with the LQG controller.

The angular displacement of the link 1 and link 2 has been presented in the Figure 31 and Figure 32 respectively.

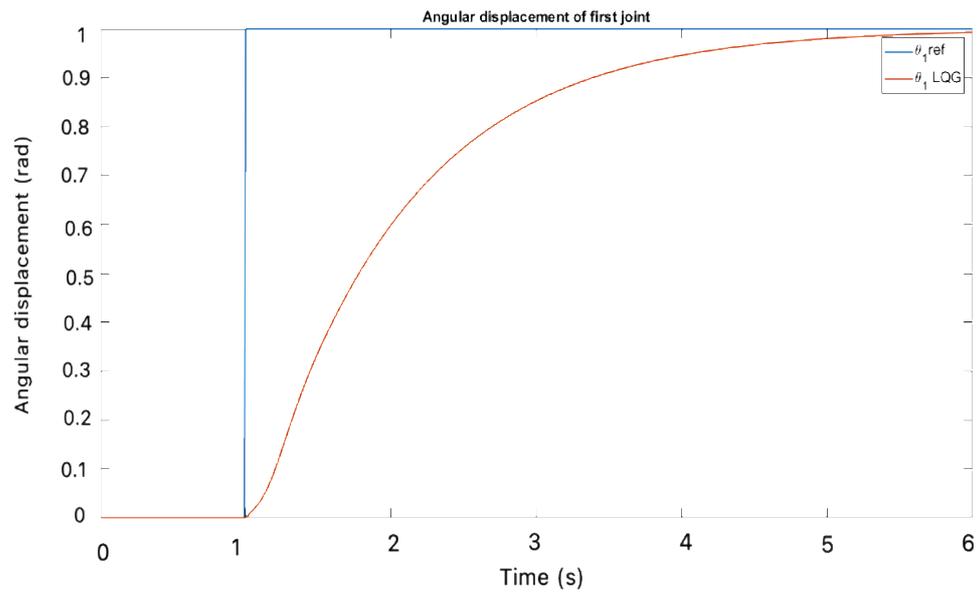


Figure 31: Angular displacement of link 1 with LQG controller

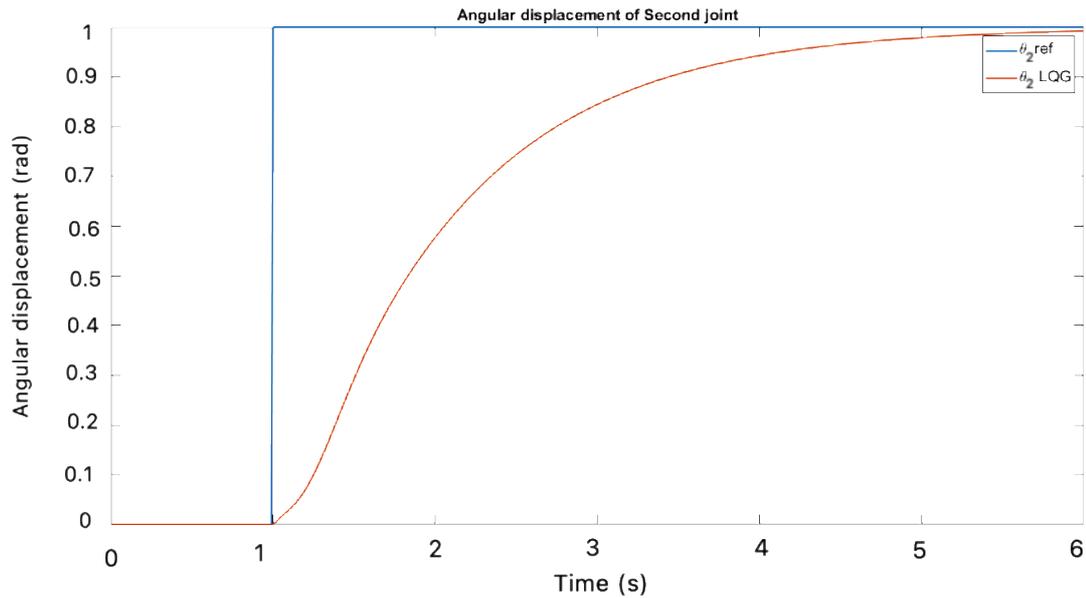


Figure 32: Angular displacement link 2 with LQG controller

In Figure 32, the response of the system dynamics is similar to the LQR state feedback controller. Using the estimator model, no significant difference in the response has been observed. The LQG feedback controller is regulating the control signal to provide a minimum overshoot. The time dynamical response of the deflection angle between the actual value of the angular displacement and desired reference has been presented in the Figure 33.

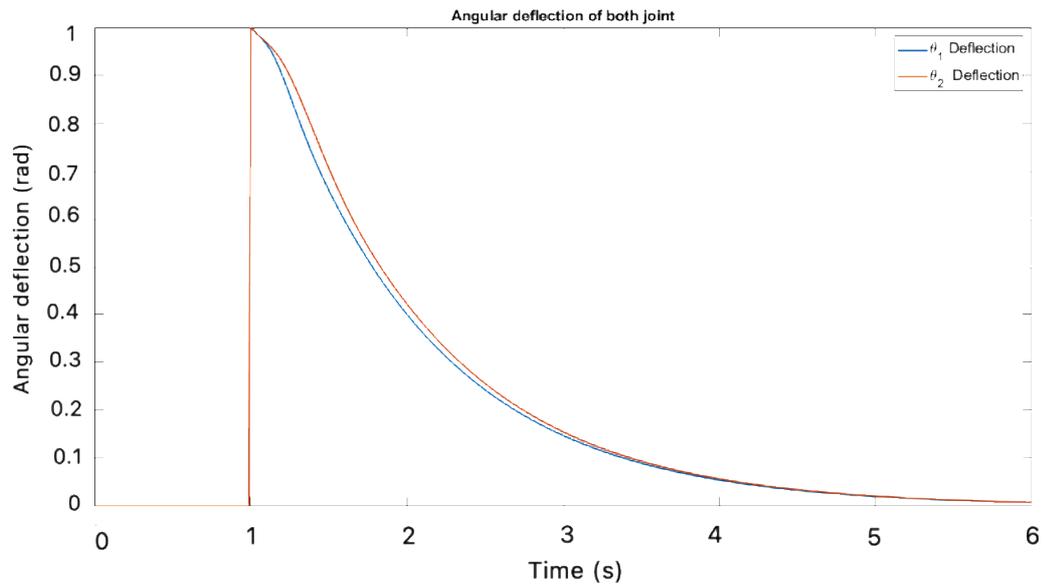


Figure 33: Angular deflection of both joints with LQG controller

From Figure 33, slow dynamical response of the system states has been observed as the error signal decreases with time. While this figure depicts an accuracy of the response with no overshoots from the desired reference. Angular velocity and angular acceleration of the two joint manipulator state space model has been presented in the Figure 34 and Figure 35.

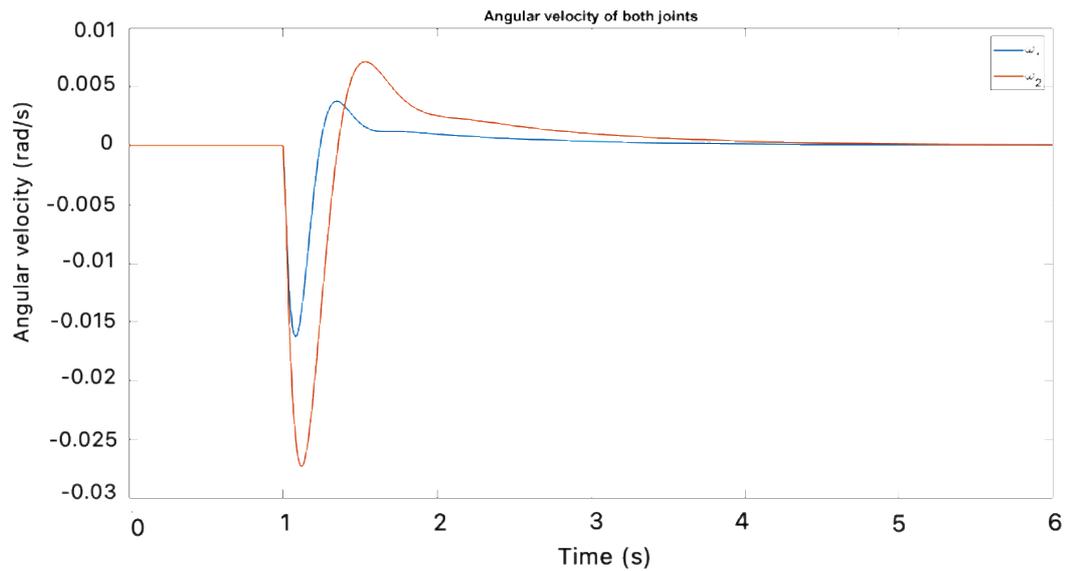


Figure 34: Angular velocity of both joints with LQG controller

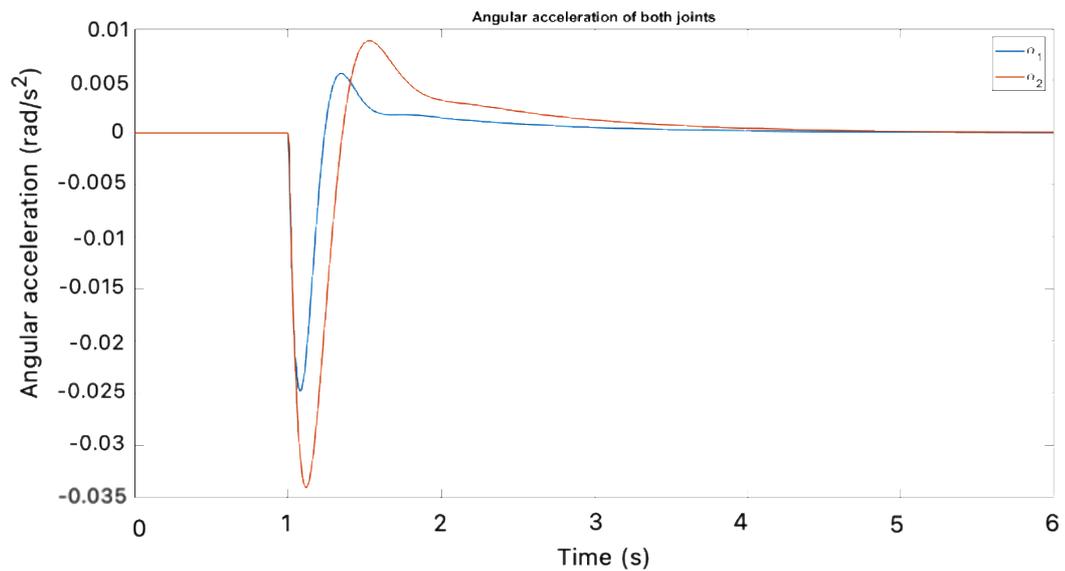


Figure 35: Angular acceleration of both joints with LQG controller

Like the LQR controller, a slow dynamical response has been observed in all the states of the system. Slow changes in acceleration and velocity leads to a stable dynamical system with a higher accuracy making it most attractive for high precision

applications. The actuation signal from the LQG controller has been presented in the Figure 36.

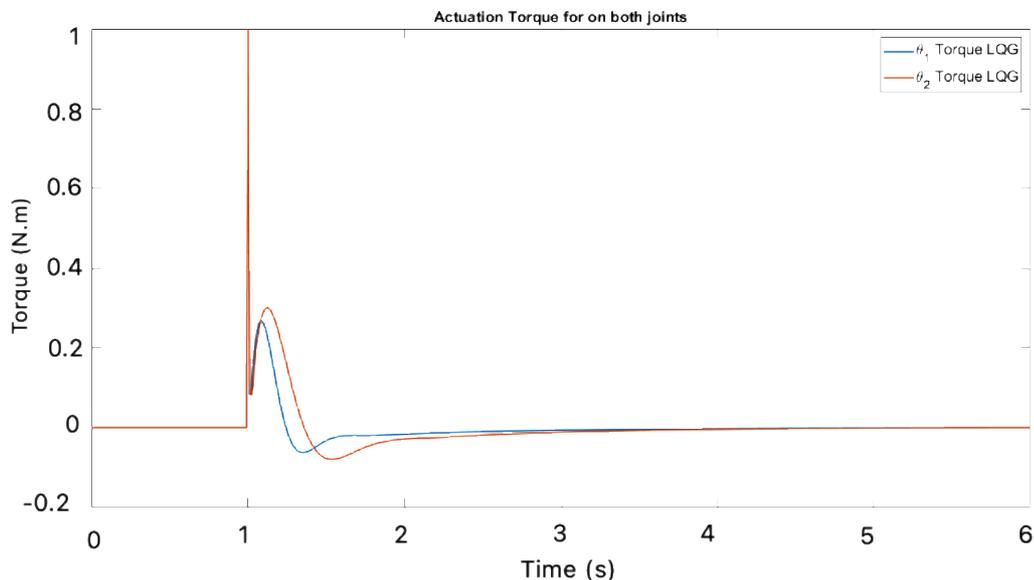


Figure 36: Actuation torque signal from the LQG controller with LQG controller

It can be observed from Figure 30 that an observer model has been implemented with an observer gain to include the response of estimator. The estimator output is fed to the LQR gain resulting in a LQG controller. System response of all the states has been presented in the Figures 31-36 in the section above. System dynamics has been subjected to a time varying desired reference for the angular displacement of the joint 1 and joint 2.

Time dynamical value of actuation signal generated by the LQG controller presented the cause of slow dynamics as due to a small value of actuation signal. However, this small and accurate signal leads to no overshoots and better stability than the PID controller.

### 4.3 Comparison Between Proposed PID Controller with the Literature

In controlling a mechanical manipulator, one should desire of to control the deflection of the arm precisely to reach and maintain with the desired deflection angle and at point of interest. Different control techniques we have studied in the literature to control the manipulator precisely so that in order for the work to be performed without any error. In the PID controller is designed to control the arm deflection of the two-link flexible manipulator using MATLAB/Simulink. The proposed PID controller is compared with Mohammed and Eltayeb (2018) and shows that the proposed model is much better in terms of to minimize the settling time and prevent oscillations. The PID controller parameters proposed in are given below:

$$k_p = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix} \quad k_d = \begin{bmatrix} 7 & 0 \\ 0 & 10 \end{bmatrix} \quad k_i = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

The response using the PID controller are depicted in Figure.36 which shows that the PID controller take at least 5 seconds to bring the arm of the manipulator in the desired location. On the hand the proposed PID in the research work take at least 2.5 seconds which is half of the time as compared with results of PID controller shown in Figure 37 [42].

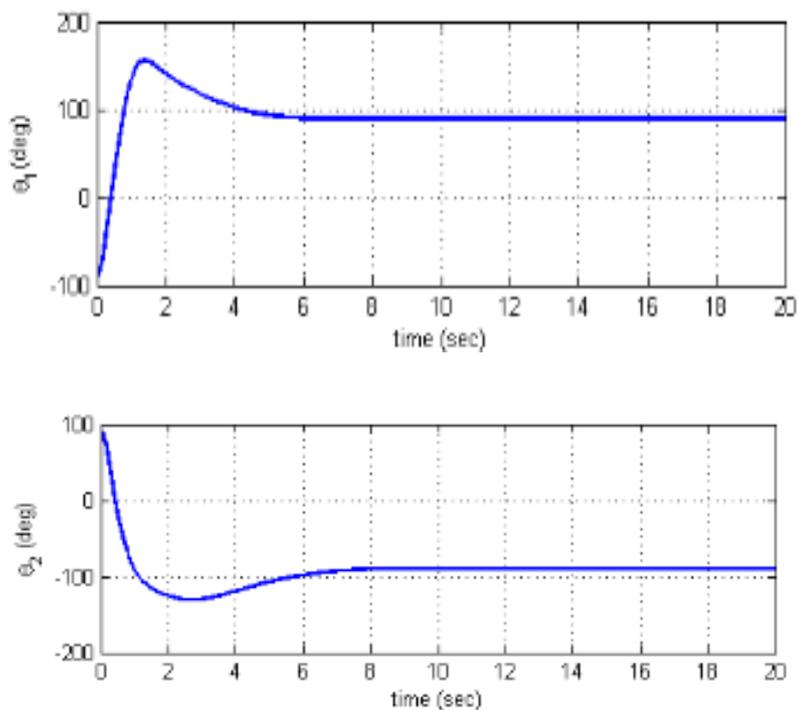


Figure 37: Trajectories of  $\theta_1$  and  $\theta_2$  of the literature

On the other hand, in comparison with the above deflecting angle the proposed deflecting angle of  $\theta_1$  and  $\theta_2$  are shown in Figures 38 and 39.

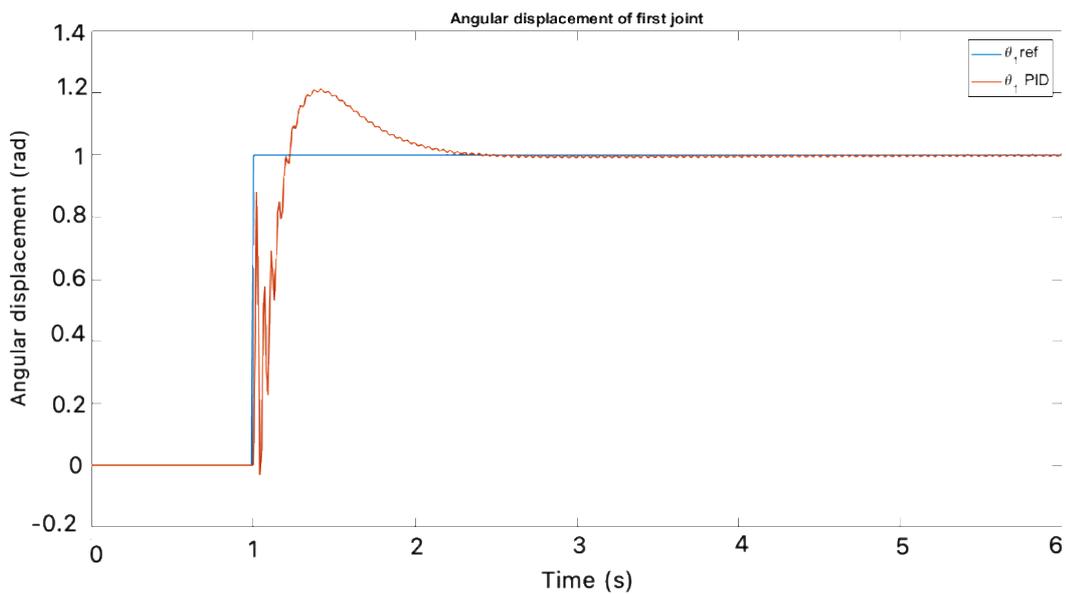


Figure 38: Angular displacement of first joint for the PID controller

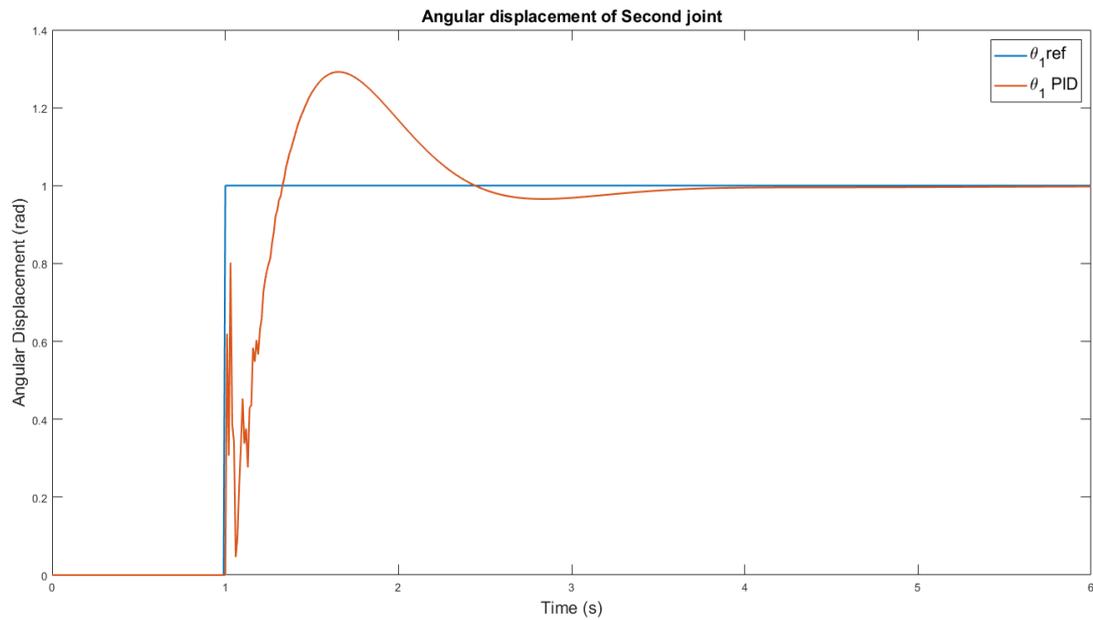


Figure 39: Angular displacement of second joint for the PID controller

From the above two figure it clearly shows that proposed PID controller out classed the PID controller reviewed in the literature in terms of settling time. The proposed PID Results are shown in Table 3.

Table 3: PID proposed results

<b>PID Tuning Results</b>	
P	0.9136
I	0.4278
D	0.3757
N	88

#### **4.4 Discussion**

After implementing the three controllers on the mathematical model of the two joint flexible manipulators, it has been clearly seen that the response of system states when implemented a modern controller, i.e. LQR and LQG have better performance than the conventional PID controller due to its relatively less sophisticated architecture. The system response against all the controllers has been plotted and presented in each section. The rise time of the PID controller was significantly lower than the LQR and LQG controllers but this fast dynamical behavior leads to overshoots in the tracked states from the desired reference. The system stability also deteriorated when using the PID controller. State response of tracked states has been presented in the figures below to present the comparative analysis of the system.

#### **4.5 Comparative Analysis of PID, LQR and LQG Controllers**

A comparative analysis has been done to analyze the performance of the mathematical model. The analysis has been conducted between PID, LQR and LQG controllers. Angular displacement and actuation torque have been analyzed to observe the response of the manipulator with different controllers. The angular displacement of link 1 and link 2 has been presented in the Figure 40 and Figure 41.

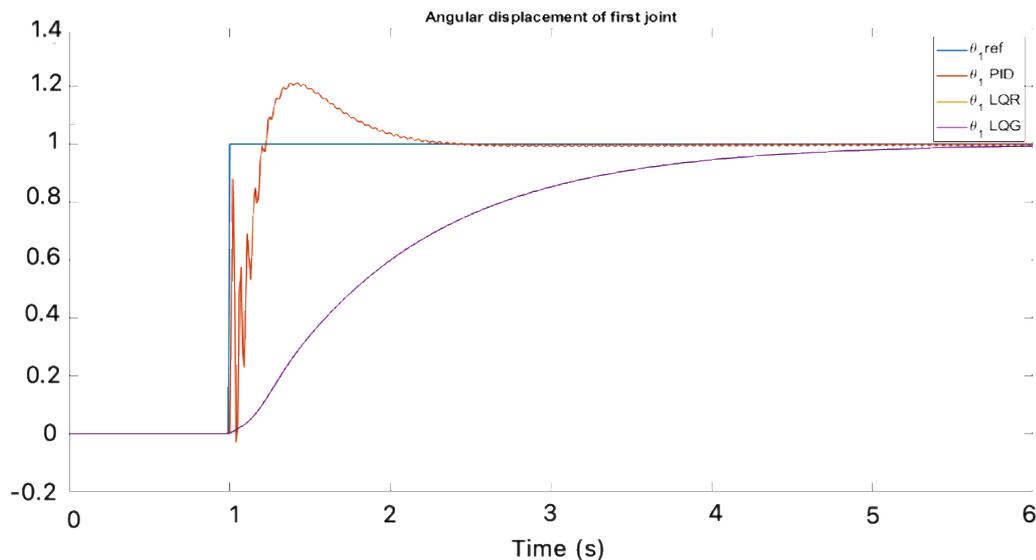


Figure 40: Comparison between angular displacement of link 1 for the three controllers

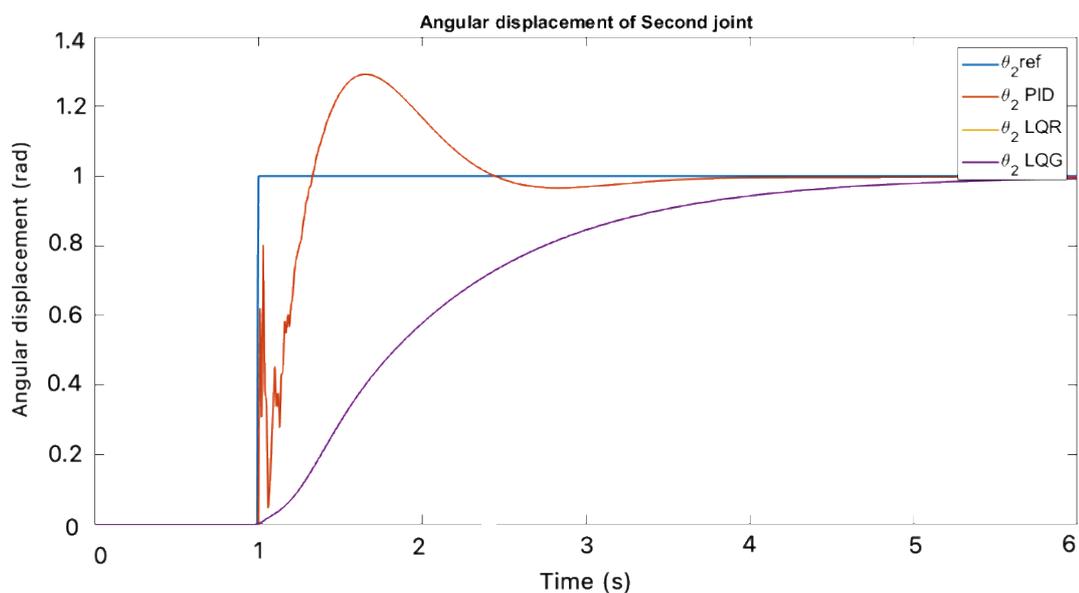


Figure 41: Comparison between angular displacement of link 2 for the three controllers

The amount of noise intensity in this controller was 0.001 rad, applied for all simulations. Higher amounts of noise intensity will lead to instability in the system response, that is why the same small amount of noise intensity was applied to all simulations.

Form Figures 10 and 41, it can be observed that the response of a system with the PID controller is faster than LQR and LQG but shows large overshoots which deteriorates the stability of the system. Whereas LQR and LQG presents a promising response with no overshoots and a minimum steady state error which leads to a more stable and accurate response. Actuation torques for link 1 and link 2 has been presented in the Figure 42 and Figure 43 respectively.

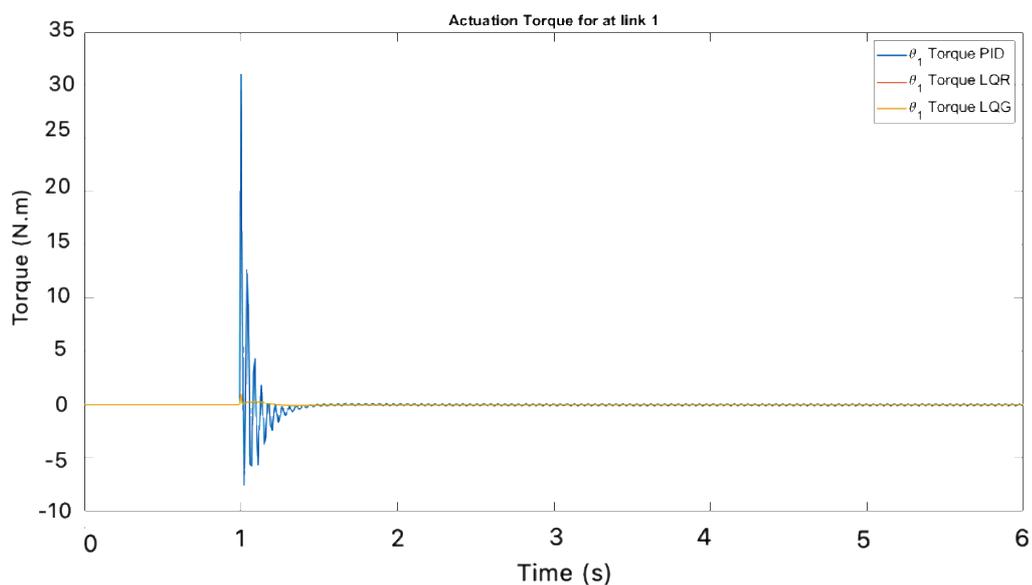


Figure 42: Comparison between actuation torque of link 1 for the three controllers

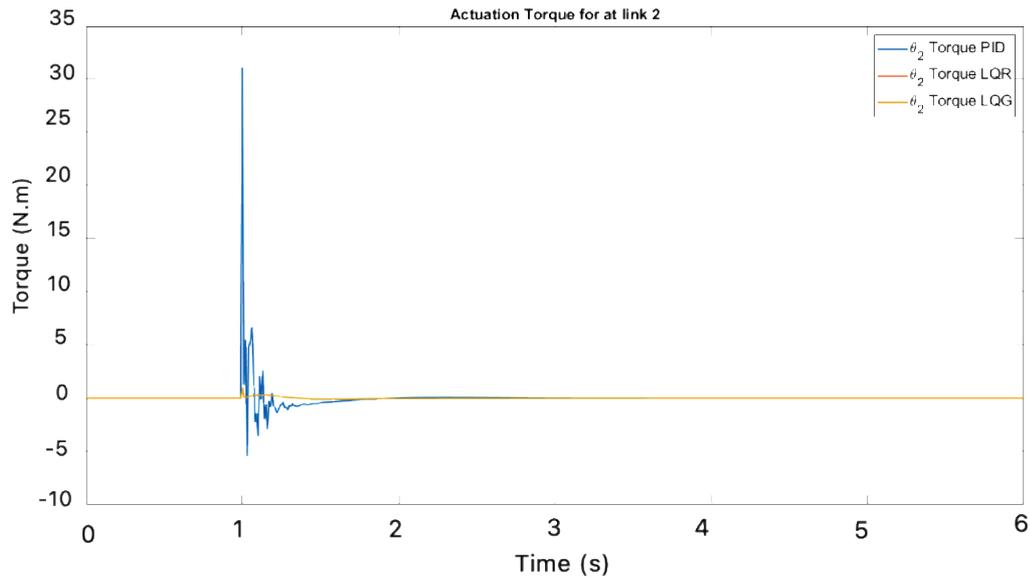


Figure 43: Comparison between actuation torque of link 2 for the three controllers

From the actuation signal, it can be observed that large overshoots have been present in the angular displacement of link 1 and link 2; in the case of the PID controller these are due to large spikes of actuation torques. These large spikes in actuation torques of both actuators in the case of the PID controller are harmful for electro-mechanical system. The large actuation signals can cause huge heat dissipation resulting in a break of fire and increased resistive losses. Best performing controllers in the proposed case are the LQR and LQG controllers which provides more stable and accurate response, making these two controllers ideal for the sophisticated applications.

Table 4 shows the comparative analysis of PID, LQR and LQG controller which shows the comparison of steady state error, percentage overshoot in the angular displacement accomplished during using respective three controllers.

Table 4: Comparative analysis of three controllers

<b>S.NO</b>	<b>Description</b>	<b>PID</b>	<b>LQR</b>	<b>LQG</b>
1.	Steady state error	0.005	0.007	0.008
2.	Percentage overshoot %	30	0	0

Table 4 also shows that the percentage overshoot does not occur in case of using LQR and LQG controller whereas 30% overshoot occur in PID controller. On the other hand, steady state error is minimum in case of PID controller with respect to the other controller. So, we have to trade-off among controller characteristics. The other performance characteristics of the controller is the Root Mean Square (RMS).

The graph of the RMS of the individual controller are shown in the following Figures 44-46.

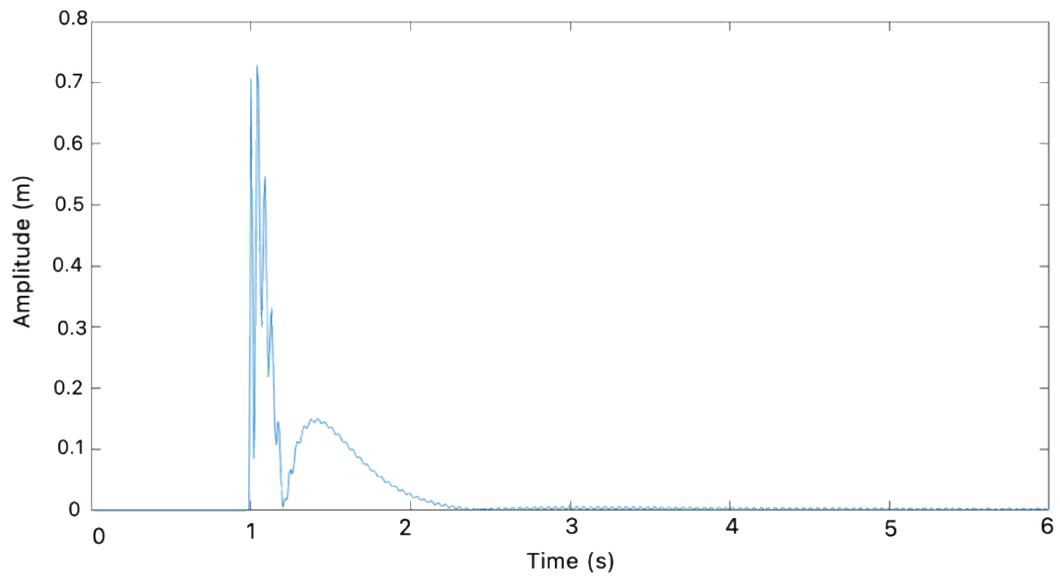


Figure 44: RMS graph of PID controller

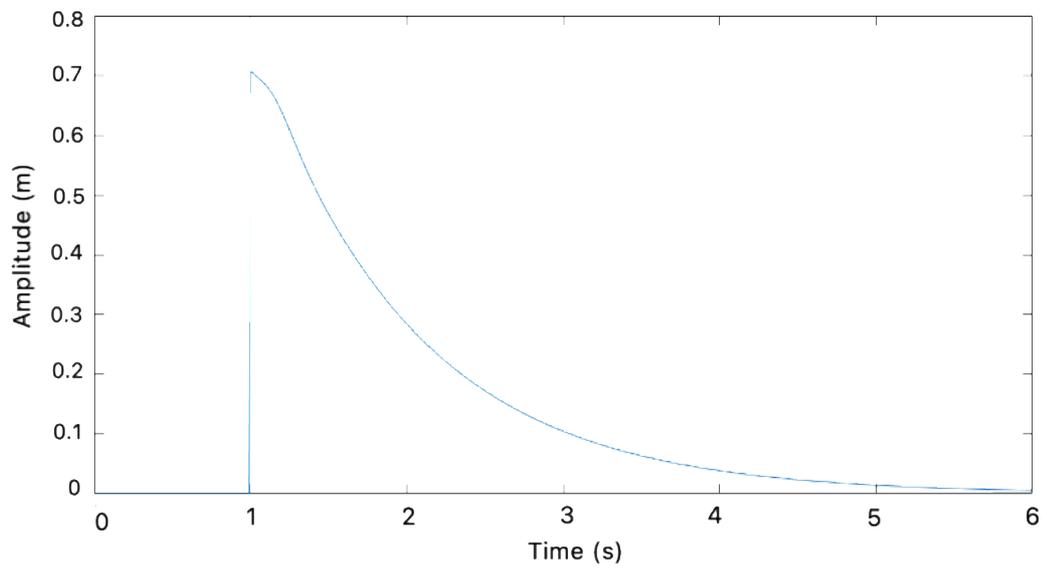


Figure 45: RMS graph of LQG controller

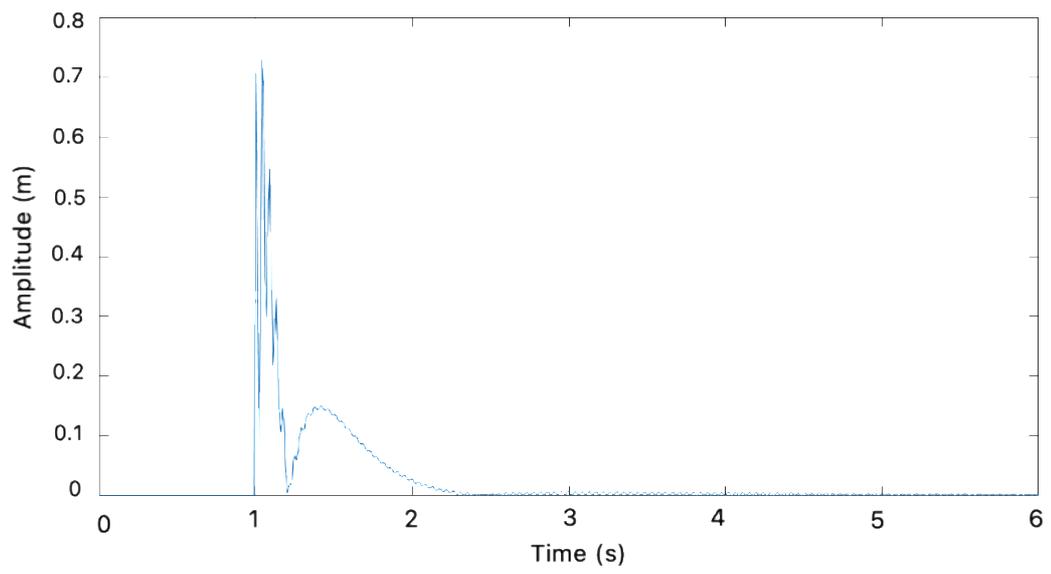


Figure 46: RMS graph of LQR controller

## Chapter 5: Modeling Based on Autodesk Inventor and MATLAB Simulink Software

A 3D model of the two-link rigid-flexible manipulator has been drawn using Autodesk Inventor software. The 3D model of Autodesk is then imported to MATLAB Simulink using SimMechanics library. The simulated model helps simulate a nonlinear dynamical model of the manipulator. The imported SimMechanics model in the Simulink has been subjected to similar dynamical desired reference for the angular displacement of the link 1 and link 2. PID, LQR and LQG controllers have been implemented to make a comparative study between SimMechanics and state space model of the manipulator. The comparative analysis has been done to validate the working and eligibility of Autodesk Inventor to model such systems.

Autodesk Inventor software has been used for 3D Mechanical design of the manipulator. The two-link rigid-flexible manipulator has been constructed as an assembly on Autodesk Inventor. The formulated mechanical model has been presented in the Figure 47.

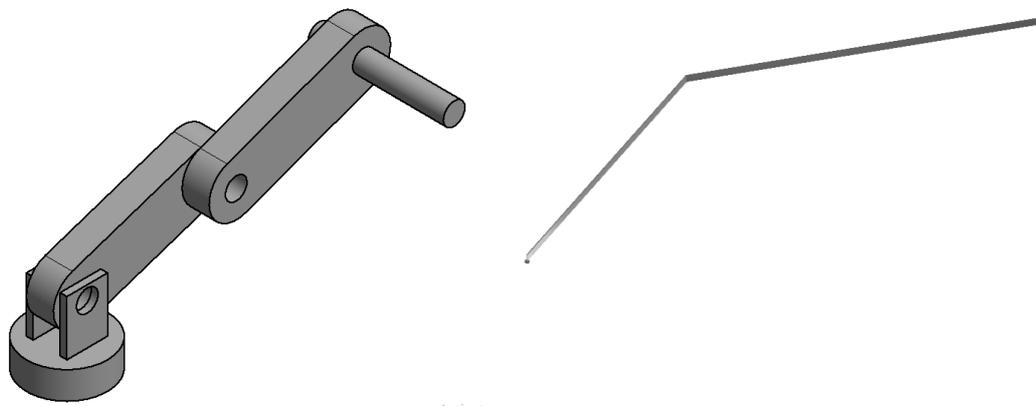


Figure 47: Two- link manipulator model in Autodesk Inventor

All the structural parameters of a two-link rigid flexible manipulator have been presented in the Table 5.

Table 5: Dimension of the simulated model

Parameters of link 1		Parameters of link 2	
Length, $L_1$	0.9 m	Length, $L_2$	1.1 m
Cross sectional area	$6.0833 \times 10^{-5}$ $\text{m}^2$	Cross sectional area	$6.0833 \times 10^{-5}$ $\text{m}^2$
Width	$18.9 \text{ mm}^2$	Width	$18.9 \text{ mm}^2$
Height	$3.22 \text{ mm}^2$	Height	$3.22 \text{ mm}^2$

The Autodesk model has been converted into a SimMechanics model by using a command presented below:

`smimport(Autodesk Model name)`

After importing the Autodesk model into MATLAB, by using this command two windows will be generated: A Simulink model using SimMechanics library of created mechanical models in Autodesk software and an m-file which has all the parameters of the designed Autodesk mechanical model. An M file generated using the above command has been present in the Appendix D.

The SimMechanics model generated from the above-mentioned command has been presented in the Figure 48.

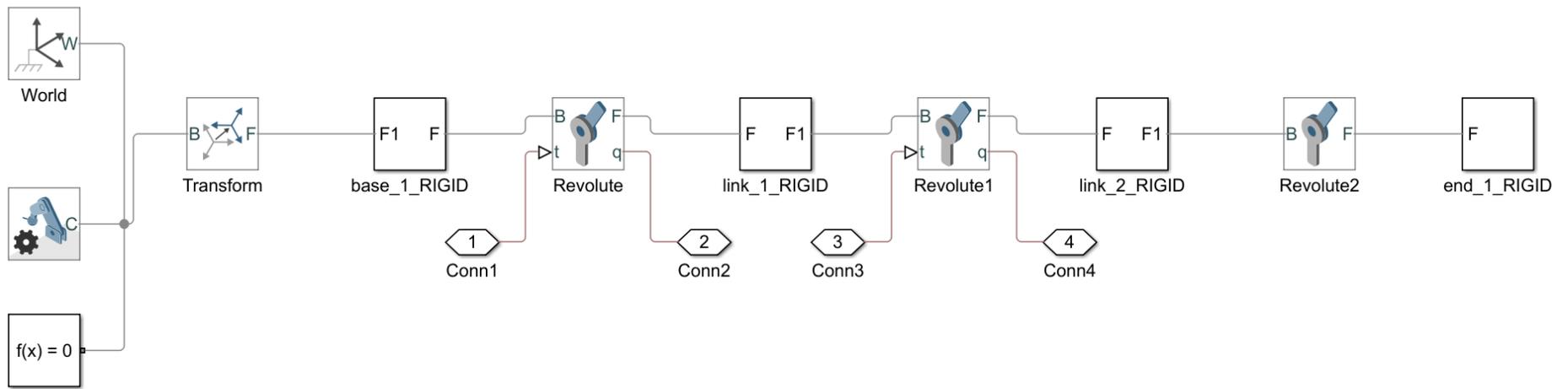


Figure 48: Model exported in MATLA

### 5.1 PID Controller Implemented on Simulated Model of Manipulator

A comparative study has been done on the behavior of two-link rigid flexible manipulator models generated by Autodesk software and then imported to Simulink and state space mathematical model using similar PID controller settings. System behavior has been simulated using similar PID controller gains, which were used during implementation of the mathematical model of manipulator. Proportional, integral, and derivative gains in the PID block Have been presented in the Table 6.

Table 6: PID controller settings

<b>PID Tuning Results</b>	
P	0.9136
I	0.4278
D	0.3757
N	88

Block setting of PID controller block has been presented in the Figure 49 below:

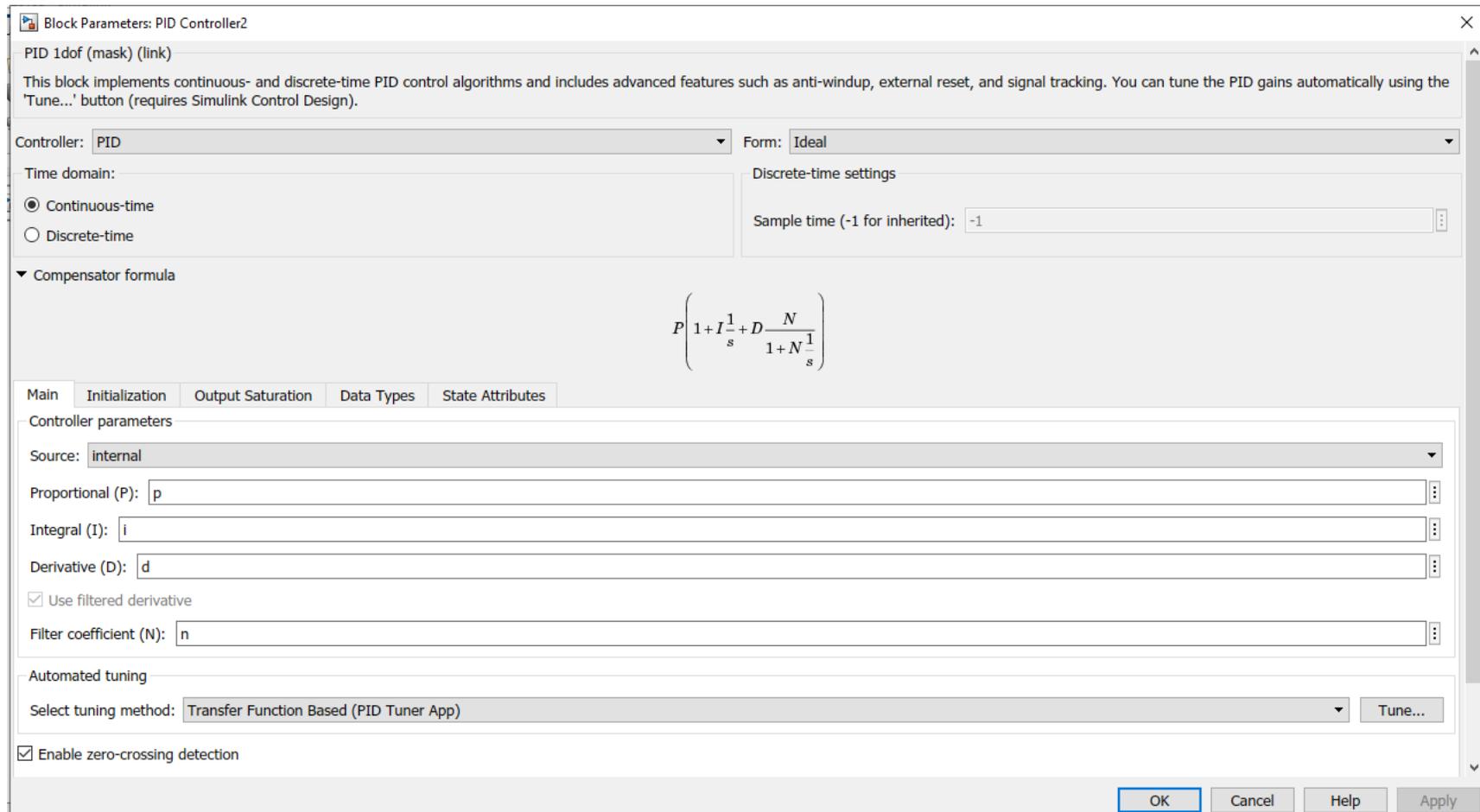


Figure 49: PID controller settings in MATLAB

A SimMechanics model generated by the command and presented in Figure 43 has been used to simulate the behavior of two-link manipulator. The Simulink model of the PID controller implemented on SimMechanics manipulator model has been presented in the Figure 50.

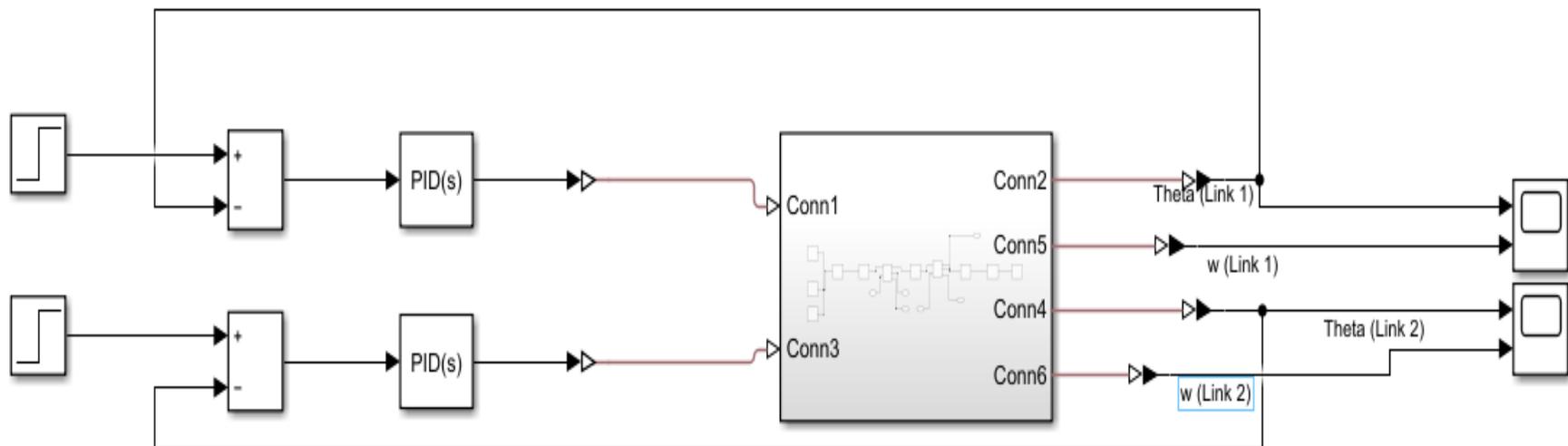


Figure 50: Complete Simulink model with PID controller

The model response has been simulated with a dynamic desired reference similar to what has been implemented with the mathematical model of the manipulator. An error signal has been generated between the desired reference of Link 1 and Link 2 and then this error signal has been fed to a PID controller, where PID generates actuation torque for both links based on its error minimization approach. The dynamical response of link 1 and link 2 has been presented against their desired reference in the Figure 51 and Figure 52 respectively.

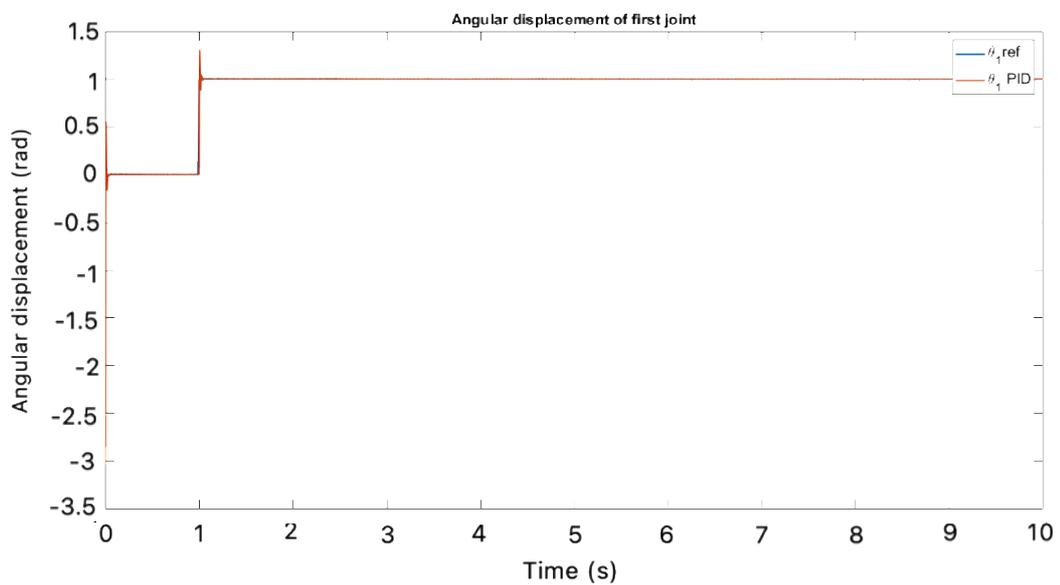


Figure 51: Angular displacement of link 1 of the simulated model with PID controller

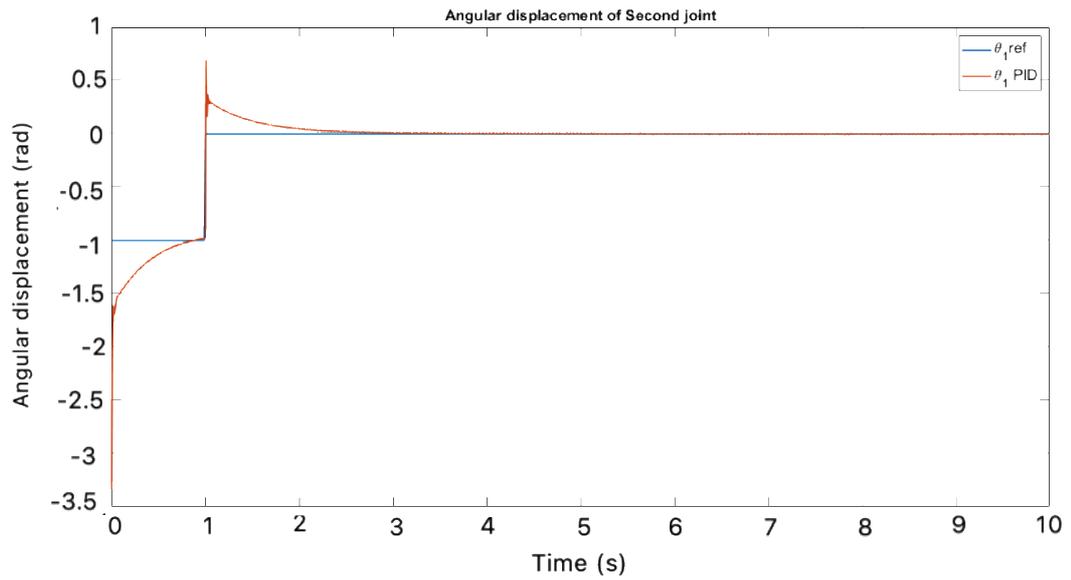


Figure 52: Angular displacement of link 2 of the simulated model with PID controller

From the simulated behavior of the system, it can be seen that the targeted link angular displacement has been tracked to the desired reference. As can be seen from the Figure 52, the PID controller performance is similar to a mathematical model implemented in the previous section of simulation and results. Similar large overshoots have been observed in angular displacements of both joints with the PID controller implementation. The error between the desired angular displacement and link 1 and link 2 dynamic value has been presented in the Figure 53.

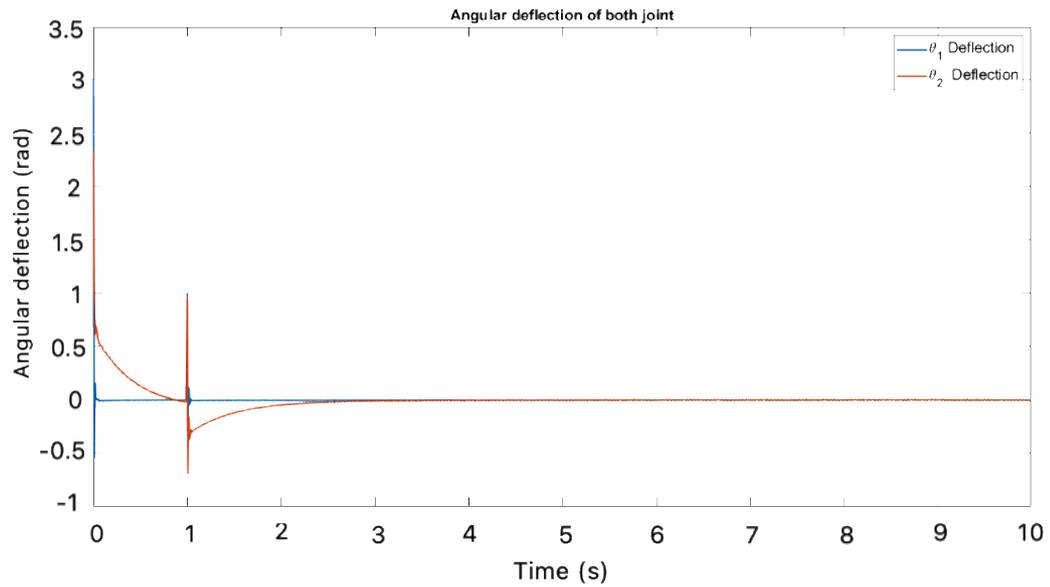


Figure 53: Angular deflection of both links of the simulated model with PID controller

As the initial conditions of the Autodesk model are at -3 rad and -2.4 rad for link 1 and link 2 respectively, initial phase of the response show some spikes. This presents the stability of the implemented system with different initial conditions. The actuation angle provided from the PID controller based on the error signal has been presented in the Figure 54.

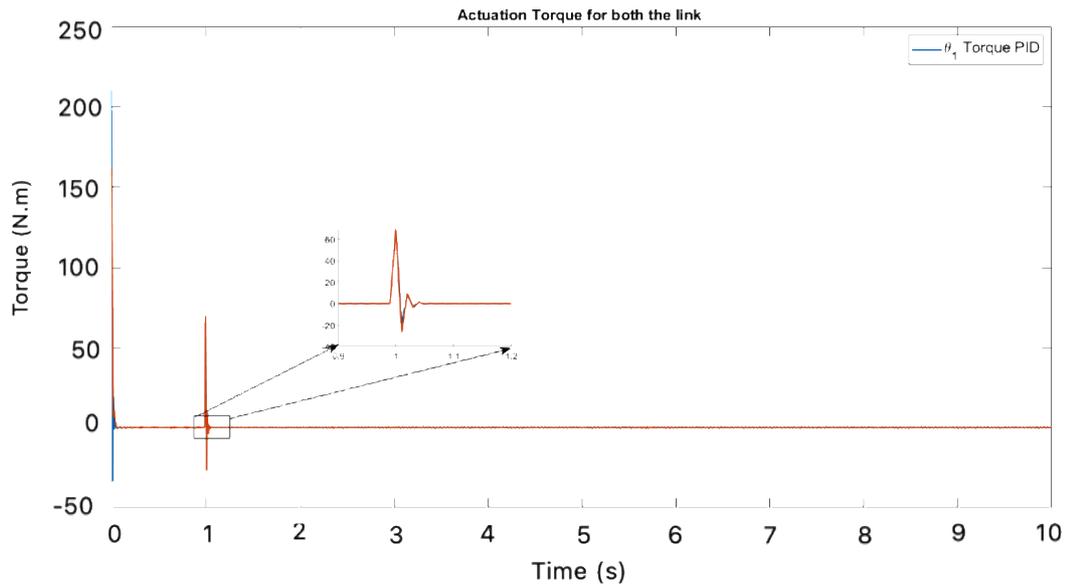


Figure 54: Actuation torque to both links of the simulated model with PID controller

It can be observed from Figure 54 that a large amount of torque has been provided by the controller to minimize the error, quickly resulting in a huge overshoot at the time of transition. These large pulses of torque lead to high spikes in response. The angular velocity and angular acceleration of the link 1 and link 2 has been presented in the Figure 55 and Figure 56.

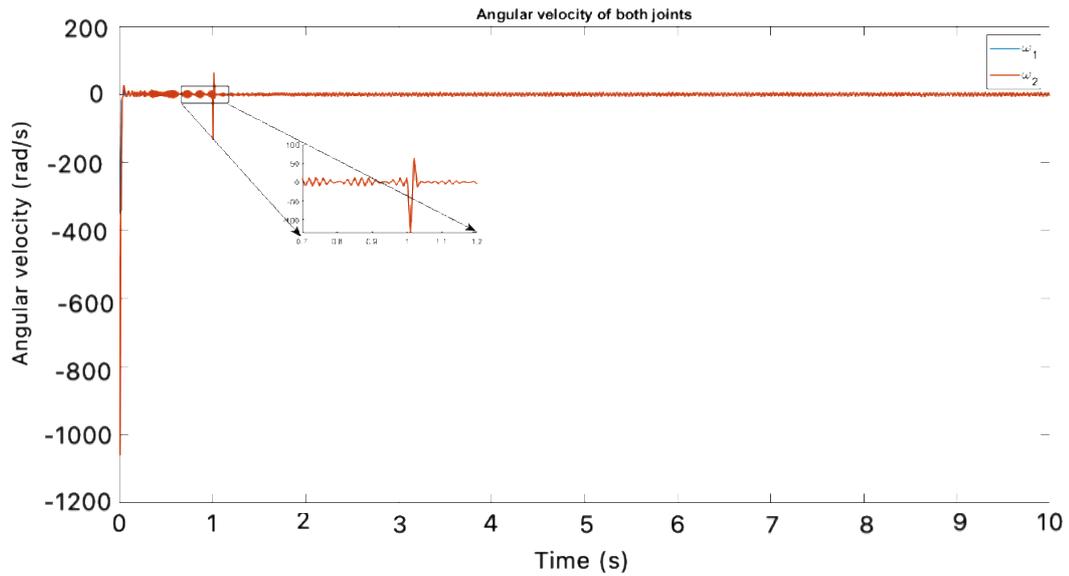


Figure 55: Angular velocity of both links of the simulated model with PID controller

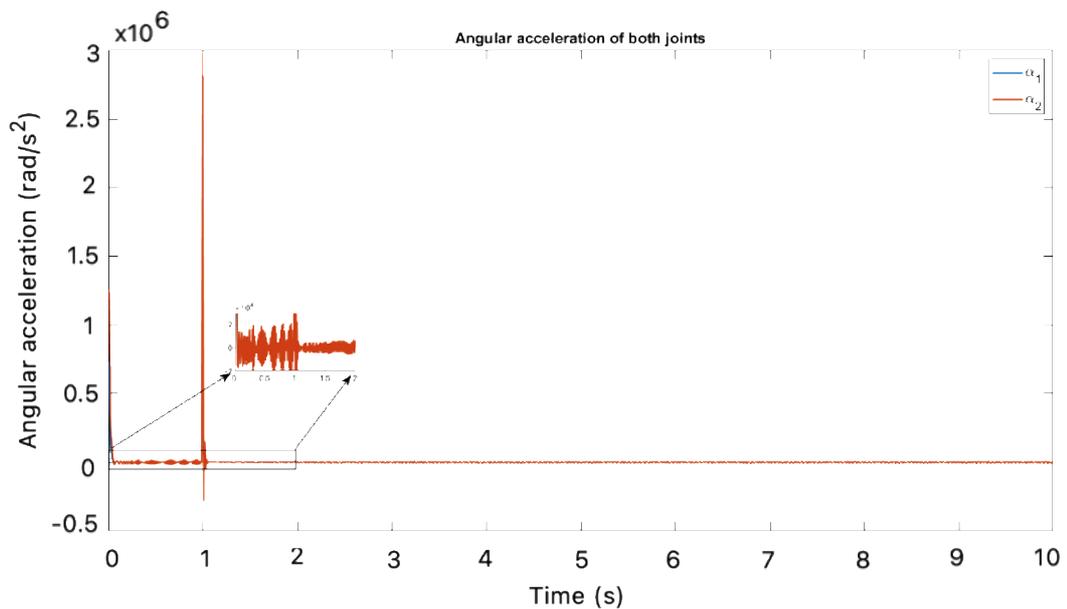


Figure 56: Angular acceleration of both links of the simulated model with PID controller

As can be seen from Figures 55 and 56, due to large spikes of actuation torque; large rate of change of angular displacement with time and large rate of changes of speed with time has been observed. Which shows the quick response time, but these large variation makes it difficult to implement a system in realistic scenario.

## **5.2 LQR Controller Implemented on SimMechanics Model of Manipulator**

To simulate the dynamic behavior of two link manipulator using the SimMechanics library, the following Simulink model has been used. To evaluate the LQR state feedback controller gain an m-file has been simulated to calculate the gains of the system. The Simulink model of the SimMechanics two-link manipulator with feedback LQR controller has been presented in the Figure 57.

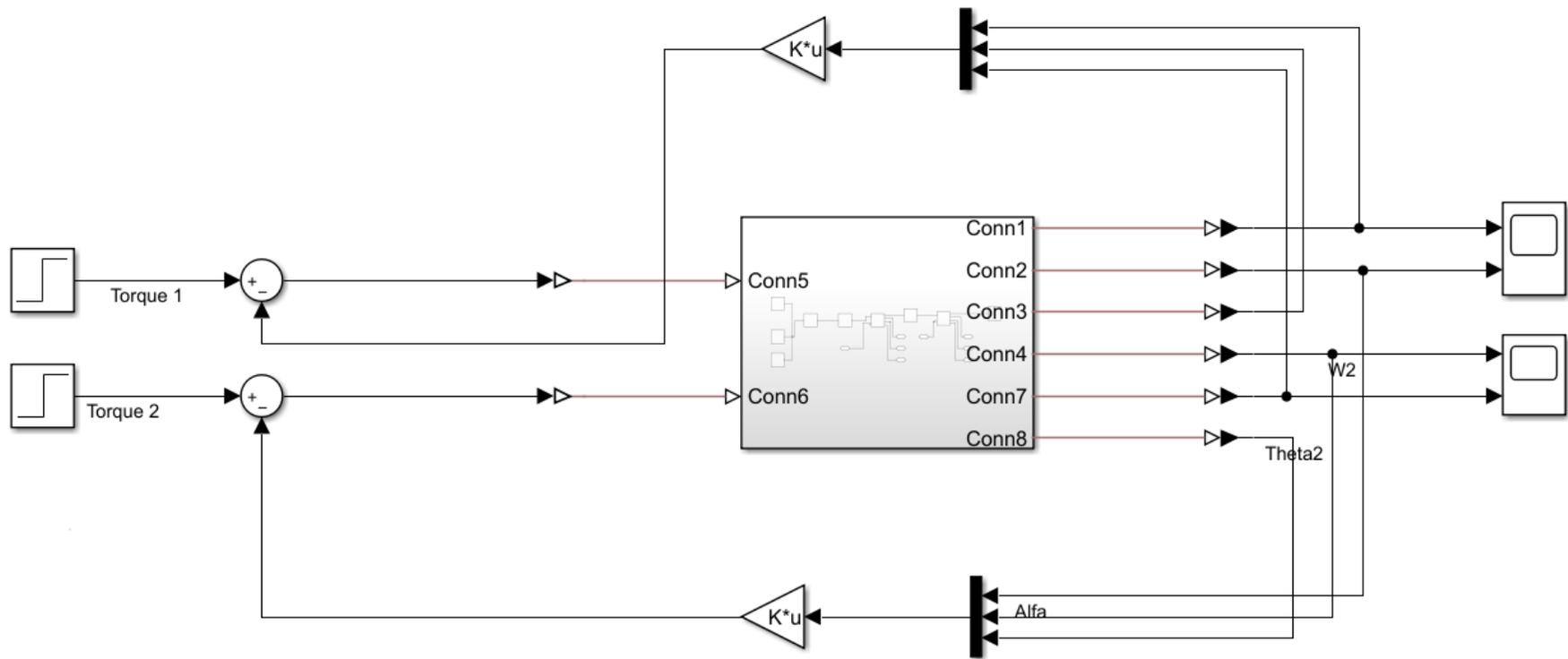


Figure 57: Simulation of two-link manipulator with LQR controller

The system response has been simulated with a dynamic desired reference for the angular displacement of link 1 and link 2. The actuation signal has been generated using state feedback LQR controller. Dynamical behavior of link 1 and link 2 has been presented in the Figure 58 and Figure 59.

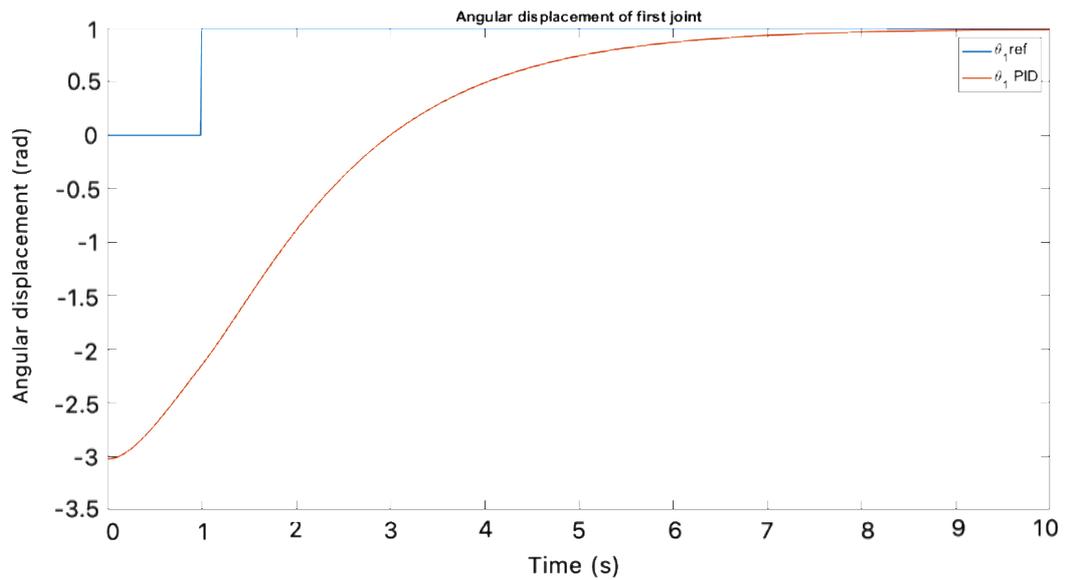


Figure 58: Angular displacement of link 1 of the simulated model with LQR controller

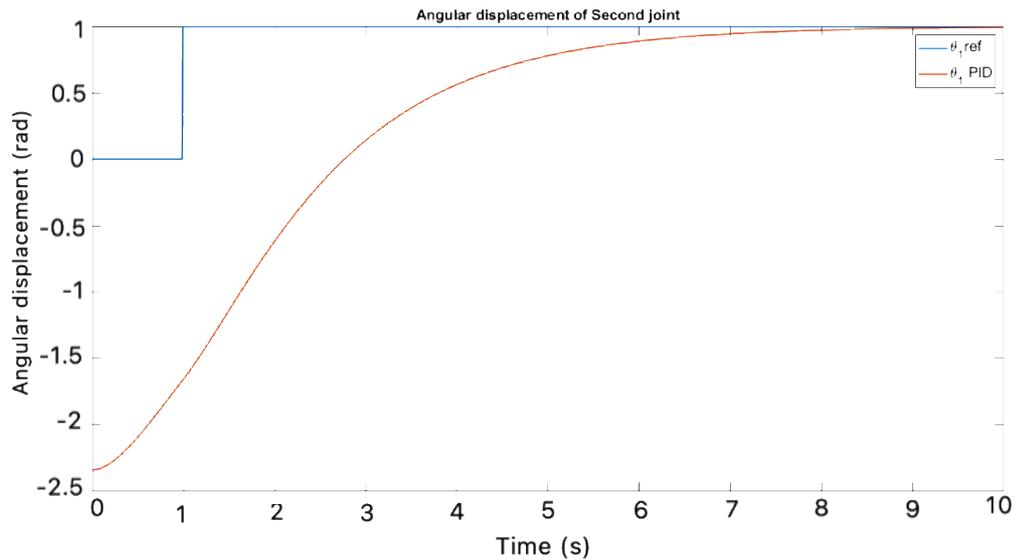


Figure 59: Angular displacement of link 2 of the simulated model with LQR controller

From Figures 58 and 59 it can be observed that the response of the manipulator in this case was subjected to a dynamical desired reference and controlled with a LQR feedback controller is slower compared to PID controller. The slow dynamics of the system are due to regulated actuation signal. The response of the system is similar to the model implemented using mathematical equations. However, similar to the mathematical model's system states, no overshoots are available. The angular deflection or error has been presented in the Figure 60.

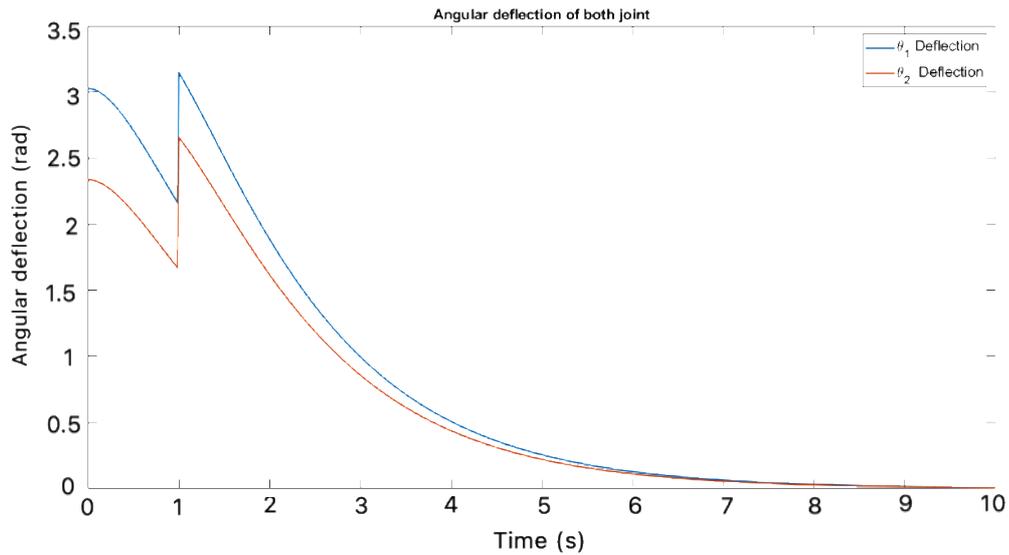


Figure 60: Angular deflection of both links of the simulated model with LQR controller

The error between the desired reference and the dynamic value of system state has been presented in Figure 60. It can be seen as the system starts at a desired location and soon, the actuation signal appears and tries to minimize the error by providing actuation torque at both links. The actuation torque to both links has been presented in the Figure 61.

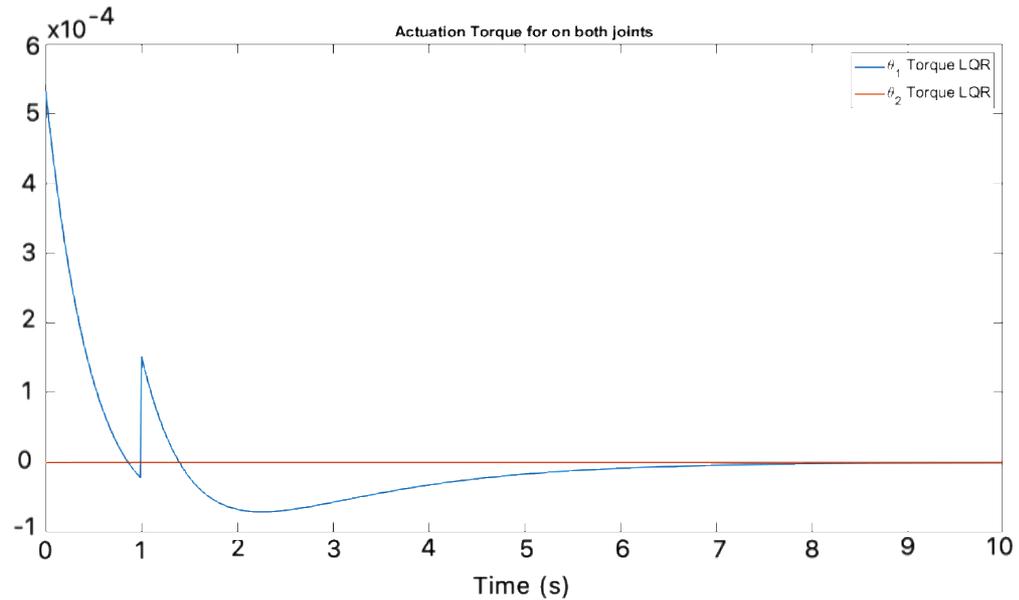


Figure 61: Actuation torque for both links of the simulated model with LQR controller

It can be observed from Figure 61 that when a change in a desired location of actuation signal appears, it tries to speed up the response by increasing the actuation torque to both links. The amount of actuation torques applied on both links is smaller than the actuation input generated by the PID controller. This optimized value of the actuation signal does not produce any overshoot in the system dynamic response. Due to small actuation torques, no chattering phenomena has been observed. The time dynamic response of angular velocity and angular acceleration has been presented in the Figure 62 and Figure 63, respectively.

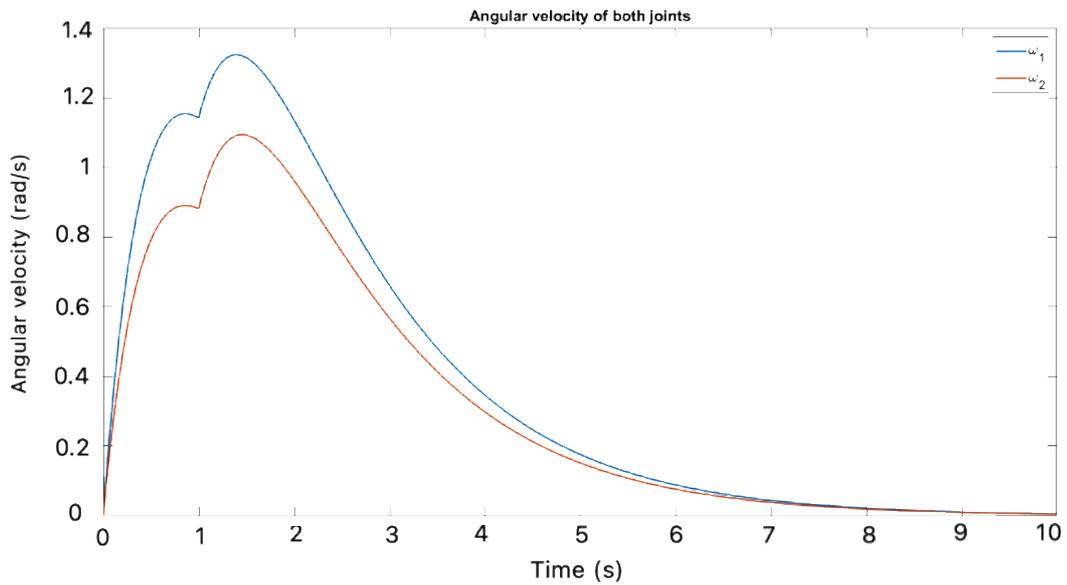


Figure 62: Angular velocity of both links of the simulated model with LQR controller

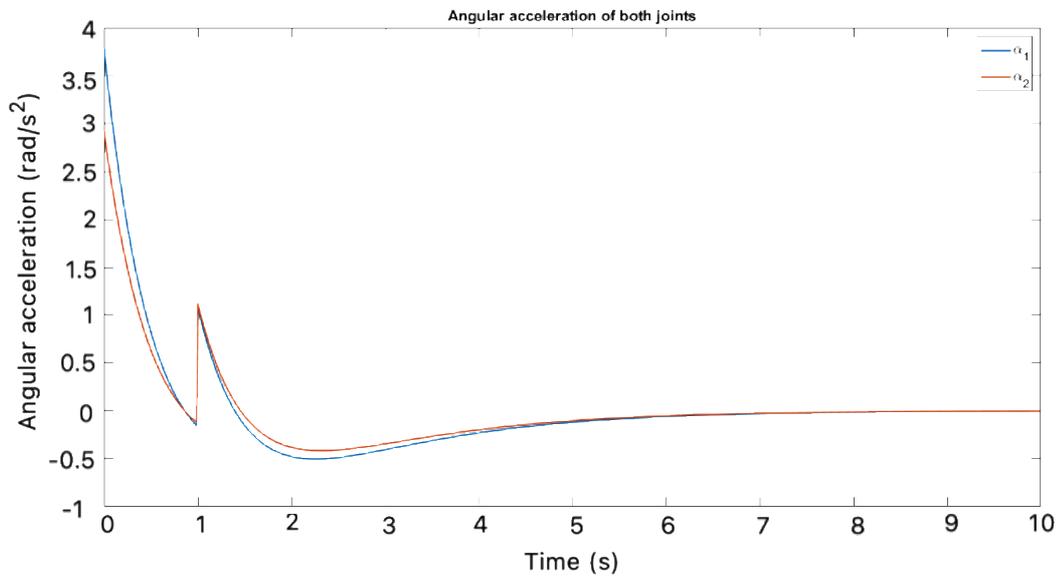


Figure 63: Angular acceleration of both joints of the simulated model with LQR controller

Figures 62 and 63 show the rate of change of angular displacement with respect to time and change in angular speed, it can be observed that this rate is much smaller than the rate of change which has been observed with the implementation of the PID

controller. This small change can be implemented on actual mechanical systems, whereas large changes to angular velocity and angular acceleration can cause electrical and mechanical losses due to jerks in the assembly.

### **5.3 LQG Controller Implemented on SimMechanics Model of Manipulator**

To simulate the dynamical behavior of system using LQG feedback controller, same SimMechanics and m-file generated for PID controller have been used. To derive the LQG gains for the simulation, mathematical equation of the system has been used. An m-file code has been simulated to get the required controller gains. Simulink model of the SimMechanics two-link manipulator with LQG feedback controller has been presented in the Figure 64.

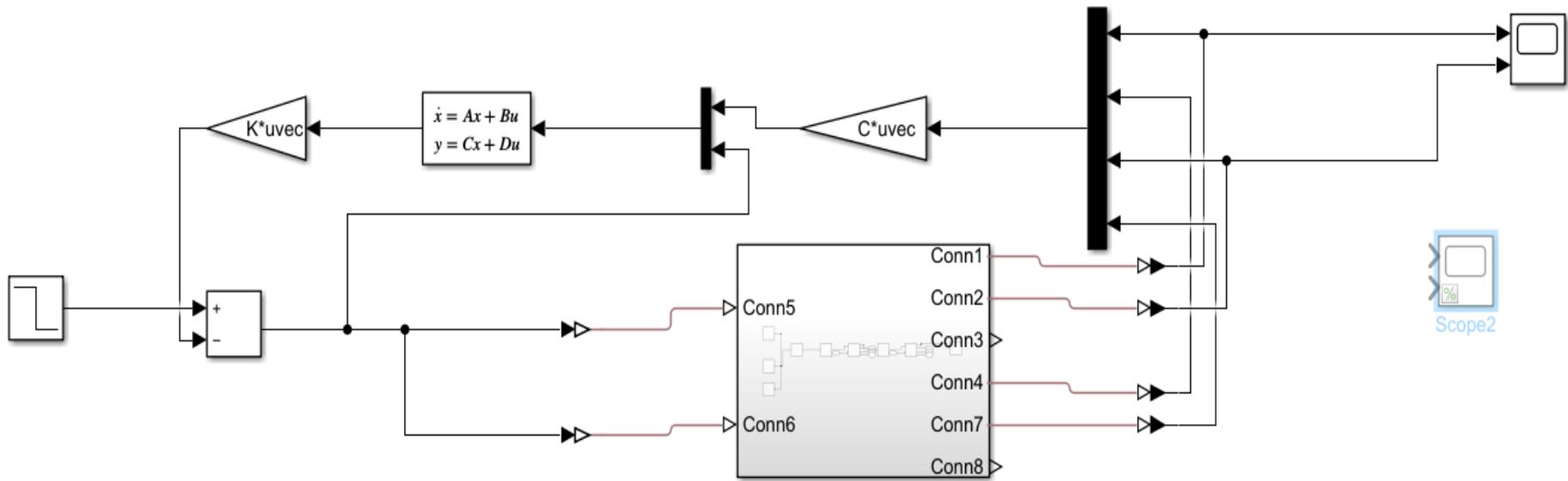


Figure 64: Simulink model of the LQG implemented controller in manipulator

The angular displacement of both links has been tracked to the desired reference. The dynamic behavior of link 1 and link 2 has been presented in the Figure 65 and Figure 66, respectively.

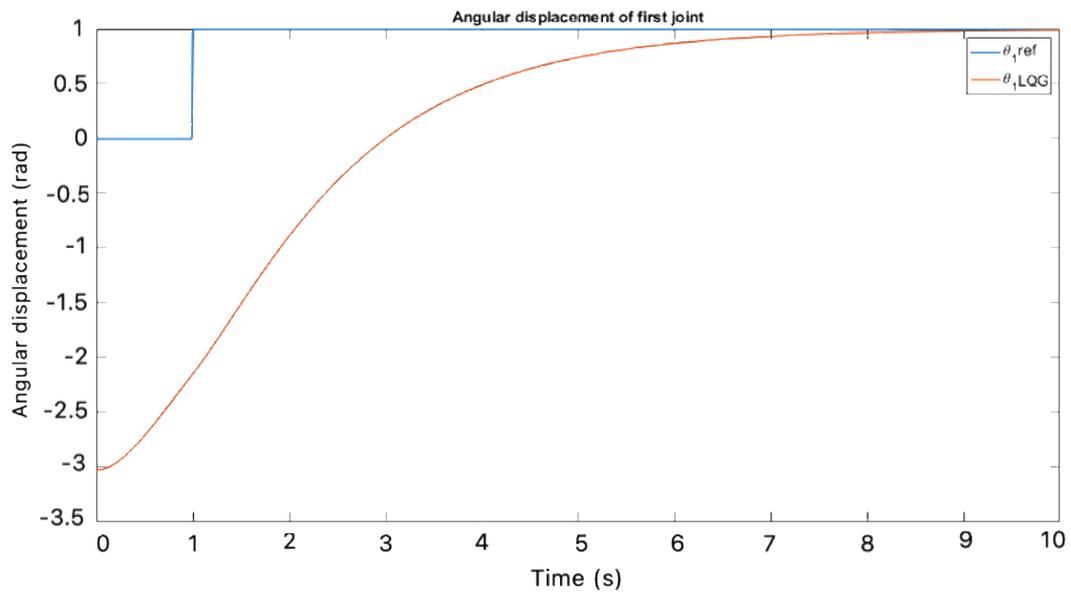


Figure 65: Angular displacement of link 1 of the simulated model with LQG controller

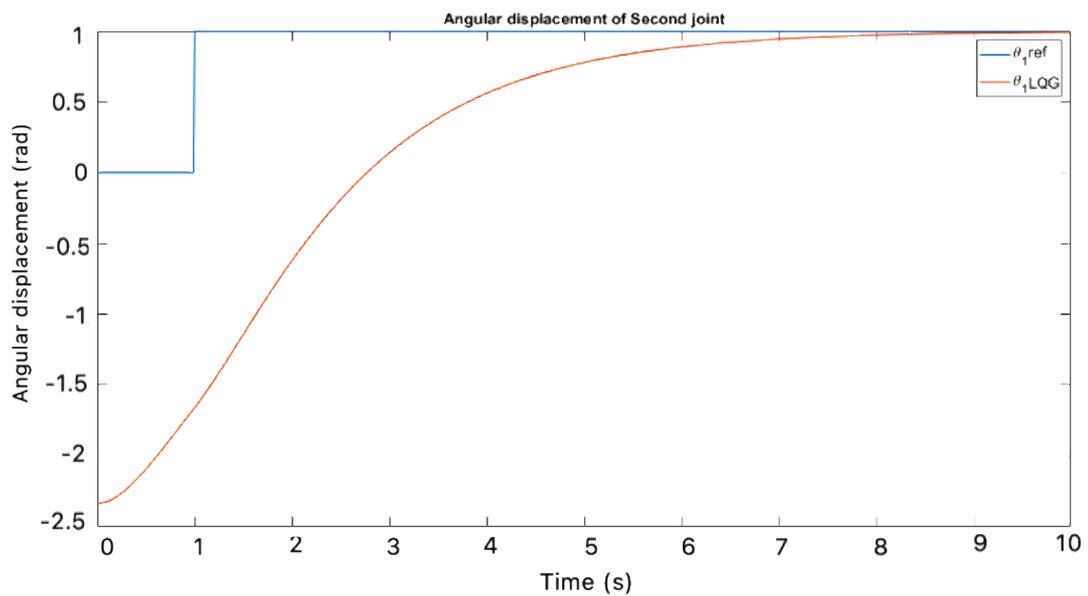


Figure 66: Angular displacement of link 2 of the simulated model with LQG controller

Angular displacement in both links has been observed as significantly smaller than it has been observed with the LQR controller, due to the LQE part of the LQG controller. The rise time of the LQG controller is better than the LQR controller. Similar to the LQR controller, the LQG controller does not show any overshoots in the system dynamic behavior. Steady state error is also negligible with the LQG controlled response. The error between the actual angular displacement and desired reference for both links has been evaluated and presented in the Figure 67.

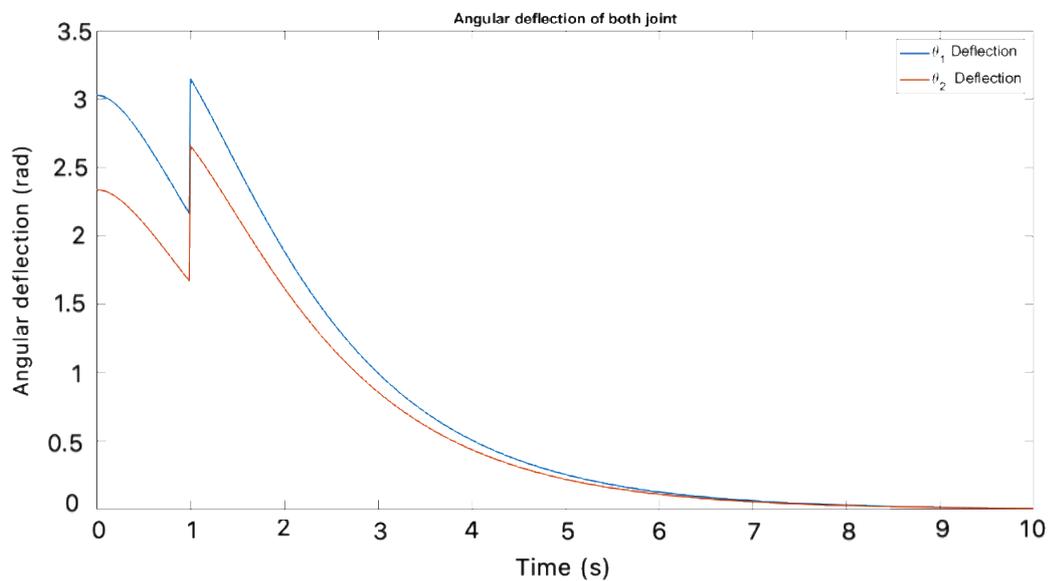


Figure 67: Angular deflection of both links of the simulated model with LQR controller

Angular deflection between both links is decreasing with a larger slop than the LQR controller but does not create large overshoots. This is due to better optimized actuation torque to both links. The actuation torque fed into both revolute has been presented in the Figure 68.

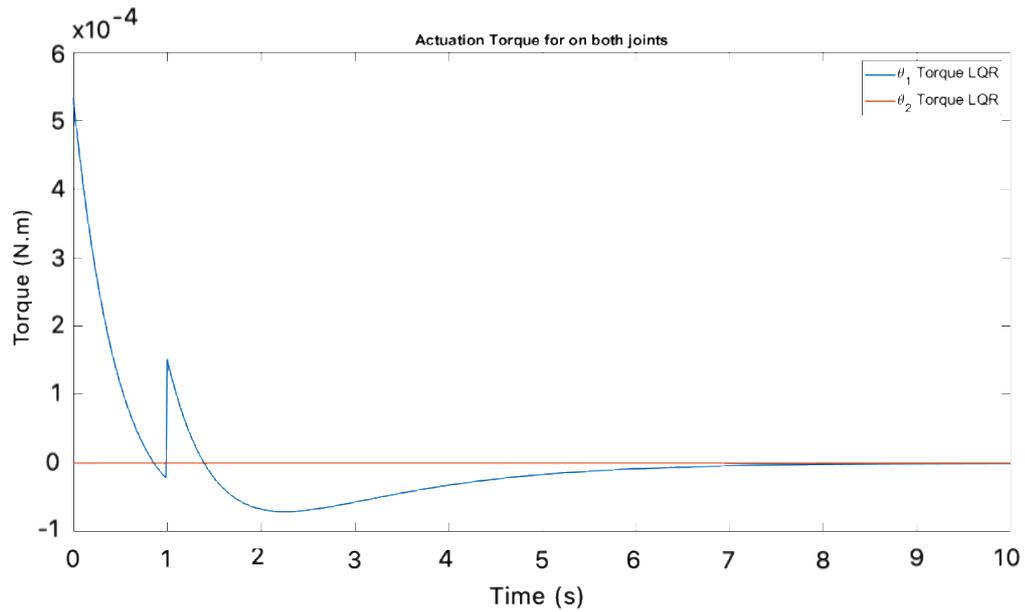


Figure 68: Actuation torque feed to both link actuators of the simulated model with LQR controller

It can be observed that the amount of torque feed into the actuators is significantly smaller than that generated by the PID controller. This smaller value of optimized torque does not take the system response quickly and leads to minimum overshoots. As can be observed due to optimized actuation torque, no overshoots from the desired reference have been observed in the dynamic response of both links' angular displacement. The angular velocity and angular acceleration of both links have been presented in the Figure 69 and Figure 70.

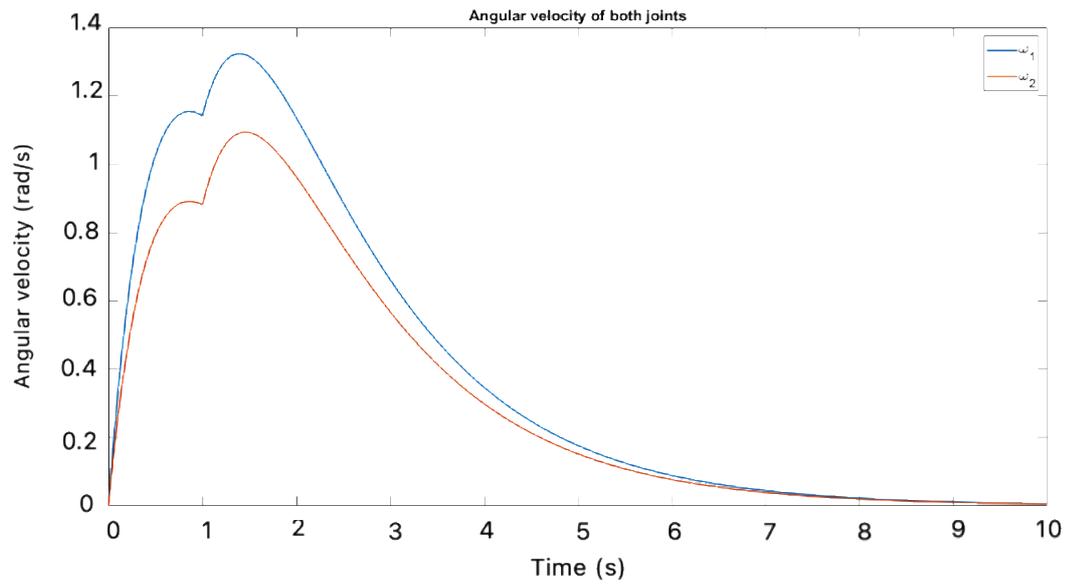


Figure 69: Angular velocity of both links of the simulated model with LQR controller

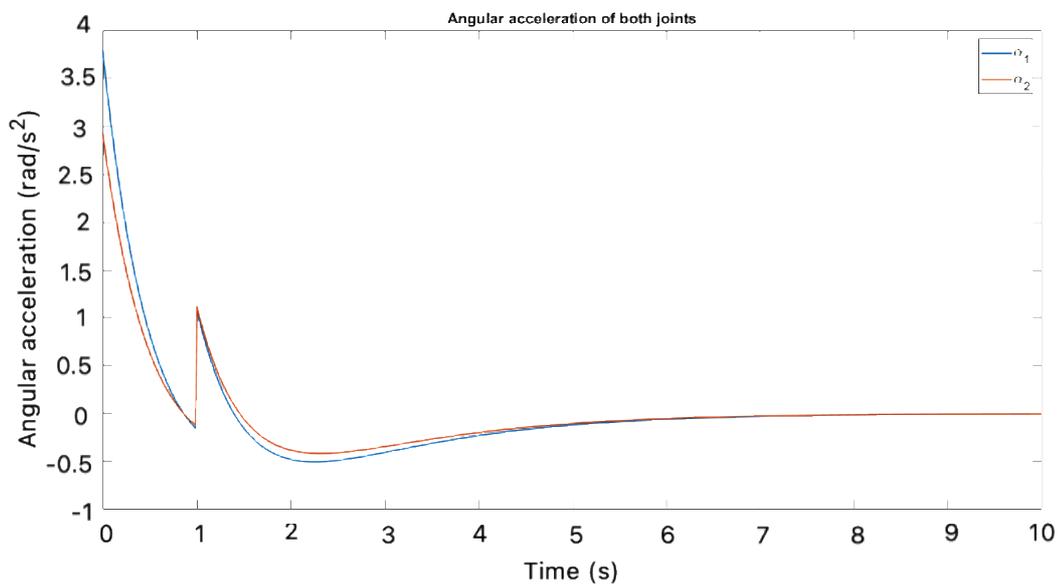


Figure 70: Angular acceleration of both links of the simulated model with LQR controller

From the implemented two-link rigid flexible manipulator actuator with LQG controller used to track the desired angular displacement of the link 1 and link 2, it has been observed that the system response is better than both PID and LQR feedback controllers. The system shows more stability when controlled with LQG controller. The amount of an angular velocity and angular acceleration assembled is realistic and will not cause any harm if this controller is implemented on a sensitive instrument.

#### 5.4 Comparative Analysis of PID, LQR and LQG Controllers

A comparative analysis has been done to present the performance analysis of PID, LQR and LQG controllers. Previously in this section, all three controllers were implemented and the dynamic response of them has been discussed, including how actuation torque is influencing the response of the system. The graphs of angular displacement of link 1 and link 2 with PID, LQR and LQG controllers has been presented in the Figure 71 and Figure 72 respectively.

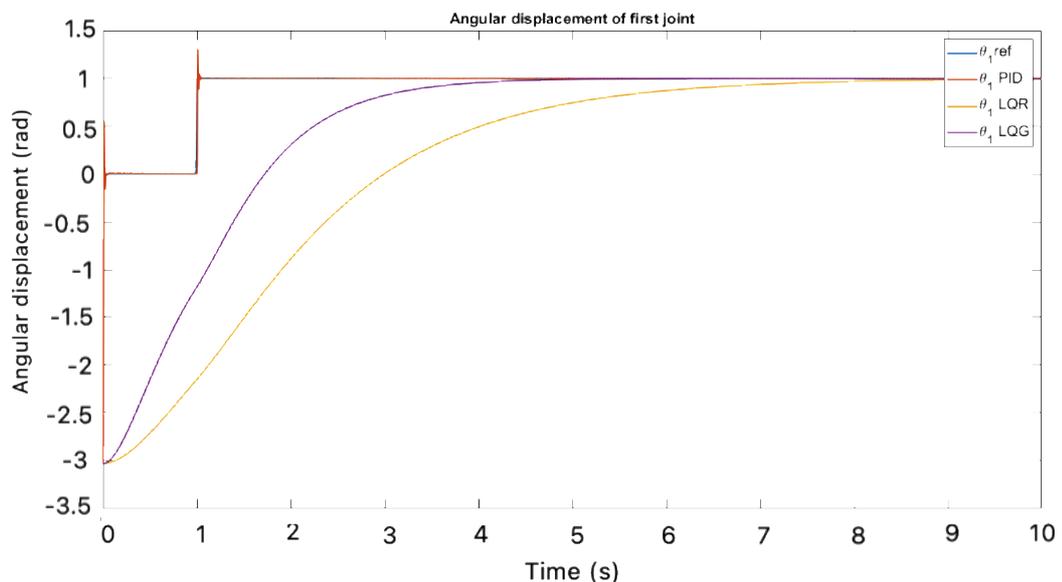


Figure 71: Angular displacement of link 1 of the simulated model with the three controllers

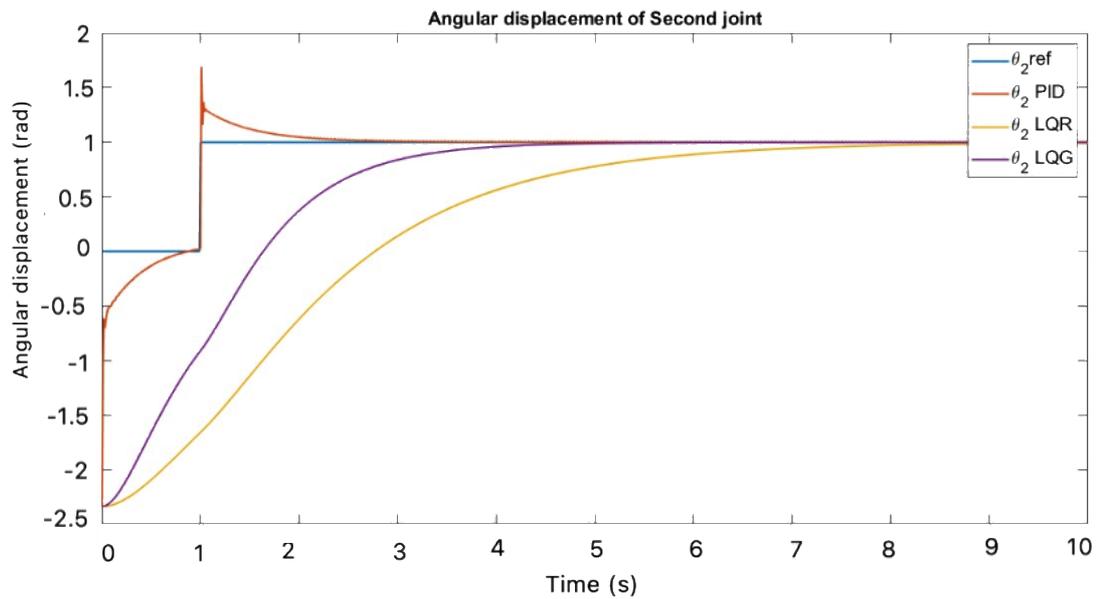


Figure 72: Angular displacement of link 2 of the simulated model with the three controllers

Dynamical response of angular displacement of link 1 and link 2 has been presented in Figures 71 and 72. It can be observed that the response of PID controller is faster than LQR and LQG controllers, but a large overshoot has been observed. But in the case of the LQR, response is smooth with no overshoot, but the rise time is very large. Similarly, in the case of the LQG controller, the response is best compared to PID and LQR. With the LQG controller, a shorter rise time compared to LQR and less overshoots compared to PID has been observed. The actuation torque for link 1 and link 2 has been presented in the Figure 73 and Figure 74.

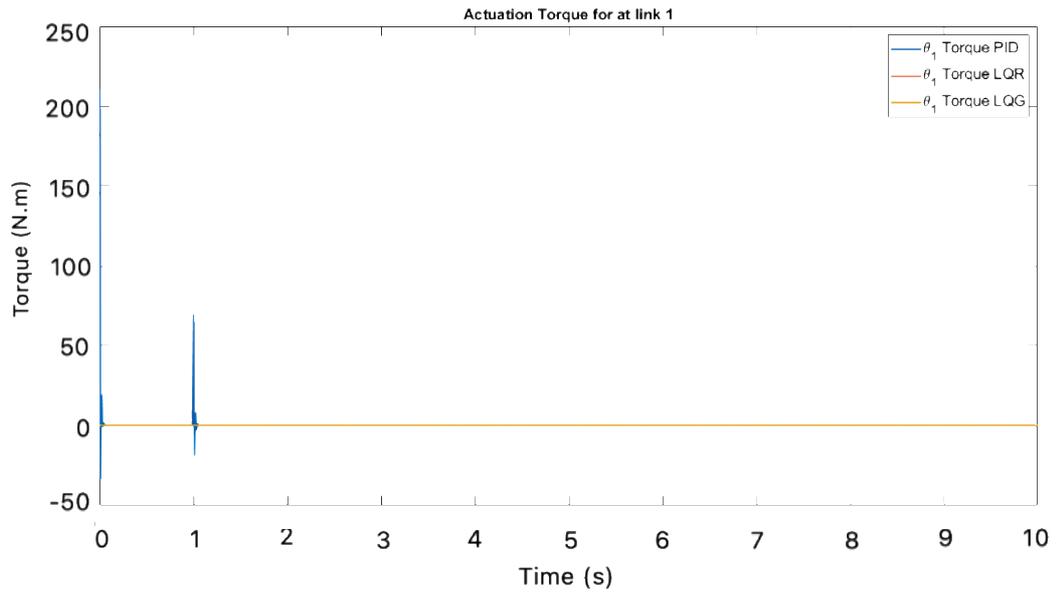


Figure 73: Actuation torque of link 1 of the simulated model with the three controllers

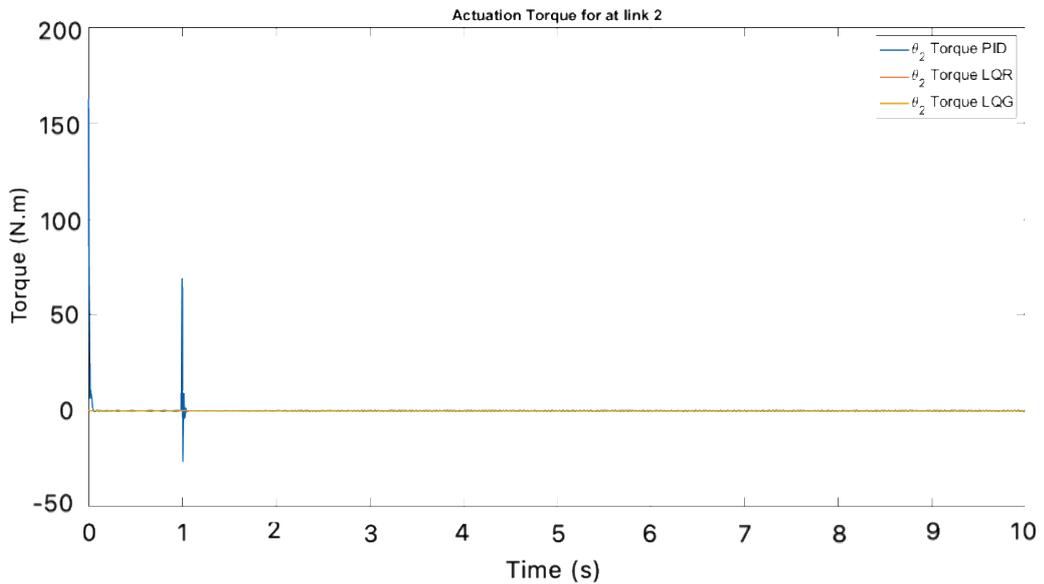


Figure 74: Actuation torque of link 2 of the simulated model with the three controllers

It can be observed for the actuation torques to the link 1 and link 2 that due to large actuation torques, when using PID, large overshoots have been resulted. In the

case of LQR and LQG controllers, the actuation torque is an optimal value which causes minimum overshoots.

The LQG controller performance is best compared to the other two controllers in terms of stability, rise time, settling time, overshoots, and steady state error. This enhanced performance will make it possible to implement a realistic scenario, as overshoots and chattering in motor can cause hysteresis, which can lead to electrical and mechanical losses. Large actuation signals cause a lot of heat in the mechanical motor, which also can lead to a hazardous scenario.

## Chapter 6: Conclusion and Future Work

For future research, some suggestions are proposed. For surgical robots, improving the path following the tip of the flexible links and reducing errors, while still using enhanced accuracy. A path following controller was needed to remove the limitation of dynamic model, but the mathematical model of this design is difficult.

Another controller can be used, such as Genetic Algorithm (GA), Artificial Intelligent (AI), and fuzzy controller to control the effect of the vibrations on the flexible manipulator. The simulation results and the response of the flexible manipulator without a PID controller have a large, undesired amplitude. In order to get a good response of system and to be more practical in daily applications, the robust controller and linear quadratic regulator should be used to improve the response of the system and take the vibration effects into account.

It has been noted from the figure that the high fluctuation and high amplitude of the hub angle and the deflection of the link without controlling and without implement the PID controller when we implement the PID, LQR and LQG controller. The enhancement in the stability of the system is important to decrease the vibration of the system and if the hub speed was increased, which in turn effects the deflection of the flexible link which also increases.

The Lagrange approach is used to derive the equation of motion of the rigid flexible link manipulator, because it is simpler than Hamilton's principle. A mathematical dynamic model of rigid-flexible links developed based on the use of the Finite Element Method and Lagrange's approach. The modeling technique is based on dividing each link into a large number of elements, which then finds the mass matrix

and stiffness matrix, and writes a MATLAB code. The mass matrix and stiffness matrix were computed for one element as seen in Chapter 3 and then written for the MATLAB code for more than one element. In this case, the mass matrix and stiffness matrix helped us to model a state space representation to find the state matrix.

The Autodesk Inventor model is used in a similar fashion and when all three controllers are applied to it in comparison, it will Simulink model and give satisfactory results. When studying this research, the deflection beam is affected by the flexibility of the system, which depends on many parameters such as cross section of beam, and dimension, type of the material, the type of boundary conditions and the value of the payload.

The proposed control that was presented in this research work can suppress the dynamic vibration of the flexible link and significantly improve the positioning accuracy of the flexible robot. It is shown by the simulation results that all the essential features of the flexible link have been modeled with good accuracy.

Another proposal for future work is to build a physical model of the manipulator, where the results obtained from the simulated models are compared to the results obtained from the physical model; to gain better consistency and reliability of simulated models.

## **6.1 Conclusion**

To analyze the behavior of any mechanical dynamic system, usually there are two ways to Evaluate the response of the system, either by making their mathematical model or by import the Autodesk system dynamics to MATLAB SimMechanics library, which evaluate the response of any dynamic system by considering detailed

linear and nonlinear properties of attached mechanical components. A two-link planner manipulator was used in this proposed study, both dynamical representations have been used to model mechanical structure. Linearized mathematical model of planner manipulator has been used to simulated behavior of the system, PID, LQR, and LQG controllers to achieve desired reference points has been designed using linear definition of the system. In PID controller, the system response quickly converges but shows large undershoot and undershoots in the angular position of both links, it is known that overshoots are the manifestation of the signal that exceeds its target; which is not desirable in any control system. Now the above shortcoming is overcome by the implementation of LQR and LQG, whose response properly converges with desired output response without any overshoots and undershoots. It has been concluded that if the hub speed was increased, the effect of the deflection of the flexible link is increased. The Lagrange approach was used to derive the equation of motion of the rigid flexible link manipulator because it is simpler than the Hamilton principle.

The other way to model the system is to import the Autodesk model to MATLAB SimMechanics library. The main advantage in importing the Autodesk mechanical model to MATLAB/Simulink using SimMechanics library is that it considers all non-linearities in the connected system components and present actual dynamical response of the system. A mathematical dynamic model of rigid-flexible links was developed based on using the Finite Element Method and Lagrange's approach. The modeling technique is based on dividing each link into multiple elements, and finding the mass matrix and stiffness matrix for those elements, then writing a MATLAB code in Autodesk. It was observed that the LQG system response has been improved in-terms of quick converges time when compared to the LQR response, because that the system state is estimated by the observer, which increase

the stability of the system by decreasing the vibration of the system. The proposed control that has been proposed in this research work can suppress the dynamic vibration of the flexible link and significantly improve the positioning accuracy of the flexible robot. The Autodesk modeling of the proposed system is best to converges the system response having dominant factor of non-linearities. The simulations show that all the essential features of the flexible link have been modeled with good accuracy.

## References

- [1] G. V. Narayana and P. Ramesh, "Modelling and Control of Single Link Manipulators for Flexible Operation by using Linearization Techniques" International Journal of Current Engineering and Technology ISSN, 2277-4106, vol. 3, no. 1, pp. 611–616, 2013.
- [2] K. Lochan, B. K. Roy, and B. Subudhi, "A review on two-link flexible manipulators," *Annu. Rev. Control*, vol. 42, pp. 346–367, 2016.
- [3] E. A. Alandoli, M. Sulaiman, M. Z. A. Rashid, H. N. M. Shah, and Z. Ismail, "A review study on flexible link manipulators", *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 2, pp. 93-99, 2016.
- [4] R. Fareh, M. Saad, and M. Saad, "Adaptive control for a single flexible link manipulator using sliding mode technique", in *2009 6th International Multi-Conference on Systems, Signals and Devices, Djerba, Tunisia, 2009*, pp. 1-6.
- [5] D. J. Inman, *Vibration: with control, measurement, and stability*, 2<sup>nd</sup> ed., Hoboken, New Jersey, US: Prentice Hall, 1989.
- [6] G. Hastings and W. Book, "A linear dynamic model for flexible robotic manipulators," *IEEE Control Syst. Mag.*, vol. 7, no. 1, pp. 61–64, 1987.
- [7] M. O. Tokhi, and A. K. M. Azad, *Flexible robot manipulators: modelling, simulation and control*, 1<sup>st</sup> ed., London, UK: Iet, 2008.
- [8] R. H. Cannon Jr, and E. Schmitz, "Precise Control of Flexible Manipulators", NASA technical reports server, Washington D.C., US, 1984.
- [9] A. De Luca and B. Siciliano, "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *IEEE Trans. Syst., Man Cybern.* vol. 21, no. 4, pp. 826-839, 1991.
- [10] S. Cetinkunt & W. L. Yu, "Closed-loop behavior of a feedback-controlled flexible arm: A comparative study," *The International journal of robotics research*, vol. 10, no. 3, pp. 263-275, 1991.
- [11] P. B. Usoro, R. Nadira, and S. S. Mahil. "A finite element/Lagrange approach to modeling lightweight flexible manipulators." *J. Dyn. Sys., Meas., Control*, vol. 108, no. 3, pp. 198-205, 1986.
- [12] W. H. Sunada, and S. Dubowsky. "On the dynamic analysis and behavior of industrial robotic manipulators with elastic members", *J. Mech., Trans., and Automation*, vol. 105, no. 1, pp. 42-51, 1983.
- [13] E. Bayo, J. G. De Jalon, and M. A. Serna. "A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems." *Computer methods in applied mechanics and engineering*, vol. 71, no. 2, pp. 183-195, 1988.
- [14] M. S. Pereira, and P. L. Proença, *Dynamic Analysis of Rigid-Flexible Mechanisms*. In *Concurrent Engineering: Tools and Technologies for Mechanical System Design*. NATO ASI Series, Apr. 6-11, 1992, Berlin, Germany. Springer, Berlin, Heidelberg, 1993.
- [15] M. De Smet, C. Liefoghe, P. Sas, and R. Snoeys. "Dynamic analysis of flexible structures using component mode synthesis", *J. Appl. Mech.*, vol. 56, no. 4, pp. 874-880, 1989
- [16] F. Rakhsha, and A. Goldenberg. "Dynamics modelling of a single-link flexible robot." In *1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, USA, 1985, pp. 984-989.

- [17] W. J. Book, and M. Majette. "Controller design for flexible, distributed parameter mechanical arms via combined state space and frequency domain technique" *J. Dyn. Sys., Meas., Control*, vol. 105, no. 4, pp. 245-254, 1983.
- [18] S. Nicosia, P. Tomei, and A. Tornambé. "Hamiltonian description and dynamic control of flexible robots" *Journal of robotic systems*, vol. 6, no. 4, pp. 345-361, 1989.
- [19] K. Park, and Y. Park. "Fourier-based optimal design of a flexible manipulator path to reduce residual vibration of the endpoint", *Robotica*, vol. 11, no. 3, pp. 263-272, 1993.
- [20] G. Hohenbichler, P. Plöckinger, and P. Lugner. "Comparison of a Modal-Expansion-and a Finite-Element-Model for a Two-Beam Flexible Robot Arm", In *Robot Control (Syroco'88)*, vol. 21, no. 16, pp. 35-39, 1989.
- [21] H. Bremer, and F. Pfeiffer. "Experiments with flexible manipulators", *Journal of Control Engineering Practice*, vol. 3, no. 9, pp.1331-1338, 1995.
- [22] A. Mustafa., and A. Al-Saif. "Modeling, simulation and control of 2-R robot", *Global Journal of Research in Engineering*, vol. 14, no. 1, 2014.
- [23] S. K. Dwivedy, and P. Eberhard. "Dynamic analysis of flexible manipulators, a literature review", *Journal of Mechanism and machine theory*, vol. 41, no. 7, pp. 749-777, 2006.
- [24] A. Abe, "Trajectory planning for residual vibration suppression of a two-link rigid-flexible manipulator considering large deformation", *Journal of Mechanism and machine theory*, vol. 44, no. 9, pp. 1627-1729, 2009.
- [25] S. M. Khot, N. P. Yelve, R. Tomar, S. Desai, and S. Vittal. "Active vibration control of cantilever beam by using PID based output feedback controller", *Journal of Vibration and Control*, vol. 18, no. 3, pp. 366-372, 2012.
- [26] M. Shahriari-Kahkeshi. "Anti-disturbance dynamic surface control scheme for a class of uncertain nonlinear systems with asymmetric dead-zone nonlinearity", *ISA transactions*, vol. 81, pp. 86-95, 2018.
- [27] R. M. Mahamood, and J. O. Pedro. "Hybrid PD/PID controller design for two-link flexible manipulators", In *2011 8th Asian Control Conference (ASCC)*, Kaohsiung, Taiwan, 2011, pp. 1358-1363.
- [28] A. Dehghani, and H. Khodadadi. "Fuzzy logic self-tuning PID control for a single-link flexible joint robot manipulator in the presence of uncertainty", In *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, Busan, Korea (South), 2015, pp. 186-191.
- [29] V. Veselý, and A. Ilka."Design of robust gain-scheduled PI controllers", *Journal of the Franklin Institute*, vol. 352, no. 4, pp. 1476-1494, 2015.
- [30] J. T. Agee, S. Kizir, and Z. Bingul. "Intelligent proportional-integral (iPI) control of a single link flexible joint manipulator", *Journal of Vibration and Control*, vol. 21, no. 11, pp. 2273-2288, 2015.
- [31] A. Kumar, V. Kumar, and P. J. Gaidhane. "Optimal design of fuzzy fractional order  $PI\lambda D\mu$  controller for redundant robot", *Journal of Procedia Computer Science*, vol. 125, pp. 442-448, 2018.
- [32] I. H. Akyuz, E. Yolacan, H. M. Ertunc, and Z. Bingul. "PID and state feedback control of a single-link flexible joint robot manipulator", In *2011 IEEE International Conference on Mechatronics*, Istanbul, Turkey, 2011, pp. 409-414.
- [33] S. Lee, and C. Lee. "Hybrid control scheme for robust tracking of two-link flexible manipulator", *Journal of Intelligent and Robotic Systems*, vol. 34, no. 4, pp. 431-452, 2002.

- [34] H. A. Talebi, K. Khorasani, and R. V. Patel. "Tracking control of a flexible-link manipulator using neural networks: experimental results", *Robotica*, vol. 20, no. 4, pp. 417-427, 2002.
- [35] C. Wei, and D. Söffker. "Optimization strategy for PID-controller design of AMB rotor systems", *IEEE transactions on control systems technology*, vol. 24, no. 3, pp. 788-803, 2015.
- [36] S. Jiang, Jian Xu, and Yao Yan. "Stability and oscillations in a slow-fast flexible joint system with transformation delay", *Acta Mechanica Sinica*, vol. 30, no. 5, pp. 727-738, 2014.
- [37] M. A. Ahmad, Z. Mohamed, H. Ishak, and A. N. K. Nasir. "Vibration Suppression Techniques in Feedback Control of a Very Flexible Robot Manipulator", In *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, Kuala Lumpur, Malaysia , 2008, pp. 469-474.
- [38] H. Nahvi, and H. Ahmadi. "Dynamic simulation and nonlinear vibrations of flexible robot arms", *Journal of Applied Sciences*, vol. 3, no. 7, pp. 510-523, 2003.
- [39] K. Nanos, and E. G. Papadopoulos. "On the dynamics and control of flexible joint space manipulators", *Journal of Control Engineering Practice*, vol. 45, pp.230-243, 2015.
- [40] R. N. Jazar, *Theory of Applied Robotics*, 2<sup>nd</sup> ed., Berlin, Germany: Springer, 2010.
- [41] A. San-Millan, V. Feliu, and A. Garcia. "A two-stage control scheme of single-link flexible manipulators", In *2015 23rd Mediterranean Conference on Control and Automation (MED)*, Torremolinos, Spain, 2015, pp. 1098-1105.
- [42] A. A. Mohammed, and A. Eltayeb. "Dynamics and control of a two-link manipulator using pid and sliding mode control", In *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEE)*, Khartoum, Sudan, 2018, pp. 1-5.
- [43] S. S. Rao, and F. F. Yap., *Mechanical Vibrations*, 5<sup>th</sup> ed., Boston, US: Pearson 2011.

## Appendices

### Appendix A: Newton's Second Law Dynamic Modelling

The next work will include addressing the robotic arm kinematic, dynamics and corresponding control.

Forward kinematics will be implemented in a Cartesian coordinates system. By this means all possible displacement and orientation of movement will be governed into time dependent equations to be derived into velocity and acceleration.

Schematic diagram for the two links manipulator is shown in Figure 75. Equation of motion for this manipulator assembly can be created depending on the free body diagram and kinetic diagram for both links. Links 1 and 2 have lengths  $L_1$  and  $L_2$ , respectively, as shown in the graph. Link 1 has a revolute joint  $R_1$  assembled with the ground, while link two has revolute joint  $R_2$  assembled with Link 1.

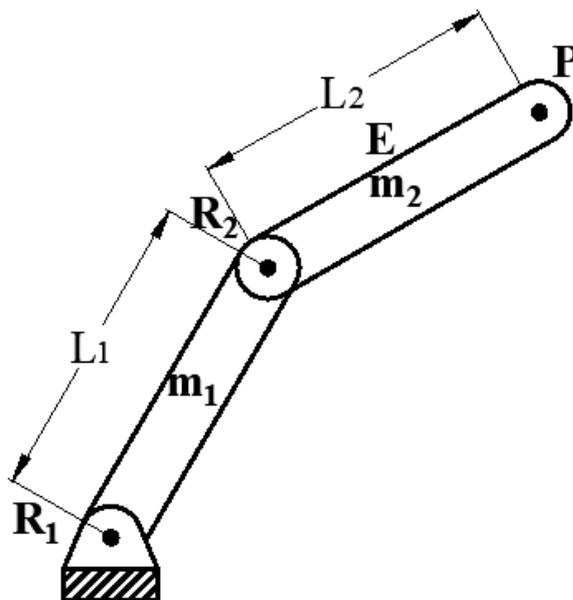


Figure 75: Free body diagram schematic for two-link manipulator

## Appendix B: Mathematical Description for Dynamic Model

The governing equation of motion for link 1 is described below. The angular momentum of inertia for link 1 is taken into consideration as the movement for this link is rotational about pivot R1. A schematic diagram for link 1 is illustrated in Figure 76. This link will turn with angle  $\theta_1$  and possess corresponding angular mass  $J_1$ . The assembly of these links will have input torque applied by a motor acting on link 1 at R1 joint. The displacement at the end position of link one is described by  $x_1$ . A spring acting at the middle of link 1 has a stiffness value of  $k_1$ .

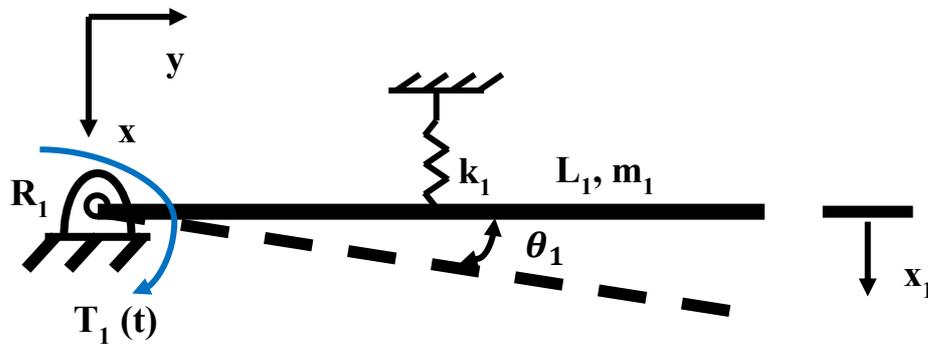
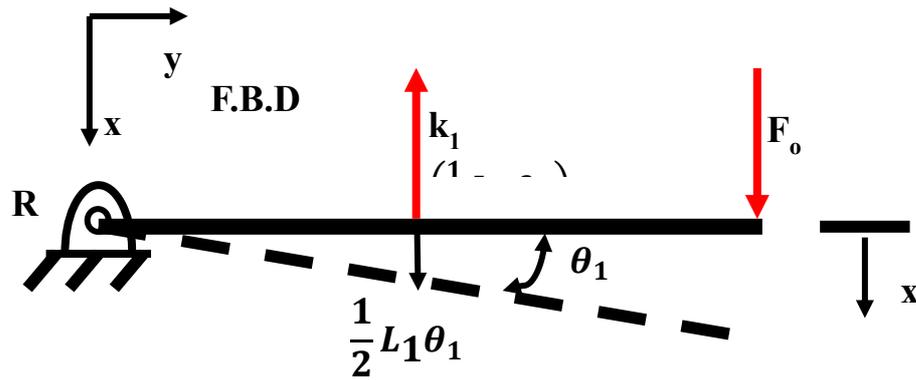
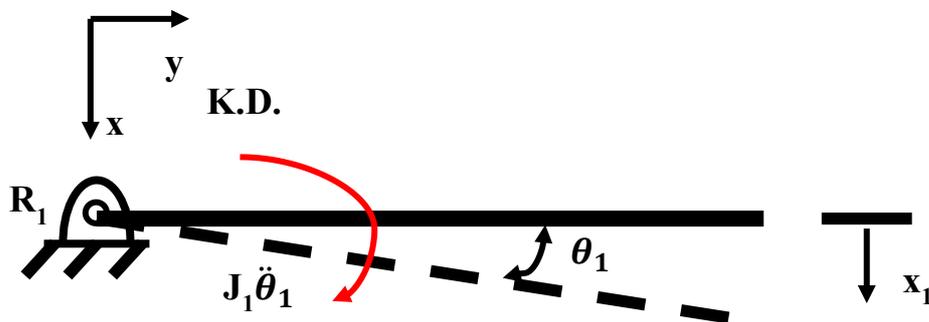


Figure 76: Link 1 schematic diagram

Applying Newton's second law was performed to generate a corresponding equation of motion for link 1. Free body (F.B.D) and kinetic (K.D.) diagrams are required in order to describe forces and help develop corresponding equation of motion. These diagrams are presented in Figure 77.



a) Free body diagram



b) Kinetic diagram

Figure 77: Link 1 free body and kinetic diagrams Free-body diagram and Kinetic diagram

The torque acting on R1 revolute joint is reported in Equation (44) and the corresponding equivalent force  $F_1$  acting at the end of link 1 is described in Equation (45).

$$T_1(t) = T_0 \cos \omega t \quad (44)$$

$$T_0 \cos \omega t = L_1 F_0 \cos \omega t = F_1(t) L_1 \quad (45)$$

Equation (45) shows that the torque acting on joint R1 has an equivalent force input  $F_1(t)$  acting at joint R2. The moment equation about pivot R1 is described in Equation (46) and (47) with clockwise direction considered as positive direction:

$$\sum M_{R1} = \sum M_{\text{effective}} \quad (46)$$

$$F_0 L_1 \cos \omega t - k_1 \theta_1 \left( \frac{L_1}{2} \right)^2 = J_1 \ddot{\theta}_1 \quad (47)$$

In order to sort  $M_R$ , angular moment of inertia  $J_1$  is defined as:

$$J_1 = \frac{1}{3} m_1 L_1^2 \quad (48)$$

hence, Equation (47) can be rewritten as:

$$F_0 L_1 \cos \omega t = \frac{1}{3} m_1 L_1^2 \ddot{\theta}_1 + k_1 \theta_1 \left( \frac{L_1}{2} \right)^2 \quad (49)$$

The homogeneous solution for the displacement function can be determined as presented in Equation (50):

$$x_1(t) = x_h(t) + x_p(t) \quad (50)$$

Where,

$$x_h(t) = A \sin \omega_n t + B \cos \omega_n t \quad (51)$$

The factors  $A$  and  $B$  can be calculated from the initial and boundary conditions. The corresponding  $\omega_n$  is evaluated as per Equation (52).

$$\omega_n = \sqrt{\frac{k_{1,eq}}{m_{1,eq}}} \quad (52)$$

The equivalent stiffness can be evaluated by kinetic energy equation for Link 1 as shown in Equations (53) and (54):

$$\frac{1}{2} k_{1,eq} (L_1 \theta_1)^2 = \frac{1}{2} k_1 \left( \frac{L_1}{2} \theta_1 \right)^2 \quad (53)$$

$$k_{1,eq} = \frac{k_1}{4} \quad (54)$$

Similarly, the equivalent mass for link 1 ( $m_{1,eq}$ ) can be obtained as follows:

$$\frac{1}{2} m_{1,eq} (\dot{x}_1)^2 = \frac{1}{2} J_1 (\dot{\theta}_1)^2 \quad (55)$$

therefore,

$$m_{1,eq} = \frac{m_1}{3} \quad (56)$$

Link 1 system can then be transformed to simple spring mass system as illustrated in Figure 78

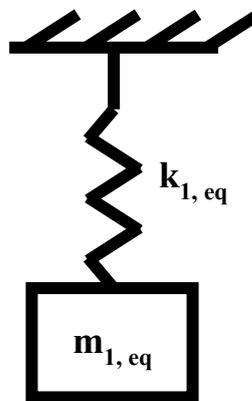


Figure 78: Equivalent simple spring mass system for link 1

The corresponding representation to  $\ddot{\theta}_1$  is defined in Equation (57):

$$x_1 = L_1 \theta_1 \quad (57)$$

Equation (49) can then be written in terms of one variable ( $x_1$ ) instead of  $\theta_1$  as shown in Equation (58).

$$F_1(t) = \frac{1}{3} m_1 L_1 \ddot{x}_1 + \frac{1}{4} k_1 L_1 x_1 \quad (58)$$

This equation form can also be solved with Laplace transformation considering  $x_1(t)$  as an output and  $F_1(t)$  as the corresponding input. This can produce the displacement

output at the revolute joint R2. Regarding the particular solution for the displacement  $x_{1p}(t)$  the following Equation (59) can be used:

$$x_{1p}(t) = \bar{\theta}_1 \cos \omega t \quad (59)$$

Where  $X_1$  is the amplitude value for the displacement  $x_1$  and the corresponding formula for  $\bar{\theta}_1$  is presented in Equation (60).

$$X_1 = \frac{F_o}{\sqrt{(k_{1,eq} - m_{1,eq}\omega^2)^2 + c^2\omega^2}} \quad (60)$$

As described in this section no damping is included for this system, hence the factor  $c^2\omega^2$  is equal to zero in (60).

Link 2 analysis are performed accordingly. As shown in Figure 79 link 2 is connected with link 1 at revolute joint R2, and exhibit two types of motion. The two motions exerted at link 2 are translational and rotational motions. Link 2 has length  $L_2$  and mass  $m_2$ .

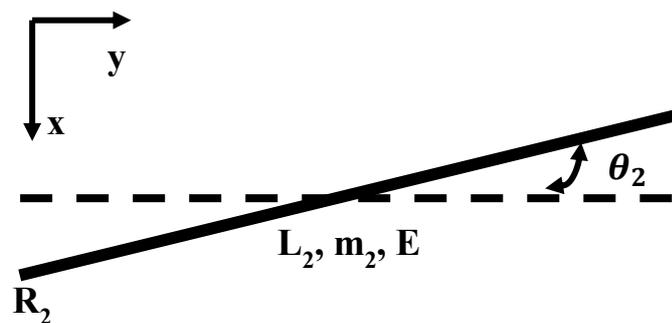
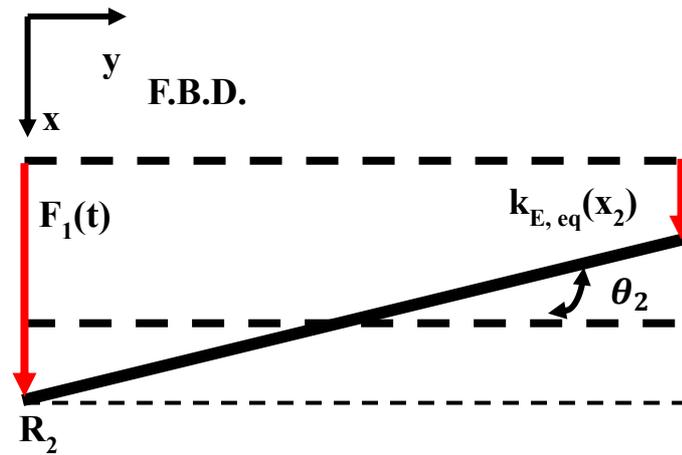
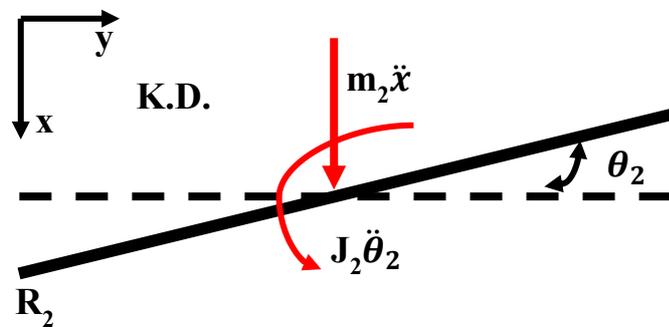


Figure 79: Link 2 schematic diagram

Elasticity value for link 2 is considered in the dynamic analysis. Angle of rotation for link 2 is  $\theta_2$  and the linear motion for this link is presented as  $x$ . The input force  $F_1(t)$  that acts on revolute joint R2 is included in the analysis for Link 2. The corresponding free body diagram and kinetic diagram are shown in Figure 80.



a) Free body diagram



b) Kinetic diagram

Figure 80: Link 2 free body and kinetic diagrams: Free-body diagram and Kinetic diagram

As link 2 is flexible beam the equivalent mass for this beam is evaluated by assuming the shape of the beam during motion to be parabolic. Therefore, the value for  $m_{2,eq}$  can be evaluated using Equation (61).

$$x = x_2 \left( \frac{y}{L_2} \right)^2 \quad (61)$$

hence applying the kinetic energy equation equivalent mass can be evaluated as per Equations (62) to (64).

$$\frac{1}{2} m_{2,eq} (\dot{x}_2)^2 = \frac{1}{2} \int_0^L \frac{m_2}{L_2} \dot{x}^2 dy \quad (62)$$

$$m_{2,eq} \dot{x}_2^2 = \dot{x}_2^2 \frac{m_2}{L_2} \int_0^L \left(\frac{y}{L_2}\right)^4 dy \quad (63)$$

$$m_{2,eq} = \frac{m_2}{5} \quad (64)$$

Also, this flexible beam possesses elasticity value ( $E_2$ ) and this value translates to equivalent stiffness. The equivalent value for stiffness associated due to beam elasticity is evaluated through Equation (65) [43]. Determining the equivalent values for mass ( $m_{2,eq}$ ) and stiffness ( $k_{E,eq}$ ) is mandatory in order to simplify link 2 to equivalent spring mass system as shown in Figure 81.

$$k_{E,eq} = \frac{3EI}{L_2^3} \quad (65)$$

Where  $I$  is moment of inertia for link 2 and can be evaluated for the rectangular cross sectional area presented in Figure 82.

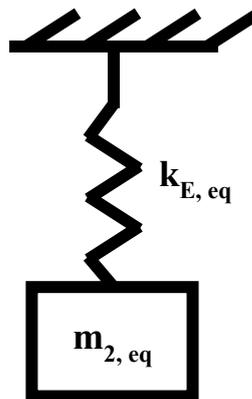


Figure 81: Equivalent simple spring mass system for link 2

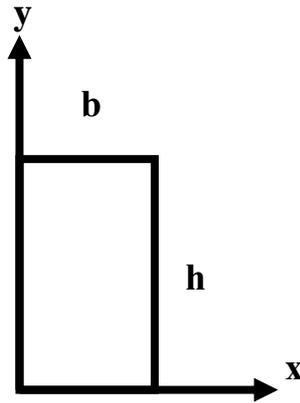


Figure 82 :Link 2 cross sectional area

The corresponding value for moment of inertia can be calculated by using Equation (66).

$$I = \frac{bh^3}{12} \quad (66)$$

Applying the momentum about R2 with considering counterclockwise as a positive direction in Equation is presented in (67) and (68).

$$\sum M_{R2} = \sum M_{effective} \quad (67)$$

$$k_{E,eq}L_2(x_1 - L_2\theta_2) = J_2\ddot{\theta}_2 \quad (68)$$

Where  $J_2$  is the angular mass for link 2 and is evaluated in Equation (69).

$$J_2 = \frac{1}{3}m_2L_2^2 \quad (69)$$

Applying Newton's second law for link 2 is addressed in Equations (70) and (71) below.

$$\sum F_2 = \sum F_{effective} \quad (70)$$

$$F_1(t) + k_{E,eq}x_2 = m_2\ddot{x} \quad (71)$$

The following step is illustration for specific displacement which helps identify equation of motion in terms of output displacement for link 2 as shown in Figure 83.

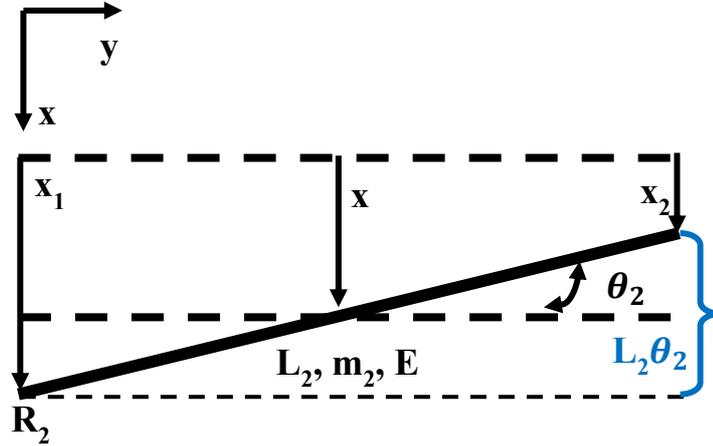


Figure 83: Link 2 specific displacement schematic

Based on the schematic shown in Figure 83 the formula for  $x$  and  $x_2$ , in Equations (72) and (73), were determined, respectively.

$$x = x_1 - \frac{1}{2}L_2\theta_2 \quad (72)$$

$$x_2 = x_1 - L_2\theta_2 \quad (73)$$

From Equations (68) and (71), the variables  $\theta_2$  and  $x$  can be written in terms of  $x_1$  and  $x_2$  as shown in Equations (74) and (75), respectively.

$$\theta_2 = \frac{1}{L_2}(x_1 - x_2) \quad (74)$$

$$x = \frac{1}{2}(x_1 + x_2) \quad (75)$$

Substituting  $\theta_2$  from Equation (74) into Equation (25) is presented in Equation (76).

$$\begin{aligned}
& k_{E,eq}L_2 \left( x_1 - L_2 \left( \frac{1}{L_2} (x_1 - x_2) \right) \right) \\
& = \frac{1}{3} m_2 L_2^2 \left( \frac{1}{L_2} (\ddot{x}_1 - \ddot{x}_2) \right)
\end{aligned} \tag{76}$$

Rearranging Equation (76) yields:

$$\ddot{x}_2 + \frac{3}{m_2} k_{E,eq} x_2 = \ddot{x}_1 \tag{77}$$

Substituting  $x$  from Equation (75) into Equation (71) yields:

$$F_1(t) + k_{E,eq}x_2 = \frac{m_2}{2} (\ddot{x}_1 + \ddot{x}_2) \tag{78}$$

By rearranging Equation (78) the equation of motion will be in terms of  $x_1$  and  $x_2$  as shown in Equation (79).

$$F_1(t) + k_{E,eq}x_2 - \frac{m_2}{2} \ddot{x}_2 = \frac{m_2}{2} \ddot{x}_1 \tag{79}$$

In order to eliminate  $x_1$ , Equation (34) can be used to substitute  $x_1$  into (36) as presented in Equation (80).

$$\begin{aligned}
& F_1(t) + k_{E,eq}x_2 - \frac{m_2}{2} \ddot{x}_2 \\
& = \frac{m_2}{2} \left( \ddot{x}_2 + \frac{3}{m_2} k_{E,eq}x_2 \right)
\end{aligned} \tag{80}$$

By rearranging Equation (80), the resultant equation of motion in terms of  $x_2(t)$  and  $F_1(t)$  is shown in Equation (81).

$$F_1(t) = m_2 \ddot{x}_2 + \frac{1}{2} k_{E,eq} x_2 \tag{81}$$

### Appendix C: Laplace Transformation

The next step is to conduct Laplace transformation for the resultant Equation (81) in order to determine the relation between the input force and output displacement as shown in Equations (82) and (83) and (84)

$$\mathcal{L}(F_1(t)) = m_2\mathcal{L}(\ddot{x}_2) + \frac{1}{2}k_{E,eq}\mathcal{L}(x_2) \quad (82)$$

$$F_1(s) = m_2(X_2(s)s^2 - sx_2(0) - \dot{x}_2(0)) + \frac{1}{2}k_{E,eq}X_2(s) \quad (83)$$

By zeroing all the initial and boundary conditions and rearranging Equation (83), the resultant Laplace transformation is presented in Equation (84).

$$\frac{X_2(s)}{F_1(s)} = \frac{2}{k_{E,eq} + 2m_2s^2} = G(s) \quad (84)$$

Figure 84 represents the block diagram for the corresponding Laplace transformation equation.

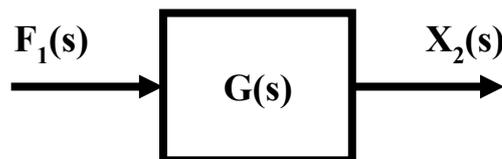


Figure 84: Laplace transformation block diagram

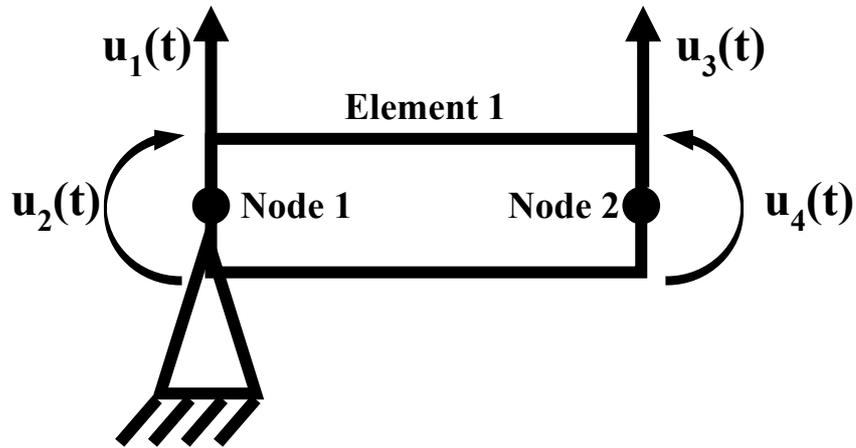


Figure 85: Element nodal displacements and constraint

Corresponding to **Error! Reference source not found.** node 1 is constrained then  $u_1 = 0$ , therefore the corresponding stiffness and mass matrices will be:

$$K_1 = \frac{EI}{l^3} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -6l & 2l^2 \\ 0 & -6l & 12 & -6l \\ 0 & 2l^2 & -6l & 4l^2 \end{bmatrix} \quad (85)$$

$$M_1 = \frac{\rho Al}{420} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4l^2 & 13l & -3l^2 \\ 0 & 13l & 156 & -22l \\ 0 & -3l^2 & -22l & 4l^2 \end{bmatrix} \quad (86)$$

Lagrange's equations,

$$\frac{du}{dt} \left( \frac{\partial T u}{\partial u_i} \right) - \frac{\partial T u}{\partial u_i} + \frac{\partial V u}{\partial u_i} = f_i(t)$$

As link1 and 2 are separated to six beam elements each the elements have similar stiffness and mass matrices as shown in Equations (87) and (88), respectively.

$$K = \frac{EI}{l^3} \begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \quad (87)$$

$$\mathbf{M} = \frac{\rho A l}{420} \begin{bmatrix} 156 & 22l & 54 & -13l \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ 6l & -3l^2 & -22l & 4l^2 \end{bmatrix} \quad (88)$$

$$\mathbf{K}_1 = \frac{EI}{l^3} \begin{bmatrix}
12 & 6l & -12 & 6l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6l & 4l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & -6l & 12 & -6l \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6l & 2l^2 & -6l & 4l^2
\end{bmatrix} \quad (89)$$

$$\mathbf{M}_1 = \frac{\rho Al}{420} \begin{bmatrix}
156 & 22l & 54 & -13l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22l & 4l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & 13l & 156 & -22l \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & -22l & 4l^2
\end{bmatrix} \quad (90)$$

In this link the local coordinated are considered coinciding with the global coordinates therefore there is no need for performing transformation matrix. Applying the boundary conditions  $u_1 = 0$  to Equations (89) and **Error! Reference source not found.** yields:

$$\mathbf{K}_1 = \frac{EI}{l^3} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & -6l & 12 & -6l \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \quad (91)$$

$$\mathbf{M}_1 = \frac{\rho A l}{420} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & 13l & 156 & -22l \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & -22l & 4l^2 \end{bmatrix} \quad (92)$$

The stiffness and mass matrices for the second link are presented in Equations (93) and (94), respectively.

$$\mathbf{K}_1 = \frac{EI}{l^3} \begin{bmatrix} 12 & 6l & -12 & 6l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6l & 4l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -12 & -6l & 24 & 0 & -12 & 6l & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6l & 2l^2 & 0 & 8l^2 & -6l & 2l^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12 & -6l & 12 & -6l & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6l & 2l^2 & -6l & 4l^2 & 0 \end{bmatrix} \quad (93)$$

$$\mathbf{M}_1 = \frac{\rho A l}{420} \begin{bmatrix}
156 & 22l & 54 & -13l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22l & 4l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & -6l & 312 & 0 & 54 & -13l \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & 0 & 8l^2 & 13l & -3l^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 54 & 13l & 156 & -22l \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -13l & -3l^2 & -22l & 4l^2
\end{bmatrix} \quad (94)$$

## Appendix D: MATLAB Codes

### Link\_1.m

```

n=1;
Mp=0;
Ih = 5.8598e-4;
l=0.9;
M=(2710*19.008e-3*3.2004e-3*1/420)*[140*1^2 63*1 14*1^2 147*1 -
21*1^2;
63*1 156 22*1 54 -13*1;
14*1^2 22*1 4*1^2 13*1 -3*1^2;

147*1 54 13*1 156 -22*1;
-21*1^2 -13*1 -3*1^2 -22*1 4*1^2];

K=(7.11e10*5.2530e-11/1^3)*[0 0 0 0 0;

0 12 6*1 -12 6*1;
0 6*1 4*1^2 -6*1 2*1^2;
0 -12 -6*1 12 -6*1;

0 6*1 2*1^2 -6*1 4*1^2];
D=[zeros(5)];
M(:,2:3)=[];
M(2:3,:)=[];
K(:,2:3)=[];
K(2:3,:)=[];
D(:,2:3)=[];
D(2:3,:)=[]
e = -(inv(M)*K);
g = -(inv(M)*D);
f=inv(M);
f(:,2:3)=[];
A1 = [zeros(3) eye(3); e g]
B1 = [zeros(3,1); f];
C1 = eye(6);
D1 = zeros(6,1);

```

### Link\_2.m

```

n=1;
Mp=0.000;
Ih = 5.8598e-4; %inertia of hub
l=1.1;
M=(2710*19.008e-3*3.2004e-3*1/420)*[140*1^2 63*1 14*1^2 147*1 -
21*1^2;
63*1 156 22*1 54 -13*1;
14*1^2 22*1 4*1^2 13*1 -3*1^2;
147*1 54 13*1 156 -22*1;
-21*1^2 -13*1 -3*1^2 -22*1 4*1^2];
K=(7.11e10*5.2530e-11/1^3)*[0 0 0 0 0;
0 12 6*1 -12 6*1;
0 6*1 4*1^2 -6*1 2*1^2;

```

```

0 -12 -6*1 12 -6*1;
0 6*1 2*1^2 -6*1 4*1^2];
D=[zeros(5)];
M(:,2:3)=[];
M(2:3,:)=[];
K(:,2:3)=[];
K(2:3,:)=[];
D(:,2:3)=[];
D(2:3,:)=[];
e = -(inv(M)*K);
g = -(inv(M)*D);

                                %For reducing the size

f=inv(M);
f(:,2:3)=[];
A2 = [zeros(3) eye(3); e g];
B2 = [zeros(3,1); f];
C2 = eye(6);
D2 = zeros(6,1);

```

### **PID.m**

```

p=0.9136
i=0.4278
d=0.3757
n= 88

```

### **lqr\_lqg\_simulink.m**

```

%State-space representation
n=1;
Mp=0.000;
Ih = 5.8598e-4;                                %inertia of hub
l=0.9;
M=(2710*19.008e-3*3.2004e-3*1/420)*[140*1^2 63*1 14*1^2 147*1 -
21*1^2;
63*1 156 22*1 54 -13*1;
14*1^2 22*1 4*1^2 13*1 -3*1^2;

147*1 54 13*1 156 -22*1;
-21*1^2 -13*1 -3*1^2 -22*1 4*1^2];
K=(7.11e10*5.2530e-11/1^3)*[0 0 0 0 0;
0 12 6*1 -12 6*1;
0 6*1 4*1^2 -6*1 2*1^2;

0 -12 -6*1 12 -6*1;
0 6*1 2*1^2 -6*1 4*1^2];
D=[zeros(5)];
M(:,2:3)=[];
M(2:3,:)=[];
K(:,2:3)=[];

```

```

K(2:3,:)=[];
D(:,2:3)=[];
D(2:3,:)=[];
e = -(inv(M)*K);
g = -(inv(M)*D);

                                % REDUCE SIZE

f=inv(M);
f(:,2:3)=[];
A1 = [zeros(3) eye(3); e g];
B1 = [zeros(3,1); f];
C1 = eye(6);
D1 = zeros(6,1);

sys=ss(A1,B1,C1,D1);

Q=2.3;
R=1;
Q1=C1'*C1;
R1=1;
K1=lqr(A1,B1,Q1,R1);
Vd=0.001*eye(6);
Vn=[0.001.*C1];
[L,P,E]=lqe(A1,Vd,C1,Vd,Vn);
Kf=(lqr(A1',C1',Vd,Vn))';
sysKF=ss(A1-Kf*C1,[B1 Kf],eye(6),0*[B1 Kf]);

%State-space representation for link 2
n=1;
Mp=0.000;
Ih = 5.8598e-4;                                %inertai of hub
l=1.1;
M=(2710*19.008e-3*3.2004e-3*1/420)*[140*1^2 63*1 14*1^2 147*1 -
21*1^2;
63*1 156 22*1 54 -13*1;
14*1^2 22*1 4*1^2 13*1 -3*1^2;
147*1 54 13*1 156 -22*1;
-21*1^2 -13*1 -3*1^2 -22*1 4*1^2];
K=(7.11e10*5.2530e-11/1^3)*[0 0 0 0 0;

0 12 6*1 -12 6*1;
0 6*1 4*1^2 -6*1 2*1^2;
0 -12 -6*1 12 -6*1;
0 6*1 2*1^2 -6*1 4*1^2];
D=[zeros(5)];
M(:,2:3)=[];
M(2:3,:)=[];
K(:,2:3)=[];
K(2:3,:)=[];
D(:,2:3)=[];
D(2:3,:)=[];

e = -(inv(M)*K);
g = -(inv(M)*D);
% REDUCE SIZE OF F(boundary conditions)
f=inv(M);
f(:,2:3)=[];
A2 = [zeros(3) eye(3); e g];

```

```

B2 = [zeros(3,1); f];
C2 = eye(6);
D2 = zeros(6,1);
Q2=C2'*C2;
R2=1;
K2=lqr(A2,B2,Q2,R2);
K2_1=[K2(1,3) K2(1,4) K2(1,5) K2(1,6)];
Vd1=0.001*eye(6);
Vn1=[0.001.*C2];

[L1,P1,E1]=lqe(A2,Vd1,C2,Vd1,Vn1);

Kf1=(lqr(A1',C1',Vd1,Vn1))';
sysKF1=ss(A2-Kf1*C2,[B2 Kf1],eye(6),0*[B2 Kf1]);

```

### **lqr\_autodesk.m**

```

%%
clear all;clc
m=1;
M=2;
L=2;
g=-5;
d=1;
s=1;

A=[0 1 0 0;
   0 -d/M -m*(-1000)/M 0;
   0 0 0 1;
   0 -s*d/(M*L) -s*(m+M)*(-1000)/(M*L) 0];
B=[0;1/M;1;s*1/(M*L)];
R=1000;

Q=[0.1 0 0 0;
   0 1 0 0;
   0 0 0.01 0;
   0 0 0 0.001];
K_lqr=lqr(A,B,Q,R)

```

### **lqg\_autodesk.m**

```

%%
clear all;clc
m=1;
M=3;
L=2;
d=1;
s=1;
A=[0 1 0 0;
   0 -d/M -m*(-10)/M 0;

```

```

    0 0 0 1;
    0 -s*d/(M*L) -s*(m+M)*(-10)/(M*L) 0];
B=[0;1/M;1;s*1/(M*L)];
C=[1 0 0 0];
D=zeros(size(C,1),size(B,2));
R=10000;
Q=[0.1 0 0 0;
    0 1 0 0;
    0 0 0.001 0;
    0 0 0 0.0001];
K_lqr=lqr(A,B,Q,R)
Vd=0.00001*eye(4);

Vn=0.00001;
[L1,P1,E1]=lqe(A,Vd,C,Vd,Vn);

Kf1=(lqr(A',C',Vd,Vn))';
sysKF2=ss(A-Kf1*C,[B Kf1],eye(4),0*[B Kf1]);

```

## x2DOF\_DataFile.m

```

% Simscape(TM) Multibody(TM) version: 7.0

% This is a model data file derived from a Simscape Multibody Import
XML file using the smimport function.
% The data in this file sets the block parameter values in an
imported Simscape Multibody model.
% For more information on this file, see the smimport function help
page in the Simscape Multibody documentation.
% You can modify numerical values, but avoid any other changes to
this file.
% Do not add code to this file. Do not edit the physical units shown
in comments.

%%VariableName:smiData

%===== RigidTransform =====%

%Initialize the RigidTransform structure array by filling in null
values.
smiData.RigidTransform(7).translation = [0.0 0.0 0.0];
smiData.RigidTransform(7).angle = 0.0;
smiData.RigidTransform(7).axis = [0.0 0.0 0.0];
smiData.RigidTransform(7).ID = '';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(1).translation = [0 0.018074041726955897
0.095592602081483125]; % m
smiData.RigidTransform(1).angle = 2.0943951023931953; % rad
smiData.RigidTransform(1).axis = [0.57735026918962584 -
0.57735026918962584 0.57735026918962584];
smiData.RigidTransform(1).ID = 'B[base-1--link-1]';

```

```

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(2).translation = [0.022898940437619815
0.0150000000000000154 -0.0079996867417752721]; % m
smiData.RigidTransform(2).angle = 0; % rad
smiData.RigidTransform(2).axis = [0 0 0];
smiData.RigidTransform(2).ID = 'F[base-1:-:link-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(3).translation = [-0.053159260620581392
0.015000000000000013 0]; % m
smiData.RigidTransform(3).angle = 3.1415926535897931; % rad
smiData.RigidTransform(3).axis = [1 0 0];
smiData.RigidTransform(3).ID = 'B[link-1:-:link-2]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(4).translation = [0.022898940437619319
0.0150000000000000336 -1.3877787807814457e-17]; % m
smiData.RigidTransform(4).angle = 0; % rad
smiData.RigidTransform(4).axis = [0 0 0];
smiData.RigidTransform(4).ID = 'F[link-1:-:link-2]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(5).translation = [-0.053159260620581364
0.015000000000000013 0.020000000000000004]; % m
smiData.RigidTransform(5).angle = 3.1415926535897931; % rad
smiData.RigidTransform(5).axis = [1 0 0];
smiData.RigidTransform(5).ID = 'B[link-2:-:end-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(6).translation = [2.4980018054066022e-16 -
4.891920202254596e-16 0.020000000000000004]; % m
smiData.RigidTransform(6).angle = 3.1415926535897931; % rad
smiData.RigidTransform(6).axis = [-1 -9.0499537651012407e-35
1.4585828000922471e-18];
smiData.RigidTransform(6).ID = 'F[link-2:-:end-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(7).translation = [0.11832125234749025
0.075282417722918687 -0.022662495524925649]; % m
smiData.RigidTransform(7).angle = 0; % rad
smiData.RigidTransform(7).axis = [0 0 0];
smiData.RigidTransform(7).ID = 'RootGround[base-1]';

%===== Solid =====%
%Center of Mass (CoM) %Moments of Inertia (MoI) %Product of Inertia
(PoI)

%Initialize the Solid structure array by filling in null values.
smiData.Solid(3).mass = 0.0;
smiData.Solid(3).CoM = [0.0 0.0 0.0];

```

```

smiData.Solid(3).MoI = [0.0 0.0 0.0];
smiData.Solid(3).PoI = [0.0 0.0 0.0];
smiData.Solid(3).color = [0.0 0.0 0.0];
smiData.Solid(3).opacity = 0.0;
smiData.Solid(3).ID = '';

%Inertia Type - Custom
%Visual Properties - Simple
smiData.Solid(1).mass = 0.45083208827597376; % kg
smiData.Solid(1).CoM = [-0.0034897986076934486 0.0039908395636401618
28.857856550698337]; % mm
smiData.Solid(1).MoI = [451.45071019612897 448.61038238181374
563.88515368759056]; % kg*mm^2
smiData.Solid(1).PoI = [-0.090456307625007976 0.082822578847644088 -
0.022079984286768654]; % kg*mm^2
smiData.Solid(1).color = [0.792156862745098 0.81960784313725488
0.9333333333333335];
smiData.Solid(1).opacity = 1;
smiData.Solid(1).ID = 'base*:Default';

%Inertia Type - Custom
%Visual Properties - Simple
smiData.Solid(2).mass = 0.0057685040127137856; % kg
smiData.Solid(2).CoM = [0 0 12.884376235820847]; % mm
smiData.Solid(2).MoI = [0.68121913141932888 0.68121913141932888
0.20865105739415105]; % kg*mm^2
smiData.Solid(2).PoI = [0 0 0]; % kg*mm^2
smiData.Solid(2).color = [0.792156862745098 0.81960784313725488
0.9333333333333335];
smiData.Solid(2).opacity = 1;
smiData.Solid(2).ID = 'end*:Default';

%Inertia Type - Custom
%Visual Properties - Simple
smiData.Solid(3).mass = 0.048545878885427822; % kg
smiData.Solid(3).CoM = [-15.475979147748843 15.000000000000005 10];
% mm
smiData.Solid(3).MoI = [5.6433388035302983 33.418002550163671
35.824949427998789]; % kg*mm^2
smiData.Solid(3).PoI = [0 0 0]; % kg*mm^2
smiData.Solid(3).color = [0.792156862745098 0.81960784313725488
0.9333333333333335];
smiData.Solid(3).opacity = 1;
smiData.Solid(3).ID = 'link*:Default';

%===== Joint =====%
%X Revolute Primitive (Rx) %Y Revolute Primitive (Ry) %Z Revolute
Primitive (Rz)
%X Prismatic Primitive (Px) %Y Prismatic Primitive (Py) %Z Prismatic
Primitive (Pz) %Spherical Primitive (S)
%Constant Velocity Primitive (CV) %Lead Screw Primitive (LS)
%Position Target (Pos)

%Initialize the RevoluteJoint structure array by filling in null
values.
smiData.RevoluteJoint(3).Rz.Pos = 0.0;
smiData.RevoluteJoint(3).ID = '';

```

```
smiData.RevoluteJoint(1).Rz.Pos = -173.49354624460116; % deg
smiData.RevoluteJoint(1).ID = '[base-1:-:link-1]';

smiData.RevoluteJoint(2).Rz.Pos = -133.93032546227619; % deg
smiData.RevoluteJoint(2).ID = '[link-1:-:link-2]';

smiData.RevoluteJoint(3).Rz.Pos = -50.436779217675031; % deg
smiData.RevoluteJoint(3).ID = '[link-2:-:end-1]';
```