POLITECNICO DI MILANO
DEPARTMENT OF MECHANICAL ENGINEERING
DOCTORAL PROGRAMME IN MECHANICAL ENGINEERING

DYNAMIC MODELING AND CONTROL OF FLEXIBLE MANIPULATORS

Doctoral Dissertation of:
Rajesh Kumar Moolam

Supervisor:
Prof. Francesco Braghin

Tutor:
Prof. Ferruccio Resta

The Chair of the Doctoral Program:
Prof. Bianca Maria Colosimo

2013 – XXVI

Abstract

LIGHT weight and flexible manipulators have many advantages over heavy and rigid manipulators such as lower energy consumption, higher payload to manipulator weight ratio, small actuators to move manipulator arm, and safe to operate with human coworker due to low inertia. These benefits come at the cost of flexibility in links or joints. The flexibility leads to vibrations on the endeffector.

The control objective in the case of flexible link manipulators is that of achieving desired endeffector trajectory tracking and suppressing vibrations of the endeffector. Control design of such systems needs accurate dynamic models to obtain the desired performance requirements. Many model based controllers were developed in the past for flexible link manipulators. But these studies were limited to planar flexible link manipulators.

In this thesis, a systematic approach for the dynamic modeling and control of spatial flexible link manipulators is presented. A general purpose code has been developed in MATLAB to derive the multi-body dynamic model of flexible manipulators for numerical simulation and control design. In the dynamic formulation, the principle of virtual work is used to derive the equations of motion in an absolute coordinate system for general purpose implementation. Then, the dynamic model in the absolute coordinates system is converted into independent coordinates form using a recursive kinematic formulation. The advantage of general purpose algorithm is it uses minimum set of equations that define the dynamics of flexible manipulator, which is required in control design to reduce computational effort. In addition, it allows the dynamic modeling of any arbitrary manipulator configuration that consists of rigid links, flexible links and flexible joints.

To control the spatial flexible link manipulator, model based controllers such as PD control, stable inversion control, nonlinear control, adaptive control are developed. The simulation results of spatial flexible link manipulator that consists of three flexible links and three revolute joints is presented to compare the performance of each controller in terms of trajectory tracking and vibration suppression.

The dynamic model validation and performance of model based controllers are experimentally tested on a single flexible link manipulator. Numerical and experimen-

tal results showed that adaptive control have better performance in terms of trajectory tracking and vibration suppression compared to PD control, Stable inversion control and nonlinear control in the presence of additional unknown payload mass.

Keywords: Dynamic modeling, spatial flexible manipulators, recursive kinematic formulation, PD control, stable inversion control, nonlinear control, adaptive control

Summary

ROBOT manipulators are designed to increase the productivity and to help humans in tedious and hazardous work environment. These manipulators are made of heavy and stiff materials to achieve high precision on endeffector motion. Heavy robot manipulators have higher mass to payload ratio, consume more power and have limited operation speed. To improve the performance of rigid robot manipulators, the focus on light weight and flexible manipulators has increased in recent years. The applications of light weight manipulators can be found in space, construction, and medical field.

The control of light weight manipulators is complex based on the nature of flexibility in the system, i.e. flexible joints, flexible links or flexible joints and links. Among them, the most difficult task is to control the flexible link manipulators because of link flexibility, under actuation and non-minimal phase nature. Under actuation is due to finite number of actuators to control infinite degrees of freedom that arise due to link flexibility. Non-minimum phase nature occurs because of non-collocation of actuators and sensors.

Despite of various advantages, the flexible link manipulators have less progress at the industrial level applications. There is a need to bring the advantages of flexible link manipulators to more general industrial applications by eliminating the difficulties surrounded to it such as modeling link flexibility, under actuation and non-minimum phase nature in control design. The dynamic modeling that includes the link or joint flexibility is considered as an important step in model based control design to achieve better performance. In the past, many model based controllers were developed for flexible links manipulators but these studies are limited to planar models.

The following research work is dedicated to identify and develop a systematic approach for the dynamic modeling that can accurately include the flexibility and to develop model based control for spatial flexible manipulators to achieve desired trajectory tracking.

In chapter [1], the introduction to the applications of flexible manipulators is presented. The state of the art on the dynamic modeling and control of flexible manipulators is discussed. Then, the objective of the research work is presented.

In chapter [2], the kinematic and dynamic formulation is presented for spatial flexible manipulators. The kinematics of flexible links is defined using floating reference formulation. The elastic deformations of flexible links are defined using the Euler Bernoulli beam formulation. Finite element method is used to discretize the flexible link to get finite dimensional dynamic models. The equations of motion of flexible manipulators are derived using the principle of virtual work in an absolute coordinate system. Then, the set of equations in absolute coordinates is converted into relative or independent coordinates using recursive kinematic formulations. A general purpose multi-body code structure developed based on the dynamic formulation is presented in this chapter. The simulation results of spatial RRR flexible manipulator is presented to study the effect of link and joint flexibility on the manipulator dynamics.

In chapter [3], model based controllers are designed for spatial flexible link manipulators. PD control, Stable inversion control, nonlinear control, and adaptive control are designed and implemented on a spatial RRR flexible link manipulator. The nonlinear control and adaptive control derived in this chapter considers the friction and damping effect at the manipulator joints. The performance of model based controllers are compared in terms of trajectory tracking and vibration suppression.

In chapter [4], the experimental setup of a single link flexible manipulator is presented. In the real system, the friction at manipulator joint plays an important role in position control. The coulomb and viscous friction compensator is designed based on set of experiments. The dynamic model is verified and performance of model based controllers are compared with additional unknown payload mass on the endeffector.

In chapter [5], the conclusions on the dynamic modeling and model based controllers for spatial flexible link manipulators are presented.

Contents

CHAPTER 1

Introduction

Robot manipulators are designed to increase the productivity and to help humans in tedious and hazardous work environment. The manipulator arms are made of heavy and stiff materials to achieve high precision on endeffector motion. However, the heavy manipulator arms are required to have bulky actuators for robot manipulation in workspace. In addition, the heavy manipulators have higher mass, consume more power and have limited operation speed with respect to operating payload. In order to built power efficient robot manipulators and to increase the operation speed, the focus is switched towards development of light weight manipulators [28].

The applications of lightweight manipulators can be found in space, construction, and medical field. In space applications, the space shuttle is equipped with a long reach manipulator to assist the astronauts in extra vehicular activities [44]. The Shuttle Remote Manipulation System (RMS), CANADARM shown in figure (1.1) is a 15.3 m long, 38 cm in diameter, and weighs 408 kg. It has six joints similar to that of human arm. The Shuttle RMS can handle payloads with masses up to 29,500 kg.

In construction applications, the truck mounted concrete boom structures shown in figure (1.2) is used to transport the concrete. These manipulators can have vertical movement up to 31.2 m, and horizontal movement up to 26.5 m.

Another attractive feature of light weight manipulators is safe to work along with human coworker and easy transportability. With these advantages, the use of light weight manipulators is emerging in the field of automotive, medical, and various general purpose industrial applications. The KUKA and DLR together developed such a light weight manipulator that, weighting 13.5 Kg, can handle 15 Kg load.

The benefits of light weight manipulators come at the cost of flexibility in joints, links or both. For instance, KUKA DLR lightweight manipulator showed in figure (1.3) is treated as flexible joint manipulator due to the harmonic drive transmission,

Figure 1.1: *Space shuttle remote manipulation system.*

differential gear drive and frictional effect [60] [2]. The shuttle RMS [17] and concrete boom structure [26] have a long slender links made of lightweight materials treated as a flexible link manipulator. The flexible links can lead to vibrations on the end-effector and complex dynamic behavior to the whole system.

The control of light weight manipulators is complex based on the nature of flexibility in the system, i.e. flexible joints, flexible links or flexible joints and links. Among them, the most difficult task is to control the flexible link manipulators because of link flexibility, under actuation and non-minimal phase nature. Under actuation is due to finite number of actuators to control infinite degrees of freedom that arise due to link flexibility [5]. Non-minimum phase nature occurs because of non-collocation of actuators and sensors [70].

Despite of various advantages, the flexible link manipulators have less the progress at the industrial level. There is a need to bring the advantages of flexible link manipulators to more general industrial applications by eliminating the difficulties surrounded to it such as modeling link flexibility, under actuation and non-minimum phase nature in control design. The dynamic modeling that includes the link or joint flexibility is considered as an important step in model based control design and to achieve better performance. The following research work is dedicated to identify and develop a systematic approach for the dynamic modeling and model based control of spatial flexible manipulators.

1.1 State of the art

The state of the art on dynamic modeling and control of flexible manipulators are discussed in this section. At first the various dynamic formulations and techniques for the modeling of flexible manipulators are presented. Then, the control techniques to achieve desired trajectory tracking and vibration suppression are presented.

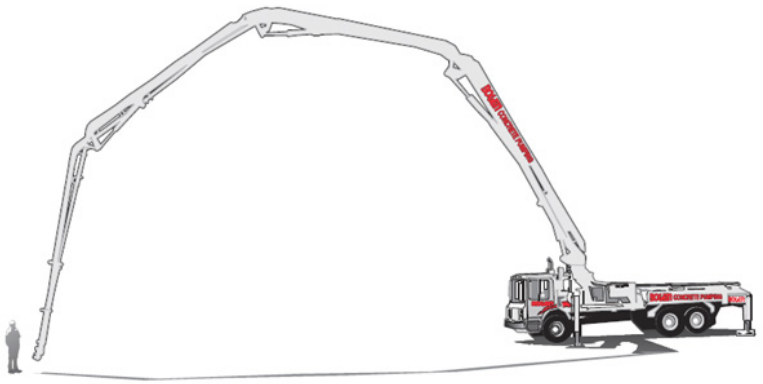

Figure 1.2: *Truck mounted concrete boom pump.*

Figure 1.3: *Kuka DLR Light weight robot.*

1.1.1 Dynamic modeling of flexible manipulators

Dynamic modeling is an important step in the control design process. The performance of the controller mainly depends on the accuracy of the dynamic models that are used in control design.

The dynamics of the robot manipulators can be derived using various methods such as Newton-Euler, principle of virtual work, Lagrangian-Euler, Gibbs-Appell, Hamilton principle. Many algorithms were developed based on these methods to define equations of motion.

The inclusion of the dynamics due to flexibility makes the robot manipulator a highly nonlinear model. It also establishes strong coupling between gross rigid body motion and elastic deformations of links or joints. The techniques to model flexibility in manipulator dynamics are presented below.

Flexible Joint Manipulators

Flexible joint manipulators are assumed to have rigid links and flexible joints. In [58] the joint flexibility was modeled as a linear spring and showed that as the stiffness goes to infinity the model behaves like a rigid manipulator.

In [53] the dynamic model of two revolute joint robot with flexible joint is derived. The flexible joint have servo stiffness and damping modeled along with stiffness and damping of drive system. A series of simple torsional spring is used to model servo stiffness and drive stiffness. The resulting torque stiffness equation at joints along with equations of motion of rigid links is a highly coupled nonlinear ordinary differential equations. From the numerical investigation, it is concluded that the servo damping plays an important role in dynamic response of the system.

A generic dynamic model was developed in [31] for an industrial KUKA IR 761 robot manipulator which includes joint elasticity due to electric drives. In [39], the dynamics of a n-link flexible manipulator was modeled with revolute flexible joints described as a linear torsional spring with known stiffness.

The DLR medical robot in [43] is a redundant robot with 4 flexible joints. It considers the elasticity of joint due to harmonic drive and differential gear transmission. The elastic joint was modeled as a linear torsional spring. The resulting dynamic model has a strong coupling between the flexible joint and link motion.

Flexible Link Manipulator

Flexible link manipulators are assumed to have rigid joints and flexible links. The links are considered to have low stiffness because of lightweight materials. The dynamics of link flexibility can be modeled using Euler-Bernoulli beam equation or Timoshenko beam equation.

The dynamics of beam deformations can be described using partial differential equation. With the inclusion of dynamics due to link flexibility, the dynamic model takes the form of coupled nonlinear ordinary and partial differential equation. The flexible link manipulator system has distributed dynamic parameters which are characterized by infinite number of degrees of freedom. The exact solution of such systems is not feasible. To simplify the modeling process, the continuous systems can be discretized by using assumed mode method (AMM), finite element method (FEM), finite difference method (FDM) and Lumped parameter method (LPM).

Assumed Mode Method

In AMM, the flexibility of link is truncated to obtain finite number of modes that can accurately represent the dynamic parameters of the flexible link. This method assumes the deflection of link is small and expresses the deflection as summation of finite number modes. Each mode is defined as product of two functions, first as a function of distance along manipulator length and other function as generalized coordinates depending on time.

In [73], the dynamic model was developed using recursive Lagrangian approach via transformation matrix and modeled the link flexibility using AMM. The transformation matrix is a 4x4 matrix that includes the joint flexibility. This method is an extension to classical rigid body transformation matrix. The advantage of this approach is that it

represents both joint and link deflection motion in the form of transformation matrix and can be easily used in algorithm implementation.

A closed form dynamic modeling approach for a planar multi-link manipulators were presented in [4]. A Lagrangian approach and AMM was used to model the dynamics of planar multi-link flexible manipulator.

In [72], a linearized dynamic model for a multi-link planar flexible manipulator was derived using Euler-Lagrangian formulation and AMM. The dynamic model is linearized around rigid body motion to decouple the rigid and elastic deformation of manipulator.

The truncated modes of flexible link to approximate the dynamic behavior of system are not the same when the manipulator has a payload mass on the endeffector. The effect of endeffector payload mass on the modes of flexible link was studied in [52] [50]. A two link planar flexible manipulator was considered to study the dynamics of system with additional payload mass on endeffector. In those studies, a closed form dynamic equation using Euler-Lagrangian formulation and AMM was derived to show the effect of payload on dynamic behavior of the system.

There were number of research studies on AMM because of the simplicity in dynamic formulation. However, the main drawback of this method is that it is difficult to find modes for non-regular cross-sections. And the choice of boundary conditions for multilink manipulator is not unique. The possible boundary conditions reported in the literature are clamped, pinned-pinned, free-free boundary conditions. In [22], it was reported that if the beam to hub inertia ratio is small, the clamped boundary condition yields better results than pinned boundary conditions.

Finite Element Method

Using Finite element method, the drawbacks of AMM such as defining boundary conditions and irregular geometry can be accounted in a straightforward way. In this method, the continuous flexible links are divided in number of small elements. The displacements at any point on the link are expressed in terms of nodal displacements and polynomial interpolation function defined in element. The dynamics of each finite element is derived first then the elements are assembled based on element connectivity to obtain dynamics of the whole system.

The comparison between two discretization techniques such as AMM and FEM was studied in [64] to efficiently define link flexibility in manipulator dynamics. Lagrangian equation was used to derive closed form dynamic equations of motions. The numerical results were presented for a flexible spherical (RRP) manipulator and stated that FEM overestimates the structural stiffness but it takes fewer operations to compute inertia matrix.

A general purpose computer program SPACAR for numerical simulations of flexible mechanism and manipulators was developed in [33]. SPACAR used a finite element based Lagrangian formulation to define dynamics of the system. The program was incorporated with virtual power type approach, to automatically eliminate the non-working constraints forces and reactions. This approach leads the Lagrangian formulation into a minimal set of ordinary differential equations.

In [57] the numerical and experimental investigation on dynamic modeling of flexible manipulators was done. A planar constrained model was considered for dynamic

model verification. The dynamic model was developed using Lagrangian approach and finite element method. The experimental validation was carried out on a single link flexible manipulator and compared with numerical finite element model in both frequency and time domain. The FEM model showed closer agreement with the experimental results.

A redundant Lagrangian FEM formulation for the dynamic modeling of flexible links and joints was presented in [47]. The elastic deformation on each link is assumed due to bending and torsion. The deformation of each link is expressed in tangential local floating frame. The constrained equations due to connectivity of each link are added to equations of motion by using Lagrangian multiplier. Other works on dynamic modeling of spatial flexible manipulator based on Lagrangian formulation and FEM were addressed in [67] [34].

The advantage of FEM method is that it can consider complex geometric shapes and have no difficulties in defining the boundary conditions that exist in AMM. Other important thing is that it can make use of existing and well defined FEM algorithms for dynamic modeling of flexible manipulators to accurately define rigid and flexible dynamics.

Finite Difference Method

Finite difference method is also used to approximate the dynamic characteristics of flexible manipulators. This method discretizes the system into several segments and develops a linear relation for deflection of the end of each segments using finite difference approximation.

An algorithm was proposed for a real time simulation of flexible manipulator using FDM in [55]. The performance of the algorithm was tested on a single link flexible incorporating hub inertia and payload.

In [56] the study on FEM and FDM was done to investigate the dynamic characteristics of the flexible manipulator system based on the accuracy, computational efficiency and computational requirement. Both methods were compared on a single link flexible manipulator and the results demonstrated that FEM representation was more accurate and more efficient performance can be achieved compared to FDM.

Lumped Parameter Method

Lumped Parameter method is a simple method to model the dynamics of flexible link. This method basically defines the continuous flexible link as lumped masses and massless springs. There are two approaches to define the characteristics of concentrated mass and springs. One of them is a numerical approach that uses finite element method or Holzer method [15]. The other one is an experimental approach that uses a series of experiments to define parameters of a flexible link.

In [68] the finite element method associated with model analysis was used to model distributed system into lumped parameter model. The lumped parameter model was defined as a cascade system of concentrated point masses and weightless linear and angular spring.

In Holzer method [20] the flexible link was partitioned in to small divisions. The total mass of each division is treated as two equal concentrated masses at the ends of division. The lumped mass on the division points is connected by an elastomer without

mass. The lumped mass is also called as station and elastomer between lumped mass is called field.

In [69] the lumped parameter model of flexible link was defined using equivalent representation of virtual rigid link and passive joints such as springs and dampers. The parameters of virtual rigid link and passive joints were identified by using measured data of real flexible link.

The advantage of lumped parameter method is that it does not involve complex mode shape functions compared to finite element method and assumed mode method. However, it cannot be used if the flexible link manipulator has complex geometric shapes.

1.1.2 Control of flexible manipulators

The primary objective of control for a flexible link manipulator is to compute input torque necessary to drive the system in a desired trajectory and minimizing the oscillation of the tip. The dynamic model of flexible manipulator is a nonlinear and coupled differential equation. As stated in previous section, strong coupling exists between the rigid body motion and the elastic modes. In control design, there should be a compromise between rigid and elastic modes to achieve desired performance.

Many model based controllers were developed for flexible manipulators to achieve desired trajectory tracking and vibration suppression. These are designed based on the control schemes available for linear and nonlinear dynamic systems. Some of them used a linearized dynamic model to simplify control design process. Moreover, the linear control techniques to analyze system stability and robustness properties are very well established.

The following nonlinear control techniques are widely used to design the controller for flexible manipulator, i.e open loop method, feedback linearization method, singular perturbation method, stable inversion method, Robust control, Adaptive control, sliding mode technique, pole placement method, output redefinition, lead-lag control, iterative learning method, and intelligent control methods such as Fuzzy Logic control, Neural Network control.

Open loop method

In open loop control, the input signal is computed based on reference trajectory and vibrations are controlled by modifying the input signal considering the physical and vibrational properties of flexible link manipulators. The open loop control methods developed for flexible link manipulators are command shaping method and computed torque method based on model inversion.

A command shaping method for a single link flexible manipulator was proposed in [75] [77]. This method computes the input signal required for a rest to rest motion in a finite time and reduces the vibration of flexible link. The advantage of this method is that it computes the input signal to suppress vibrations without measurement data by solving a set of linear equation.

In [51] a feedforward control based on input shaping technique was proposed to suppress vibrations. In the experiments, an unshaped bang-bang torque was used to determine the dynamic parameters of the system. The parameters are then used in input shaping technique to suppress vibrations.

In computed torque method, the input torque is computed using model inversion based of the desired output trajectory. In [66] a feedforward control was developed based on inverse dynamic model. The dynamic model was derived in the form of differential algebraic equations. In [1], an energy saving open loop control scheme was proposed for a single link flexible manipulator to perform point to point motion in a fixed time. For a given tip trajectory, the joint angle was computed using Artificial Neural Network, and vector evaluated particle swarm optimization was used as a learning algorithm. The vibration suppression was realized along the joint trajectory using minimum driving energy consumption. This method has an advantage of energy saving because the residual vibrations are suppressed without measuring vibrations.

The open loop control methods are simple and require less measurement but they are very sensitive to the model inaccuracy, system parameter variation, and uncertainty in the implementation of desired trajectory. To overcome these issues, a feedback is necessary to monitor the system behavior to improve the performance of the controller and ensure stability of the system.

Feedback linearization Method

The first published and known work on feedback control of flexible manipulator was presented in [74]. The theoretical studies of PD feedback control of two-link two joint robot manipulator was presented. The joint position and velocity errors are used in feedback loop with constant gains. The gains were calculated based on vibrational properties to damp out the vibrations. The stability of the closed loop control was analyzed using root-locus method.

In [54] studied extensively on control of flexible arms and influence of unmodeled dynamics in the system controllability and performance which is widely known as spillover. After successful studies on different control challenges, there was a need to develop and test these techniques on real platform. The first known flexible arm of single link was developed at Aerospace robotics lab in Stanford University. In [63] experimental studies on precise positioning of flexible arm by sensing the tip and arm joint position are carried out. In this work, the concepts of non-collocated system robot and non-minimum phase nature were addressed.

In [19] a model based controller was proposed using approximate inverse dynamics and passive feedback control for a six degrees of freedom manipulator with flexible links. In this approach, the non-minimum phase nature of the system is solved by introducing μ-tip rate which is output of the system and passive to input torque. The actual tip position is approximated by μ-tip position using joint angles and elastic coordinates by a parameter μ. The global asymptotic stability was proved using Lyapuonv function for a PD feedback control. The advantage of this approach is that it does not require measurement of elastic coordinates; the controller provides stable tracking by just using joint angles and rates plus endeffector position and rates.

The collocated and noncollocated PD control for a single link manipulator was proposed in [38]. Minimum phase system is achieved by exact transfer function which has joint torque as input and joint angle plus weighted value of tip deflection as an output. The non-minimum phase nature occurs when the tip position is considered as an output. In this approach, the output of the link i.e. tip position is defined as joint angle plus weighted value of tip deflection. The conditions for weighted value of tip is defined

using infinite product expansion, root locus method such that the transfer function does not have any open right half plane zeros.

Singular Perturbation Method

A Singular perturbation approach can be applied to control rigid body motion and stabilize vibration along the trajectory. A well know work of trajectory control using singular perturbation approach is in [16]. Using this approach, the system dynamics are divided into slow and fast dynamics and a two stage controller was developed to tackle system vibrations and trajectory tracking. Slow dynamics are mainly to control the rigid body motion along the predefined trajectory along joint space. Fast dynamics are responsible for stability of vibrations along the joint trajectory. In this approach, the spill over problem in the fast dynamics is bounded by a perturbation parameter. Later on, this approach is applied to design many controllers. Fast dynamic control and slow dynamics control are controlled separately and many improvements are made on each control to achieve desired performance.

A robust-optimal controller was proposed in [71] for a single link flexible manipulator using singular perturbation approach. Slow dynamics (rigid modes) are controlled using sliding mode method and optimal LQR is designed to stabilize the fast dynamics (flexible modes). Angular position error is used to control rigid body dynamics, tip deflections are measured using strain gauge. The proposed controller tested on a real platform and the results showed good stability along trajectory.

A composite controller was developed in [6] for a three link spatial flexible manipulator using singular perturbation method. The composite controller is based on Cartesian based PI control for endeffector trajectory tracking and pole placement feedback control to damp the vibrations along trajectory. The advantage of this method is it does not require derivative of the endeffector position, and derivatives of strain gauge in the feedback loop. The proposed controller was tested on a real platform FLEBOT II.

Stable Inversion Method

Stable Inversion Method is another method widely used to design trajectory control of flexible link manipulators. In this method, the dynamic model is inverted in the form of Input-Output terms. In dynamic model inversion, the system has a non-minimum phase nature when the endeffector is considered as output. There are several approaches to make the model inversion stable in control design for trajectory tracking.

In [9] a joint based inversion approach was used to overcome the non-minimum phase nature. In this case, the output of the system is considered as joint coordinates. In this form, the model inversion has acceleration as an input and torque as an output. To compute the torque corresponding to the desired endeffector trajectory with minimum vibrations, both joint and elastic states are required. The elastic states for the reference joint based trajectory should be computed first by solving the internal dynamics off-line. In addition to computed torque for the desired reference trajectory, the system has a PD feedback loop for a joint trajectory tracking to make control robust. The proposed stable inversion control is robust at joint space and the vibrations are suppressed by solving the internal dynamic off-line. However, it does not consider the endeffector in the control loop, so the stability of tip vibrations is not guaranteed always.

Later on, the output of the system i.e. tip position is defined as joint coordinates plus weighted value of tip deflection [12] for accurate endeffector trajectory tracking. As usual, the model inversion has unobservable internal dynamics i.e. elastic states for the reference trajectory. Three methods were proposed to solve internal dynamics i.e. approximate non-linear regulation, Iterative inversion in frequency domain, Iterative learning in time domain [8] [13] [11]. Experimental results were presented on a two link flexible manipulator FLEXARM to show the robustness of the proposed stable inversion method. The disadvantage of these approaches is that the internal dynamics should be solved first off-line.

The stable Inversion method in time domain and frequency domain was used in [21] [23]. In [23], a model inversion method was proposed for endeffector tracking of flexible manipulator. It takes end point acceleration as input and computes the torque in frequency domain. The proposed model inversion is non-causal, because the output, i.e. torque, must begin before the input, i.e. where the end point acceleration begins. This approach showed good results for end point trajectory tracking but have the disadvantage of excessive computation burden due to dynamic model; input trajectory should be transformed from time domain to frequency domain and output should be transformed back to time domain. Later on, a convolution integration method [24] was used to reduce this computational burden.

In [21], dynamic model inversion was proposed in time domain to reduce the computation effort. The inverse dynamic model is treated as causal part and anti-casual part to compute bounded torque for endpoint trajectory. This model can produce the accurate endpoint trajectory but shows some position error due to present of friction at joint. A simple PD feedback control loop is introduced to compensate the frictional effects.

Recently, in [62] a stable inversion method for two link flexible manipulator is proposed. The dynamic system is converted into the input-output form. The output of the system is considered as joint angles plus weighted value of the tip deflection. To compute the bounded torque required for given reference trajectory, bounded elastic states are computed first by solving internal dynamics as a two sided boundary problem using MATLAB bvp5c solver. In addition, a design optimization was proposed for choosing weights of tip deflection. Some of the other works on stable inversion method for flexible manipulators can be referred in [78], [45], [10], [46], and [46].

The stable inversion method proposed in the literature for a flexible link manipulator assumes the dynamic model is accurate such that zero and poles of the system can be located accurately. It does not consider the model uncertainty; any model parameter such as mass of the link, stiffness, and unknown payload mass on the endeffector can alter the location of poles and zeros of the system. In this case, stability of the system is no longer assured using stable inversion method.

Robust control

The robust controllers were proposed for accurate trajectory tracking of flexible manipulators in the presence of unknown payload, and parameter uncertainty such as stiffness and joint friction. In [35], a robust control design was proposed for a single link flexible manipulators. The model uncertainty was considered due to stiffness, damping, and payload mass. Uncertainty of payload in left-hand side of inertia matrix and un-

certainty parameters included in right-hand side stiffness and damping are treated with polytopic and descaling techniques respectively. Then robust controller design problem is solved using linear matrix inequalitys (LMI).

In [3], a robust pole placement control was proposed using dynamic output feedback. The robustness of the controller is analyzed using LMI techniques. In [65] addressed the unstable closed loop response due to over-estimation of natural frequency from dynamic model derived from finite element method. A robustness of the controller is analyzed using second method Lyapunov function for a bounded model uncertainty. The proposed robust controller is a two stage controller, first stage controller is responsible for stable joint trajectory tracking and second stage controller is to suppress vibrations of the tip.

In [49] and [14] a robust controllers for flexible manipulators is proposed and the robustness of the system using μ-synthesis is analyzed. In [49] designed a collocated and noncollocated controller, and considered the model uncertainties are due to unmolded high frequency dynamics, error in natural frequencies, damping levels, actuators and sensors.

Robust controller are designed, analyzed and proved to be stable in the presence of bounded model uncertainty. When model uncertainties are large and cannot estimate the bounds of uncertainty, then robust controllers may not guarantee the stability. To overcome these difficulties and to make controller stable in unknown parameter uncertainty, adaptive controllers were proposed for flexible manipulators.

Adaptive control

Adaptive control for flexible link manipulators is designed mainly for accurate trajectory tracking and stabilizes tip vibrations for large unknown payload mass. The nonlinear dynamic system of flexible manipulator can be expressed in the fixed parameter form, which is the product of regression term and an unknown constant parameter.

An adaptive controller was proposed in [18] for a flexible link manipulator carrying payload much greater than manipulator mass. The input-output mapping of flexible manipulator was defined using μ-tip rate similar to [19] and a passive approach to solve non-minimum phase nature was applied. The passive nature of the output and fixed parameter form of system dynamics allows us to easily extend the rigid body adaptive controllers to flexible arms. The controller has feedforward term which is adaptive form and strictly passive PD feedback law. The adaptive controller ensures global stable tracking of cartesian endeffector coordinates with the help of only tip position and rate error.

The extended version of adaptive controller was proposed in [25]; a joint space feedforward term is derived for multilink case where task space dynamics is non-trivial. The proposed task space adaptive control and joint space adaptive control was experimentally verified on a three link planar manipulator that has one rigid link and two flexible links.

In [36] an adaptive variable structure was proposed for flexible link manipulators. The dynamics of flexible link manipulator was converted from the flexible mode dependent dynamics to strain dependent one, to reduce online computation burden on feedback signal of the flexible modes. The nonlinear dynamics of the flexible manipulator was expressed as a linear-in-parameter form for the feedforward term. The

stability of the systems was analyzed using Lyapunov function. The advantage of this approach is that it has less computational effort and just uses strain measurement in the feedback loop instead of complete elastic mode information. But, the adaptive variable structure scheme comes with chattering effect along the trajectory. To overcome this effect a saturation type adaptive law was proposed.

A nonlinear controller law both in non-adaptive and adaptive version was compared numerically and experimentally on a two link flexible manipulator [37]. The controller ensures accurate joint space trajectory tracking using joint position and velocity measurement and vibration suppression using strain measurement. The stability of the controller was analyzed using Lyapunov analysis. To avoid spillover problem, a second order analog filter in strain gauge amplifier and first order analog filter in strain gauge measurement was employed with a proper cutoff frequency to ensure closed loop stability. In [61] an adaptive control using sliding mode technique similar to [37] for a single link flexible manipulator is designed.

Intelligent control Methods

Soft computing techniques such as Fuzzy logic (FL), Neural Networks (NN), Reinforcement learning (RL) are also used in the control of flexible manipulators. The control with some intelligence is sometimes required to tackle unknown environment and to improve the whole performance of the system. These techniques are employed to choose suitable gains in feedforward compensation, feedback loop or both.

A vibration control using neural network was proposed in [48] for planar multilink flexible structures. In [29] a NN was developed to incrementally change the joint trajectory to achieve tip trajectory tracking in operational space. The tip position error was utilized as inputs to NN. The joint trajectory from NN was then controlled by PD control in joint space.

A cascade Fuzzy logic control (FLC) was proposed in [32] for a single link flexible manipulator. The control has a 3 FLCs, first fuzzy logic control has the input of joint position and derivate error, second FLC has tip position and derivate error. The first and second FLC outputs are the inputs to third FLC.

A Fuzzy-Neuro control was developed in [41] [42] for a planar two link rigid-flexible manipulator. Fuzzy logic control was used in feedback loop and dynamic recurrent neural network (DRNN) was used in feedforward term. The parameter output from FLC is used to update the adaptive DRNN model. The controller was applied for first three vibration modes of flexible arm.

In [30] a composite controller was developed using singular perturbation approach. The rigid body modes (slow dynamics) were controlled using NN and flexible modes (fast dynamics) were controlled using PD control. The proposed controller was compared with classical PD (slow dynamics)-PID (fast dynamics) control with additional frictional term. The composite controller showed good tracking in the presence of friction compared to PD-PID control. In [40] a fuzzy logic control using singular perturbation approach is proposed. Slow dynamics are controlled using fuzzy logic for joint trajectory tracking and fast dynamics are controlled using optimal LQR control to suppress vibrations.

In [59] a real time adaptive control was developed for a flexible manipulator carrying variable payload mass using reinforcement learning (RL). The performance of nonlin-

ear adaptive control (NAC), fuzzy-based adaptive control (FBAC), reinforcement based adaptive control was compared for different payload mass. First nonlinear adaptive control was designed using linear-in parameter form similar to [37] with some arbitrary gains. These gains can produce stable trajectory tracking with the certain parameter range. To improve the performance of controller NAC for varying payload, a FL was used to choose appropriate gains with in parameter range. For FBAC, a prior knowledge on parameter range is required to assign suitable gains. To overcome this, RL based adaptive control was design. Actor-critic based RL adopts actor-critic weights according to payload variation. The experimental results showed that RL based adaptive control has good performance over FBAC and NAC for payload variation.

1.2 Research Objective

Many model based controllers were developed in the past for trajectory tracking and vibrations suppression. However, these are limited to numerical or experimental studies on planar flexible link manipulators. The current research will be focused to develop a systematic approach for the dynamic modeling and model based control design of spatial flexible manipulators.

The research activity on flexible manipulators is divided into two parts. The first part will be focused on dynamic modelling of spatial flexible manipulators while the second part will be focused on control design for trajectory tracking and vibration suppression.

1.2.1 Dynamic modelling of flexible manipulators

The contribution of this thesis will be to develop a general purpose multi-body code that can accurately model the link and joint flexibility in the manipulator dynamics. The dynamic formulation will be developed based on the principle of virtual work and recursive kinematic formulation. The deformation of each link is assumed to be due to both bending and torsion. The deformation of the joints is assumed to be due to pure torsion. The flexible links are discretized using finite element method to get finite dimensional dynamic model.

A spatial flexible manipulator will be considered to study the effect of link and joint flexibility on manipulator dynamics. The following cases are considered for the study:

1. Rigid links and Rigid joints
2. Flexible links and Rigid joints
3. Flexible links and Fleixble joints

1.2.2 Control modelling of flexible manipulators

Many model based controllers were proposed for flexible manipulators for trajectory tracking applications. These are designed based on the control schemes available for linear and nonlinear dynamic systems. Some of them used a linearized dynamic model to simplify control design process and the linear control techniques to analyze system stability and robustness properties. The following control schemes are widely used to design control for flexible manipulator, i.e PID control, singular perturbation method,

stable inversion method, robust control, adaptive control, pole placement method, output redefinition, lead-lag control, iterative learning method, sliding mode control, and intelligent controls such as Fuzzy Logic control and Neural Network control. However, these are limited to numerical or experimental studies on planar flexible link manipulators.

The contribution of this thesis is to study the advantages and disadvantages of well developed controllers for planar flexible link manipulators and these methods are extended and improved to spatial flexible link manipulators. The following controllers will be developed for spatial flexible link manipulator.

1. PD Control
2. Stable Inversion Control
3. A Nonlinear Control
4. Adaptive Control

CHAPTER 2

Dynamic modeling

In this chapter a systematic approach for the dynamic modeling of flexible manipulator is presented. Rigid links, flexible links, and flexible joints are considered in the dynamic formulation. The kinematics of flexible links are derived using the floating reference formulation. The flexible links are deformable due to bending and torsion. The elastic deformations of flexible link due to bending is defined using the Euler-Bernoulli beam formulation. The inclusion of the dynamics due to link flexibility makes the robot manipulator a continuous system and require infinite degrees of freedom to estimate the dynamic parameters. It is not feasible to include infinite DOF in the dynamic model from the numerical simulations and control design point of view. Thus, the finite element method is used to discretize the flexible link to get the finite dimensional dynamic model.

The equations of motion are derived using the principle of virtual work in an absolute coordinate system for the general purpose implementation. Then, the set of equations in absolute coordinates is converted into relative or independent coordinates using the recursive kinematic formulations. The dynamic model that are derived using the principle of virtual work and finite element method considers the coupling effect of rigid body motion and elastic deformations of flexible link.

A general purpose multi-body code has been developed based on systematic approach that is presented for dynamic formulation of flexible manipulators. The input to the multi-body code are physical parameters of the flexible manipulator and the output is finite dimensional dynamic model of flexible manipulator. The multi-body code can be used for numerical simulation and control design purpose.

2.1 Kinematic Description

In this section, the kinematic equations that describe the position, velocity and acceleration of an arbitrary point is presented. A set of coordinate systems i.e. global coordinate system, body-fixed coordinate system, and floating coordinate system are used to derive the kinematic equations. The body-fixed coordinate system is used to define the translation and rotation of rigid link, where as a floating coordinate system is used to define the translation and rotation of flexible link. The general displacements of a point is described using Chasles theorem. It defines an arbitrary displacement as a sum of the translation of a point and a rotation along the axis of rotation.

According to Chasles theorem [7], the arbitrary displacement of a point can be defined as

$$r = R + A\overline{u} \tag{2.1}$$

where $r = [r_x\ r_y\ r_z]^T$ is the global position vector of an arbitrary point, $R = [R_x\ R_y\ R_z]^T$ is the position vector of the body coordinate system. A is the coordinate transformation matrix, and $\overline{u} = [\overline{u}_x\ \overline{u}_y\ \overline{u}_z]$ is the local position vector defined with respect to the body coordinate system.

The transformation matrix A is defined as

$$A = \begin{bmatrix} 2(\beta_0^2 + \beta_1^2) - 1 & 2(\beta_1\beta_2 - \beta_0\beta_2) & 2(\beta_1\beta_3 + \beta_0\beta_2) \\ 2(\beta_1\beta_2 + \beta_0\beta_2) & 2(\beta_0^2 + \beta_2^2) - 1 & 2(\beta_2\beta_3 - \beta_0\beta_1) \\ 2(\beta_1\beta_3 - \beta_0\beta_2) & 2(\beta_2\beta_3 + \beta_0\beta_1) & 2(\beta_0^2 + \beta_3^2) - 1 \end{bmatrix} \tag{2.2}$$

where β_0, β_1, β_2, and β_3 are the Euler parameters. These quantities are defined as

$$\beta_0 = \cos(\frac{\beta}{2}) \tag{2.3}$$

$$\beta_1 = \upsilon_1 \sin(\frac{\beta}{2}) \tag{2.4}$$

$$\beta_2 = \upsilon_2 \sin(\frac{\beta}{2}) \tag{2.5}$$

$$\beta_3 = \upsilon_3 \sin(\frac{\beta}{2}) \tag{2.6}$$

in which υ_1, υ_1, and υ_1 are components of the unit vector υ along the axis of rotation. β is the angle of rotation.

Using Chasles theorem, the kinematic equations for the position, velocity and acceleration of rigid and flexible links are derived.

2.1.1 Kinematics of Rigid link

The representation of an arbitrary point $\overline{O}_i$ on the rigid link *i* is shown in Figure [2.1]. The kinematic equation that defines an arbitrary displacement of a rigid link *i* is derived using the body-fixed coordinate system. The body coordinate system $X_iY_iZ_i$ is attached to the rigid link *i* to identify the position and orientation of the rigid link *i* in space. The configuration of each point on the rigid link *i* in space can be described

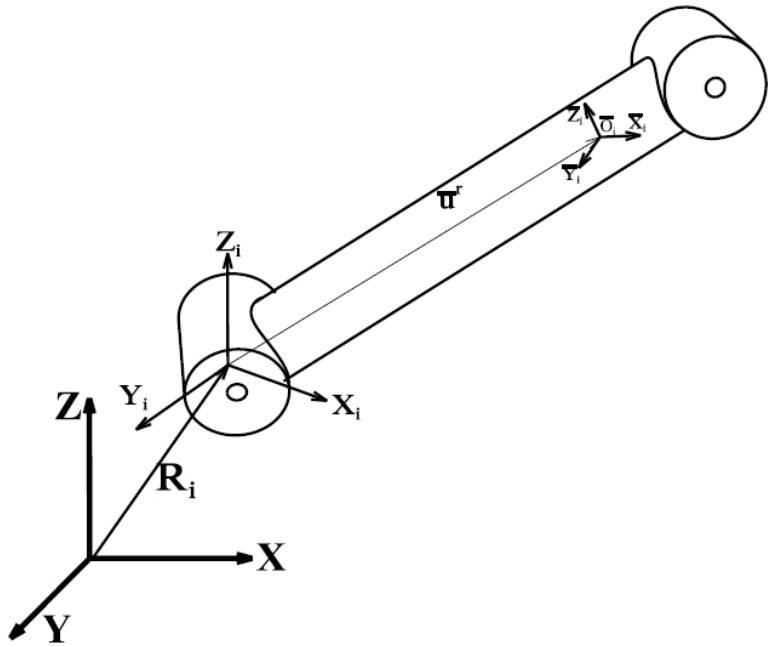

Figure 2.1: *Representation of an arbitrary point on a rigid link.*

using the position and orientation of the body coordinate system $X_iY_iZ_i$ that are defined with respect to the global coordinate system XYZ.

The position vector of an arbitrary point on the rigid link *i* is defined as

$$r_i = R_i + A_i\overline{u}_i \tag{2.7}$$

where $R_i = [R_x^i \; R_y^i \; R_z^i]^T$ is the position vector of the body coordinate system $X_iY_iZ_i$. A_i is the transformation matrix defined using equation [2.2], and $\overline{u}_i = [\overline{u}_x^i \; \overline{u}_y^i \; \overline{u}_z^i]^T$ is the local position vector defined with respect to $X_iY_iZ_i$. For the rigid link, the local position vector $\overline{u}_i$ is a constant vector.

The velocity vector of an arbitrary point on the rigid link *i* is obtained by differentiating equation [2.7]. It is written as

$$\dot{r}_i = \dot{R}_i + A_i(\overline{\omega}_i \times \overline{u}_i) \tag{2.8}$$

where $\overline{\omega}_i$ is the angular velocity vector defined in the body coordinate system $X_iY_iZ_i$. It is expressed as

$$\overline{\omega}_i = \overline{G}_i\dot{\beta}_i \tag{2.9}$$

where

$$\overline{G}_i = 2 \times \begin{bmatrix} -\beta_1 & \beta_0 & \beta_3 & -\beta_2 \\ -\beta_2 & -\beta_3 & \beta_0 & \beta_1 \\ -\beta_3 & \beta_2 & -\beta_1 & \beta_0 \end{bmatrix} \tag{2.10}$$

Equation [2.8] is written in partitioned form as

$$\dot{r}_i = \begin{bmatrix} I & -A_i\widetilde{\overline{u}}_i\overline{G}_i \end{bmatrix} \begin{bmatrix} \dot{R}_i \\ \dot{\beta}_i \end{bmatrix} \tag{2.11}$$

The equation [2.11] can be written as

$$\dot{r}_i = L_i \dot{q}_i \tag{2.12}$$

where

$$L_i = \begin{bmatrix} I & -A_i \widetilde{\overline{u}}_i \overline{G}_i \end{bmatrix} \tag{2.13}$$

$$\dot{q}_i = \begin{bmatrix} \dot{R}_i & \dot{\beta}_i \end{bmatrix}^T \tag{2.14}$$

in which $\dot{q}_i$ are the generalized velocities of the rigid link *i* defined in absolute coordinate system.

The acceleration vector of an arbitrary point on the rigid link *i* is obtained by differentiating equation [2.12]. It is written as

$$\ddot{r}_i = L_i \ddot{q}_i + A_i \left(\widetilde{\overline{\omega}}_i\right)^2 \overline{u}_i \tag{2.15}$$

where

$$\ddot{q}_i = \begin{bmatrix} \ddot{R}_i & \ddot{\beta}_i \end{bmatrix}^T \tag{2.16}$$

in which $\ddot{q}_i$ is the generalized accelerations of rigid link *i* defined in absolute coordinate system.

2.1.2 Kinematics of Flexible link

The representation of an arbitrary point $\overline{O}_i$ on the flexible link *i* is shown in Figure [2.2]. The kinematic equations that can define an arbitrary displacement of a flexible link *i* is derived using floating reference frame formulation [7]. Floating reference frame formulation uses two sets of coordinates i.e., body reference coordinates and elastic coordinates. The body reference coordinates describe the position and orientation of body coordinate system $X_iY_iZ_i$ with respect to the global coordinate system XYZ. The elastic coordinates describe the local displacements of flexible link *i* with respect to the body coordinate system $X_iY_iZ_i$. The elastic deformations of flexible link are approximated using the finite element method to obtain finite set of elastic coordinates. The elastic coordinates of finite element shown in Figure [2.3] is defined using element coordinate system $X_{ij}Y_{ij}Z_{ij}$ with respect to body coordinate system $X_iY_iZ_i$.

The position vector of an arbitrary point on flexible link *i* is defined as

$$r_i = R_i + A_i \overline{u}_i \tag{2.17}$$

where $R_i = [R_x R_y R_z]^T$ is the position vector of the body coordinate system $X_iY_iZ_i$. A_i is the transformation matrix defined using equation [2.2], and $\overline{u}_i$ is the local position vector defined with respect to $X_iY_iZ_i$.

For the flexible link, the local position vector $\overline{u}_i$ is defined as the sum of undeformed position vector and elastic deformation vector. The local position vector $\overline{u}_i$ is written as

$$\overline{u}_i = \overline{u}_i^r + \overline{u}_i^e \tag{2.18}$$

where $\overline{u}_i^r$ is the undeformed position vector, and $\overline{u}_i^e$ is elastic deformation vector that is defined as

$$\overline{u}_i^e = S_i q_i^e \tag{2.19}$$

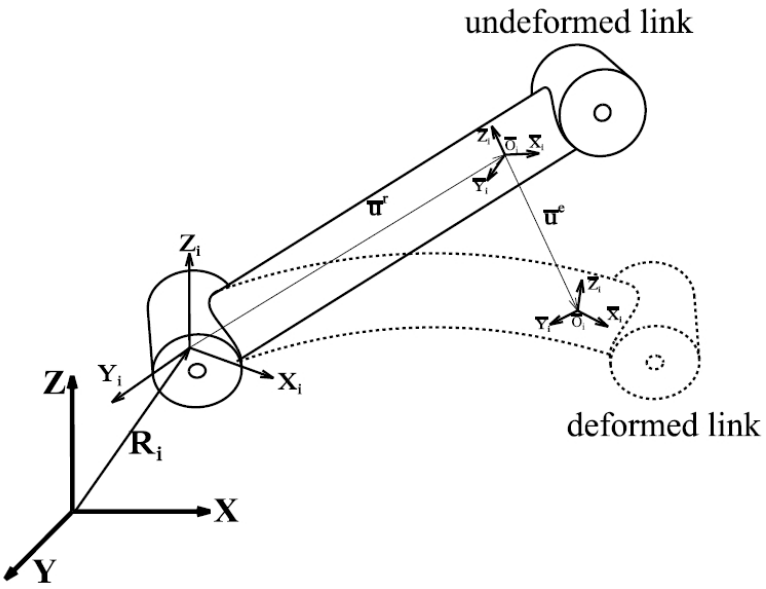

Figure 2.2: *Representation of an arbitrary point on a flexible link.*

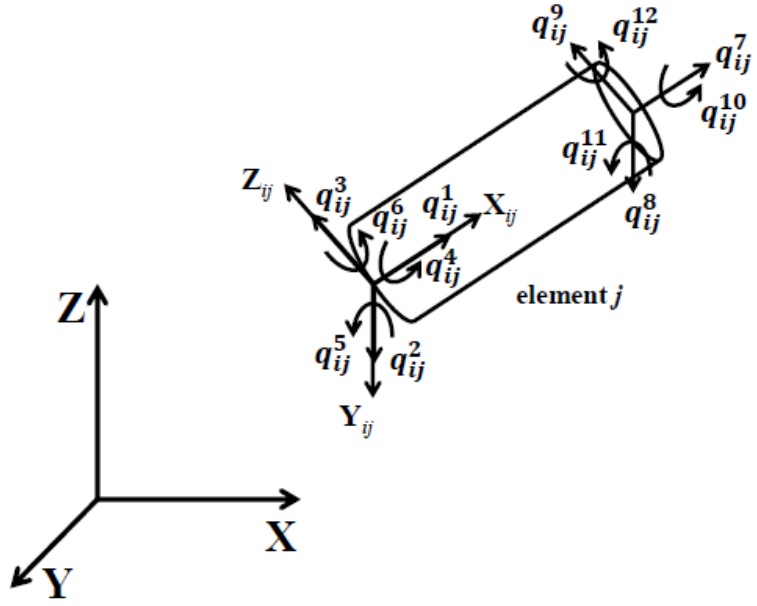

Figure 2.3: *Elastic coordinates on finite element.*

in which S_i is the shape function matrix, and q_i^e is the elastic coordinates vector. The shape function S_i is defined as

$$S_i = \begin{bmatrix} 1-\xi & 0 & 0 \\ 6(\xi-\xi^2)\eta & 1-3\xi^2+2\xi^3 & 0 \\ 6(\xi-\xi^2)\xi & 0 & 1-3\xi^2+2\xi^3 \\ 0 & -(1-\xi)\ell\zeta & -(1-\xi)\ell\eta \\ (1-4\xi+3\xi^2)\ell\zeta & 0 & (-\xi+2\xi^2-\xi^3)\ell \\ (-1+4\xi-3\xi^2)\ell\eta & (\xi-2\xi^2+\xi^3)\ell & 0 \\ \xi & 0 & 0 \\ 6(-\xi+\xi^2)\eta & 3\xi^2-2\xi^3 & 0 \\ 6(-\xi+\xi^2)\zeta & 0 & 3\xi^2-2\xi^3 \\ 0 & -\ell\xi\zeta & -\ell\xi\eta \\ (-2\xi+3\xi^2)\ell\zeta & 0 & (\xi^2-\xi^3)\ell \\ (2\xi-3\xi^2)\ell\eta & (-\xi^2+\xi^3)\ell & 0 \end{bmatrix}^T \tag{2.20}$$

in that

$$\xi = \frac{u_x}{\ell}; \eta = \frac{u_y}{\ell}; \zeta = \frac{u_z}{\ell}; \tag{2.21}$$

where ℓ is length of element, and u_x, u_y, u_z are spatial coordinates along element axis.

The velocity vector of an arbitrary point on the flexible link *i* is obtained by differentiating equation [2.17]. It is written as

$$\dot{r}_i = \dot{R}_i + A_i(\overline{\omega}_i \times \overline{u}_i) + A_i S_i \dot{q}_i^e \tag{2.22}$$

where $\overline{\omega}_i$ is angular velocity vector defined in body coordinate system $X_iY_iZ_i$. It is expressed as

$$\overline{\omega}_i = \overline{G}_i \dot{\beta}_i \tag{2.23}$$

where

$$\overline{G}_i = 2 \times \begin{bmatrix} -\beta_1 & \beta_0 & \beta_3 & -\beta_2 \\ -\beta_2 & -\beta_3 & \beta_0 & \beta_1 \\ -\beta_3 & \beta_2 & -\beta_1 & \beta_0 \end{bmatrix} \tag{2.24}$$

Equation (2.22) is written in partitioned form as

$$\dot{r}_i = \begin{bmatrix} I & -A_i\widetilde{\overline{u}}_i\overline{G}_i & A_iS_i \end{bmatrix} \begin{bmatrix} \dot{R}_i \\ \dot{\beta}_i \\ \dot{q}_i^e \end{bmatrix} \tag{2.25}$$

The Equation (2.25) can be written as

$$\dot{r}_i = L_i \dot{q}_i \tag{2.26}$$

where

$$L_i = \begin{bmatrix} I & -A_i\widetilde{\overline{u}}_i\overline{G}_i & A_iS_i \end{bmatrix} \tag{2.27}$$

$$\dot{q}_i = \begin{bmatrix} \dot{R}_i & \dot{\beta}_i & \dot{q}_i^e \end{bmatrix}^T \tag{2.28}$$

in which $\dot{q}_i$ is generalized velocities of flexible link *i* defined in absolute coordinate system.

The acceleration vector of an arbitrary point on the flexible link *i* is obtained by differentiating equation [2.26]. It is written as

$$\ddot{r}_i = L_i \ddot{q}_i + A_i \left(\widetilde{\overline{\omega}}_i\right)^2 \overline{u}_i + 2A_i \widetilde{\overline{\omega}}_i S_i \dot{q}_i^e \tag{2.29}$$

where

$$\ddot{q}_i = \begin{bmatrix} \ddot{R}_i & \ddot{\beta}_i & \ddot{q}_i^e \end{bmatrix}^T \tag{2.30}$$

in which $\ddot{q}_i$ is generalized accelerations of the flexible link *i* defined in absolute coordinate system.

2.2 Dynamics description

The equations of motion are derived using the principle of virtual work in absolute coordinate system. In this section, the dynamic equations of rigid link, flexible link, and flexible joint are presented.

2.2.1 Rigid link modeling

The virtual work of total forces acting on rigid link *i* is defined as

$$\delta W_i = \delta W_i^i + \delta W_i^e \tag{2.31}$$

where δW_i^i, and δW_i^e are respectively the virtual work of inertia forces, and external forces.

The virtual work of inertia forces acting on rigid link *i* is written as

$$\delta W_i^i = \int_{V_i} \rho_i \, \ddot{r}_i^T \, \delta r_i \, dV_i \tag{2.32}$$

where ρ_i and V_i are respectively, the mass density and volume of rigid link *ij*; $\ddot{r}_i$ and δr_i are respectively the acceleration vector and virtual displacements of an arbitrary point on rigid link *i*.

The virtual displacement δr_i is written as

$$\delta r_i = L_i \delta q_i \tag{2.33}$$

where

$$L_i = \begin{bmatrix} I & -A_i \widetilde{\overline{u}}_i \overline{G}_i \end{bmatrix} \tag{2.34}$$

$$\delta q_i = \begin{bmatrix} \delta R_i & \delta \beta_i \end{bmatrix}^T \tag{2.35}$$

where q_i is generalized coordinates of rigid link *i* defined in absolute coordinate system.

The acceleration vector $\ddot{r}_i$ of an arbitrary point can be defined using equation [2.15]. It is expressed as

$$\ddot{r}_i = L_i \ddot{q}_i + Q_i \tag{2.36}$$

in which $\ddot{q}_i$ is the generalized accelerations and Q_i is the quadratic term which is written as

$$Q_i = \ A_i \left(\widetilde{\overline{\omega}}_i\right)^2 \overline{u}_i \tag{2.37}$$

Substituting acceleration vector $\ddot{r}_i$ and virtual displacements δr_i in equation (2.32) gives

$$\delta W_i^i = \int_{V_i} \rho_i \ \ddot{q}_i^T \ L_i^T \ L_i \delta q_i \ dV_i + \int_{V_i} \rho_i \ Q_i^T \ L_i \delta q_i \ dV_i \tag{2.38}$$

$$\delta W_i^i = \left[\ddot{q}_i^T \ M_i - Q_i^{v^T}\right] \delta q_i \tag{2.39}$$

where M_i and Q_i^v are respectively the inertia matrix and quadratic velocity term.

$$M_i = \int_{V_i} \rho_i \begin{bmatrix} I & -A_i \ \widetilde{\overline{u}}_i \ \overline{G}_i \\ -A_i \ \widetilde{\overline{u}}_i \ \overline{G}_i & \overline{G}_i^T \widetilde{\overline{u}}_i^T \ \widetilde{\overline{u}}_i \overline{G}_i \end{bmatrix} dV_i \tag{2.40}$$

and

$$Q_i^v = - \int_{V_i} \rho_i \begin{bmatrix} I \\ -\overline{G}_i^T \widetilde{\overline{u}}_i^T \ A_i^T \end{bmatrix} Q_i \ dV_i \tag{2.41}$$

The virtual work of external forces acting on rigid link *i* is defined as

$$\delta W_i^e = -Q_i^{e^T} \delta q^i \tag{2.42}$$

Substituting δW_i^i, and δW_i^e in equation [2.31] yields

$$\delta W_i = \left[\ddot{q}_i^T \ M_i - Q_i^{v^T} - Q_i^{e^T}\right] \delta q_{ij} \tag{2.43}$$

From equation [2.43] the equations of motion can be rearranged as

$$M_i \ddot{q}_i = Q_i^e + Q_i^v \tag{2.44}$$

where Q_i^e and Q_i^v are respectively the external forces applied and quadratic velocity term. Equation [2.44] is the equations of motion of rigid link in absolute coordinate system.

2.2.2 Flexible link modeling

The virtual work of total forces acting on flexible link *i* is defined as

$$\delta W_i = \delta W_i^i + \delta W_i{}^s + \delta W_i{}^e \tag{2.45}$$

where δW_i^i, δW_i^s, and δW_i^e are respectively the virtual work of inertia forces, elastic forces, and external forces acting on flexible link *i*. The flexible link *i* is discretized using finite element method to get finite dimensional dynamic model. The representation of finite element *ij* on link *i* is shown in Figure [2.3]. The virtual work of flexible link *i* can be obtained by summing up the virtual work of its elements.

The virtual work of total forces acting on element *ij* is defined as

$$\delta W_{ij} = \delta W_{ij}{}^i + \delta W_{ij}{}^s + \delta W_{ij}{}^e \tag{2.46}$$

where δW^i_{ij}, δW^s_{ij}, and δW^e_{ij} are respectively the virtual work of inertia forces, elastic forces, and external forces on element *ij*.

The virtual work of inertia forces acting on element *ij* is written as

$$\delta W^i_{ij} = \int_{V_{ij}} \rho_{ij}\ \ddot{r}^T_{ij}\ \delta r_{ij}\ dV_{ij} \tag{2.47}$$

where ρ_{ij} and V_{ij} are respectively, the mass density and volume of element *ij*. $\ddot{r}_{ij}$ and δr_{ij} are respectively the acceleration vector and virtual displacements of an arbitrary point on element *ij*.

The virtual displacement δr_{ij} is written as

$$\delta r_{ij} = L_{ij}\delta q_{ij} \tag{2.48}$$

where

$$L_{ij} = \begin{bmatrix} I & -A_i\widetilde{\overline{u}}_{ij}\overline{G}_i & A_iS_{ij} \end{bmatrix} \tag{2.49}$$

$$\delta q_{ij} = \begin{bmatrix} \delta R_{ij} & \delta\beta_{ij} & \delta q^e_{ij} \end{bmatrix}^T \tag{2.50}$$

where q_{ij} are the generalized coordinates of element *ij*.

The acceleration vector $\ddot{r}_{ij}$ of an arbitrary point can be defined using equation [2.29]. It is expressed as

$$\ddot{r}_{ij} = L_{ij}\ddot{q}_{ij} + Q_{ij} \tag{2.51}$$

where

$$\ddot{q}_{ij} = \begin{bmatrix} \ddot{R}_{ij} & \ddot{\beta}_{ij} & \ddot{q}^e_{ij} \end{bmatrix}^T \tag{2.52}$$

and

$$Q_{ij} = A_i\left(\widetilde{\overline{\omega}}_i\right)^2\overline{u}_{ij} + 2A_i\widetilde{\overline{\omega}}_iS_{ij}\dot{q}^e_{ij} \tag{2.53}$$

in which $\ddot{q}_{ij}$ are the generalized accelerations, and Q_{ij} is the quadratic term.

Substituting acceleration vector $\ddot{r}_{ij}$ and virtual displacements δr_{ij} in equation (2.47) gives

$$\delta W^i_{ij} = \int_{V_{ij}} \rho_{ij}\ \ddot{q}^T_{ij}\ L^T_{ij}\ L_{ij}\delta q_{ij}\ dV_{ij} + \int_{V_{ij}} \rho_{ij}\ Q^T_{ij}\ L_{ij}\delta q_{ij}\ dV_{ij} \tag{2.54}$$

$$\delta W^i_{ij} = \left[\ddot{q}^T_{ij}\ M_{ij} - Q^{v^T}_{ij}\right]\delta q_{ij} \tag{2.55}$$

where M_{ij} and Q^v_{ij} are respectively the inertia matrix and quadratic velocity term.

$$M_{ij} = \int_{V_{ij}} \rho_{ij}\begin{bmatrix} I & -A_i\ \widetilde{\overline{u}}_{ij}\ \overline{G}_i & A_i\ S_{ij} \\ & \overline{G}^T_i\widetilde{\overline{u}}^T_{ij}\ \widetilde{\overline{u}}_{ij}\overline{G}_i & \overline{G}^T_i\widetilde{\overline{u}}^T_{ij}S_{ij} \\ symmetric & & S^T_{ij}S_{ij} \end{bmatrix} dV_{ij} \tag{2.56}$$

and

$$Q^v_{ij} = -\int_{V_{ij}} \rho_{ij}\begin{bmatrix} I \\ -\overline{G}^T_i\widetilde{\overline{u}}^T_{ij}A^T_i \\ S^T_{ij}A^T_i \end{bmatrix} Q_{ij}\ dV_{ij} \tag{2.57}$$

The virtual work of elastic forces due to the deformation of element *ij* can be defined as

$$\delta W_{ij}^{s} = -\int_{V_{ij}} \sigma_{ij}^{T} \delta\varepsilon_{ij} dV_{ij} \tag{2.58}$$

where σ_{ij} and ε_{ij} are respectively the stress and the strain vectors of element *ij*.

$$\varepsilon_{ij} = D_{ij}\overline{u}_{ij}^{e} = D_{ij}S_{ij}q_{ij}^{e} \tag{2.59}$$

$$\sigma_{ij} = E_{ij}\varepsilon_{ij} = E_{ij}D_{ij}S_{ij}q_{ij}^{e} \tag{2.60}$$

Substituting equation [2.59] and [2.60] in equation [2.58] gives

$$\begin{aligned} \delta W_{ij}^{s} &= -q_{ij}^{e^T}\left[\int_{V_{ij}} (D_{ij}S_{ij})^{T} E_{ij} D_{ij} S_{ij} dV_{ij}\right]\delta q_{ij}^{e} \\ &= -q_{ij}^{e^T} K_{ij}^{e} \delta q_{ij}^{e} \end{aligned} \tag{2.61}$$

where K_{ij}^{e} is the element stiffness matrix defined as

$$K_{ij}^{e} = \int_{V_{ij}} (D_{ij}S_{ij})^{T} E_{ij} D_{ij} S_{ij} dV_{ij} \tag{2.62}$$

in which D_{ij} is the differential operator, S_{ij} is the element shape function matrix and E_{ij} is the elastic coefficient.

The virtual work of external forces acting on element *ij* is defined as

$$\delta W_{ij}^{e} = -Q_{ij}^{e^T} \delta q^{i} \tag{2.63}$$

Substituting δW_{ij}^{i}, δW_{ij}^{s} and δW_{ij}^{e} in equation [2.46] yields

$$\delta W_{ij} = \left[\ddot{q}_{ij}^{T}\, M_{ij} - Q_{ij}^{v^T} - q_{ij}^{e^T} K_{ij}^{e} - Q_{ij}^{e^T}\right]\delta q_{ij} \tag{2.64}$$

From equation [2.64] the equations of motion can be rearranged as

$$M_{ij}\ddot{q}_{ij} = Q_{ij}^{e} + Q_{ij}^{v} + Q_{ij}^{s} \tag{2.65}$$

where Q_{ij}^{e} are the applied external forces. Q_{ij}^{v} and Q_{ij}^{s} are respectively the quadratic velocity term and elastic forces acting on element *ij*.

The virtual work of total forces acting on flexible link *i* is defined as

$$\delta W_{i} = \sum_{j=1}^{n_e} \delta W_{ij}{}^{i} + \sum_{j=1}^{n_e} \delta W_{ij}{}^{s} + \sum_{j=1}^{n_e} \delta W_{ij}{}^{e} \tag{2.66}$$

where n_e is the number of finite elements.

Equation [2.65] can be extended to all finite elements in flexible link *i* and assembled based on element connectivity to form a dynamic model of flexible link.

Using the equation [2.66], the equations of motion of the flexible link *i* is defined as

$$M_{i}\ddot{q}_{i} = Q_{i}^{e} + Q_{i}^{v} + Q_{i}^{s} \tag{2.67}$$

where Q_{i}^{e} are the external forces applied on flexible link *i*. Q_{i}^{v} and Q_{i}^{s} are respectively the quadratic velocity term and elastic forces acting on flexible link *i*.

The equations of motion in absolute coordinate system for n link manipulator can be defined as

$$\begin{bmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & M_n \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_n \end{bmatrix} = \begin{bmatrix} Q_1^e \\ Q_2^e \\ \vdots \\ Q_n^e \end{bmatrix} + \begin{bmatrix} Q_1^v \\ Q_2^v \\ \vdots \\ Q_n^v \end{bmatrix} + \begin{bmatrix} Q_1^s \\ Q_2^s \\ \vdots \\ Q_n^s \end{bmatrix} \tag{2.68}$$

The equation [2.68] expressed in compact form as

$$\overline{M}\ddot{\overline{q}} = \overline{Q}^e + \overline{Q}^v + \overline{Q}^s \tag{2.69}$$

where

$$\overline{M} = \begin{bmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & M_n \end{bmatrix} \tag{2.70}$$

$$\ddot{\overline{q}} = \begin{bmatrix} \ddot{q}_1 & \ddot{q}_2 & \cdots & \ddot{q}_n \end{bmatrix}^T \tag{2.71}$$

$$\overline{Q}^e = \begin{bmatrix} Q_1^e & Q_2^e & \cdots & Q_n^e \end{bmatrix}^T \tag{2.72}$$

$$\overline{Q}^v = \begin{bmatrix} Q_1^v & Q_2^v & \cdots & Q_n^v \end{bmatrix}^T \tag{2.73}$$

$$\overline{Q}^s = \begin{bmatrix} Q_1^s & Q_2^s & \cdots & Q_n^s \end{bmatrix}^T \tag{2.74}$$

The equation [2.68] is the dynamic model derived in absolute coordinates. The relative motions between the flexible links are imposed using recursive kinematic formulation. The recursive kinematic formulation is presented in section [2.3]

2.2.3 Flexible joint modeling

The revolute joint *j* with actuator and transmission system is shown in Figure [2.4]. The actuator is assumed as electric motor and torsional spring represents the flexibility induced due to the transmission system. θ_j and θ_i respectively are the rotations of actuator *j* and link *i*.

The virtual work of torque exserted on link *i* by actuator *j* and transmission system is defined as

$$\delta W_j = T\ \delta\theta_{ij} \tag{2.75}$$

where T is the total torque acting at the joint. The equation [2.75] is written in explicit form as

$$\delta W_j = (J_j\ddot{\theta}_j + C_j(\dot{\theta}_j - \dot{\theta}_i) + K_j(\theta_j - \theta_i) - T_j)\ \delta\theta_{ij} \tag{2.76}$$

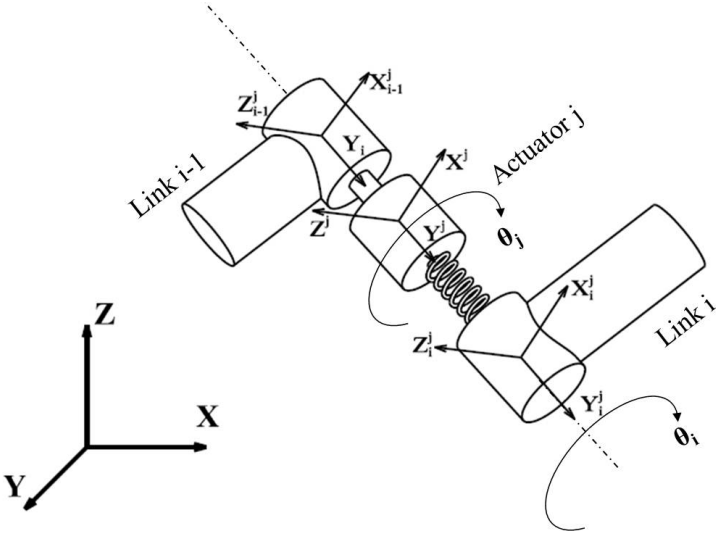

Figure 2.4: *Flexible joint j assembly*

where J_j is the inertia of the motor. K_j and C_j are the stiffness and damping coefficients of the transmission system. T_j is the torque produced by motor. $\delta\theta_{ij}$ is the virtual change at joint. The equations of motions of joint assembly is written as

$$J_j\ddot{\theta}_j + C_j(\dot{\theta}_j - \dot{\theta}_i) + K_j(\theta_j - \theta_i) = T_j \tag{2.77}$$

2.3 Recursive Kinematic Formulation

Consider two flexible link $i-1$ and *i* shown in Figure [2.5] which are connected by a revolute joint *j*. The joint allows relative rotation along joint axis and has one rigid body coordinates θ_i.

The following kinematic relationship for revolute joint holds the relation between generalized coordinates and joint coordinates [76]

$$R_i + A_i\overline{u}_i^j - R_{i-1} - A_{i-1}\overline{u}_{i-1}^j = 0 \tag{2.78}$$

$$\omega_i = \omega_{i-1} + \omega_{i-1}^j - \omega_i^j + \omega_{i,i-1} \tag{2.79}$$

where $\overline{u}_i^j$ and $\overline{u}_{i-1}^j$ are local position vectors of joint defined on link *i* and $i-1$ respectively. ω_i^j and ω_{i-1}^j are respectively the local angular velocity vectors of joint due to elastic deformations on link *i* and $i-1$. These vector quantities are defined as

$$\omega_{i-1}^j = A_{i-1}S_{i-1}^{jr}\dot{q}_{i-1}^e \tag{2.80}$$

$$\omega_i^j = A_iS_i^{jr}\dot{q}_i^e \tag{2.81}$$

in which S_i^{jr} and S_{i-1}^{jr} are respectively the constant shape function matrix of joint rotations due to elastic deformations on link *i* and $i-1$. $\omega_{i,i-1}$ is relative angular velocity vector of link *i* with respect to link *i-1* is expressed as

$$\omega_{i,i-1} = \nu_{i-1}\dot{\theta}_i \tag{2.82}$$

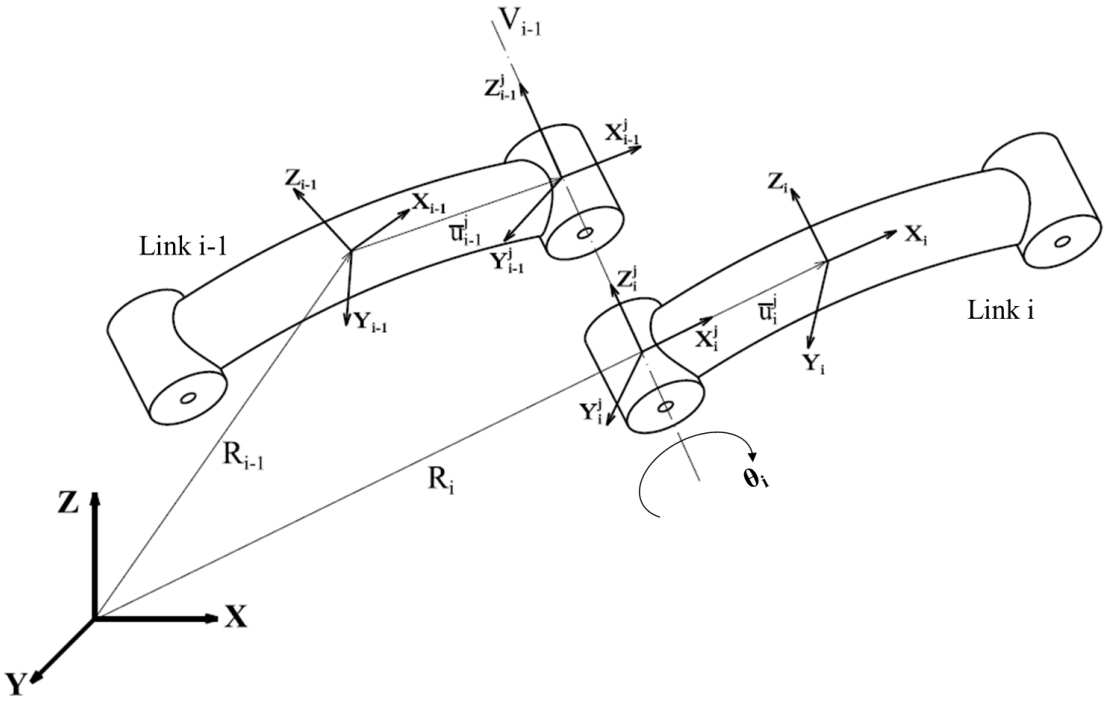

Figure 2.5: *Representation of Relative body Coordinates*

where ν_{i-1} is rotation axis defined with respect to link $i-1$ in global coordinate system XYZ.

$$\nu_{i-1} = A_{i-1}\overline{\nu}_{i-1} \tag{2.83}$$

$\overline{\nu}_{i-1}$ is constant vector defined with respect to link $i-1$ in body coordinate system $X_{i-1}Y_{i-1}Z_{i-1}$.

Differentiating equation [2.78] twice with respect to time and equation [2.79] once with respect to time gives

$$\begin{aligned}\ddot{R}_i - A_i\widetilde{\overline{u}}_i^j\overline{G}_i\ddot{\beta}_i + A_iS_i^{jt}\ddot{q}_i^e = \ddot{R}_{i-1} - A_{i-1}\widetilde{\overline{u}}_{i-1}^j\overline{G}_{i-1}\ddot{\beta}_{i-1} \\ + A_{i-1}S_{i-1}^{jt}\ddot{q}_{i-1}^e + \gamma_R\end{aligned} \tag{2.84}$$

$$\dot{\omega}_i = \dot{\omega}_{i-1} + A_{i-1}S_{i-1}^{jr}\ddot{q}_{i-1}^e - A_iS_i^{jr}\ddot{q}_i^e + A_{i-1}\overline{\nu}_{i-1}\ddot{\theta}_i + \gamma_\beta \tag{2.85}$$

where S_i^{jt} and S_{i-1}^{jt} are respectively the shape functions of joint translations defined on link *i* and *i-1*. γ_R and γ_β are written as

$$\begin{aligned}\gamma_R = -A_i\left(\widetilde{\overline{\omega}}_i\right)^2\overline{u}_i^j + A_{i-1}\left(\widetilde{\overline{\omega}}_{i-1}\right)^2\overline{u}_{i-1}^j - 2A_i\widetilde{\overline{\omega}}_iS_i^{jt}\dot{q}_i^e \\ + 2A_{i-1}\widetilde{\overline{\omega}}_{i-1}S_{i-1}^{jt}\dot{q}_{i-1}^e\end{aligned} \tag{2.86}$$

$$\gamma_\beta = A_{i-1}\widetilde{\overline{\omega}}_{i-1}\overline{\nu}_{i-1}\dot{\theta}_i + A_{i-1}\widetilde{\overline{\omega}}_{i-1}S_{i-1}^{jr}\dot{q}_{i-1}^e - A_i\widetilde{\overline{\omega}}_iS_i^{jr}\dot{q}_i^e \tag{2.87}$$

The equation [2.84] and [2.85] can be written in a compact form as

$$D_i\ddot{q}_i = D_{i-1}\ddot{q}_{i-1} + H_i\ddot{P}_i + \gamma_i \tag{2.88}$$

where

$$D_i = \begin{bmatrix} I & -A_i \, \widetilde{\overline{u}}_i^j \, \overline{G}_i & A_i S_i^{jt} \\ 0 & A_i \overline{G}_i & A_i S_i^{jr} \\ 0 & 0 & I \end{bmatrix} \tag{2.89}$$

$$D_{i-1} = \begin{bmatrix} I & -A_{i-1} \widetilde{\overline{u}}_{i-1}^j \overline{G}_{i-1} & A_{i-1} S_{i-1}^{jt} \\ 0 & A_{i-1} \overline{G}_{i-1} & A_{i-1} S_{i-1}^{jr} \\ 0 & 0 & 0 \end{bmatrix} \tag{2.90}$$

$$H_i = \begin{bmatrix} A_{i-1} \overline{\nu}_{i-1} & 0 \\ 0 & I \end{bmatrix} \tag{2.91}$$

$$\ddot{P}_i = \begin{bmatrix} \ddot{\theta}_i & \ddot{q}_i^e \end{bmatrix}^T \tag{2.92}$$

$$\gamma^i = \begin{bmatrix} \gamma_R & \gamma_\beta \end{bmatrix}^T \tag{2.93}$$

The generalization of recursive kinematic formulation for a n link manipulator is expressed as

$$\begin{bmatrix} D_1 & 0 & \cdots & 0 \\ -D_1 & D_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & -D_{n-1} & D_n \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_n \end{bmatrix} = \begin{bmatrix} H_1 & 0 & \cdots & 0 \\ 0 & H_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & H_n \end{bmatrix} \begin{bmatrix} \ddot{P}_1 \\ \ddot{P}_2 \\ \vdots \\ \ddot{P}_n \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \end{bmatrix} \tag{2.94}$$

The equation [2.94] can be written in compact form as

$$\overline{D} \, \ddot{\overline{q}} = \overline{H} \, \ddot{\overline{P}} + \overline{\gamma} \tag{2.95}$$

where

$$\overline{D} = \begin{bmatrix} D_1 & 0 & \cdots & 0 \\ -D_1 & D_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & -D_{n-1} & D_n \end{bmatrix} \tag{2.96}$$

$$\ddot{\overline{q}} = \begin{bmatrix} \ddot{q}_1 & \ddot{q}_2 & \cdots & \ddot{q}_n \end{bmatrix}^T \tag{2.97}$$

$$\begin{bmatrix} H_1 & 0 & \cdots & 0 \\ 0 & H_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & H_n \end{bmatrix} \tag{2.98}$$

$$\ddot{\overline{P}} = \begin{bmatrix} \ddot{P}_1 & \ddot{P}_2 & \cdots & \ddot{P}_n \end{bmatrix}^T \tag{2.99}$$

$$\ddot{\overline{\gamma}} = \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_n \end{bmatrix}^T \tag{2.100}$$

The generalized accelerations of n link manipulator $\ddot{\overline{q}}$ in absolute coordinate system can be expressed in terms of relative or independent coordinates as

$$\ddot{\overline{q}} = \overline{B}\,\ddot{\overline{P}} + \widetilde{\gamma} \tag{2.101}$$

where

$$\overline{B} = \overline{D}^{-1}\overline{H} \tag{2.102}$$

$$\widetilde{\gamma} = \overline{D}^{-1}\overline{\gamma} \tag{2.103}$$

Substituting the generalized acceleration $\ddot{\overline{q}}$ in equation [2.69] and premultiplying with $\overline{B}^T$ gives the dynamic model of n link manipulator in relative or independent coordinates form. It is written as

$$\overline{B}^T\overline{M}\;\overline{B}\ddot{P}_i = \overline{B}^T(\overline{Q}^e + \overline{Q}^v + \overline{Q}^s - \overline{M}\widetilde{\gamma}) \tag{2.104}$$

which is written as

$$M\ddot{P} = Q \tag{2.105}$$

where

$$M = \overline{B}^T\overline{M}\;\overline{B} \tag{2.106}$$

$$Q = \overline{B}^T(\overline{Q}^e + \overline{Q}^v + \overline{Q}^s - \overline{M}\widetilde{\gamma}) \tag{2.107}$$

The equation [2.105] is a coupled and nonlinear dynamic model of n link manipulator which can be used for numerical simulation and model based control design purpose.

2.4 Multi-Body Code Structure

A gerneral purpose multi-body code has been developed in MATLAB to get the dynamic model for numerical simulation and model based control design purpose. The structure of the multi-body code is shown in Figure [2.6]. The input to the multi-body code consists of physical parameters of body, joint, and actuators. The dynamic model is nonlinear and configuration dependent. Hence, to get the dynamic parameters such as inertia matrix, coriolis and centrifugal matrix, stiffness matrix, and damping matrix the position and velocities of manipulator links are necessary. In the multi-body code the damping matrix is defined using Rayleigh damping. Overall the input file format consists of body definition, joint definition, actuator definition, position vector and velocity vector of the manipulator.

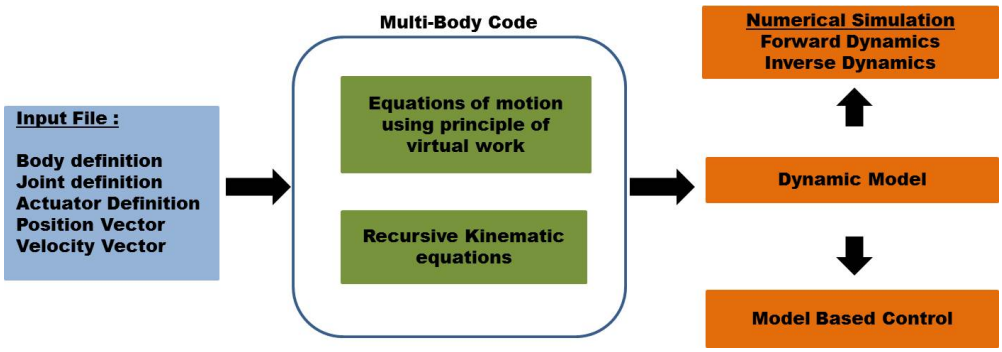

Figure 2.6: *The structure of the multi-body code.*

Body definition

The body definition is declared as a MATLAB structure called body structure. The input fields to the body structure is shown in Table [2.1]. The body structure contains all the necessary fields that are required for dynamic modeling. The user can choose the rigid link modeling or flexible link modeling in the structure input field "body".

Table 2.1: *The input fields of body structure.*

Field	Input Format	Description
name	String	Input to assign name to the body
body	Number	1 = Rigid link / 2 = Flexible link
element	Number	Number of finite element to discretize the link (1 - for rigid link, n - for flexible link)
density	Number	Density of the link (Kg/m^3)
youngs_mod	Number	Youngs Modulus of the link (MPa)
shear_mod	Number	Shear Modulus of the link (MPa)
moment_inertia	Number	Area Moment of Inertia (m^4)
polar_inertia	Number	Moment of Inertia (m^4)
length	Number	Length of the link (m)
area	Number	Cross-section area (m^2)
alpha	Number	Rayleigh Damping constant
beta	Number	Rayleigh Damping constant

Joint definition

The joint definition is declared as a MATLAB structure called joint structure. The input fields to the joint structure is shown in Table [2.2]. The user can choose the rigid joint or flexible joint modeling in the structure input field "joint". In the joint structure, inertia, damping, and stiffness properties are considered to model dynamics of flexible joint. The flexible joint is necessary for the modeling of flexible joint manipulator such as KUKA-DLR light weight manipulator.

Table 2.2: *The input fields of joint structure.*

Field	Input Format	Description
name	String	Input to assign name to the joint
joint	Number	1 = rigid joint ; 2 = flexible joint;
inertia	Number	Inertia of the motor
stiff	Number	Stiffness of the joint
damp	Number	Damping of the joint
axis	Number	1 = X-axis; 2 = Y-axis; 3 = Z-axis;
body1	Number	Link number
body2	Number	Link number
body1_frame	Vector	Joint location w.r.t body1
body2_frame	Vector	Joint location w.r.t body2

Actuator definition

The actuator definition is declared as a MATLAB structure called actuator structure. The actuator structure is optional choice. It is particularly useful for forward dynamic simulation. The input fields to the actuator structure is shown in Table [2.3].

Table 2.3: *The input fields of actuator structure.*

Field	Input Format	Description
name	String	Input to assign name to the actuator
load	Number	Input torque (N-m)
axis	Number	1 = X-axis; 2 = Y-axis; 3 = Z-axis;
body1	Number	Link number
body2	Number	Link number

2.5 Study of flexibility effects on spatial manipulator

A spatial RRR manipulator shown in Figure [3.6], is considered to demonstrate the effect of link and joint flexibility on manipulator dynamics. The physical parameters of a RRR spatial manipulator is presented in Table [2.4]. Uniform cross-section and material properties are assumed on each link.

Table 2.4: *The physical parameters of a RRR flexible manipulator.*

Parameter	Link 1	Link 2	Link 3
Link Length (m)	1	4.0	3.5
C/s Area (m^2)	0.028	0.0020	0.0008
Moment of Inertia (m^4)	8.33×10^{-7}	6.24×10^{-7}	5.37×10^{-7}
Polar moment of Inertia (m^4)	1.66×10^{-6}	1.24×10^{-6}	1.07×10^{-6}
Tensile Modulus (MPa)	206000		
Shear Modulus (MPa)	79300		
Density (Kg/m^3)	8253		

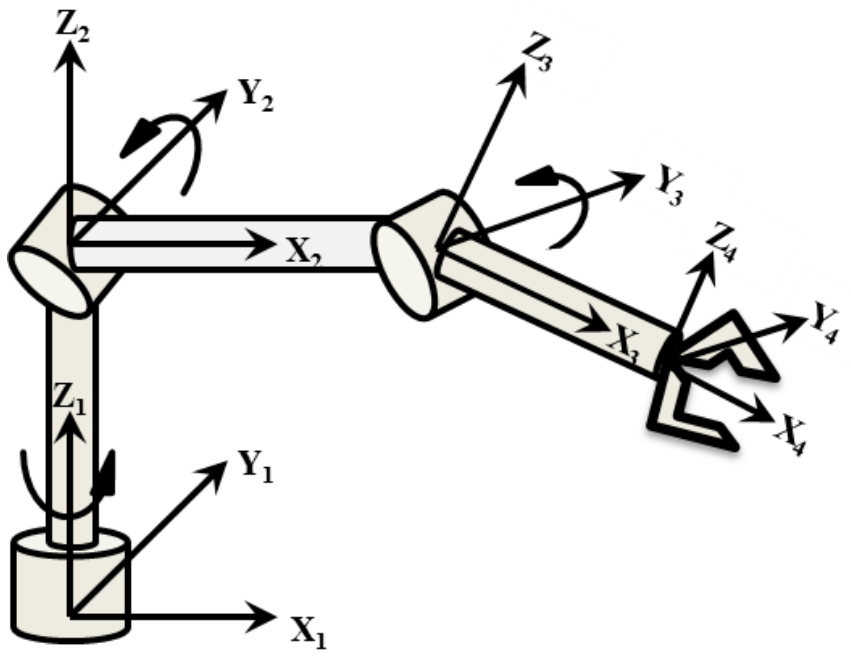

Figure 2.7: *Spatial RRR flexible manipulator.*

The following cases are considered to study the effect of flexibility on RRR spatial manipulator dynamics

1. Rigid links and Rigid joints
2. Flexible links and Rigid joints
3. Flexible links and Fleixble joints

Each flexible link is discretized using two finite element beams with six degrees of freedom on each node and one degree of freedom for rigid body rotations, i.e. θ_i where $i = 1, 2, 3$. Manipulator joint has one rigid body rotation, i.e. θ_j where $j = 1, 2, 3$. The torsional stiffness K_j for $j = 1, 2, 3$ at manipulator joints is defined as 5000 Nm/rad. The damping effects on links and joints are ignored in the numerical simulation.

2.5.1 Simulation results

A constant torque of 400 Nm, is applied at each manipulator joint for each case to compare manipulator endeffector $X_4Y_4Z_4$ motion. The endeffector $X_4Y_4Z_4$ motion in global coordinate system along the X,Y and Z direction is shown in Figure [2.8] - [2.10]. The elastic displacements of manipulator endeffector $X_4Y_4Z_4$ along the X,Y and Z direction is shown in Figure [2.11] - Figure [2.13]. It shows the elastic displacements of flexible manipulator (i.e. Flexible links and Rigid joints, Flexible links and Fleixble joints) endeffector $X_4Y_4Z_4$ with respect to the rigid manipulator (i.e. Rigid links and Rigid joints) endeffector $X_4Y_4Z_4$ motion.

The joint response of RRR manipulator along joint 1, joint 2 and joint 3 is shown in Figure [2.8] - Figure [2.10]. The joint deformations of flexible manipulator (i.e.

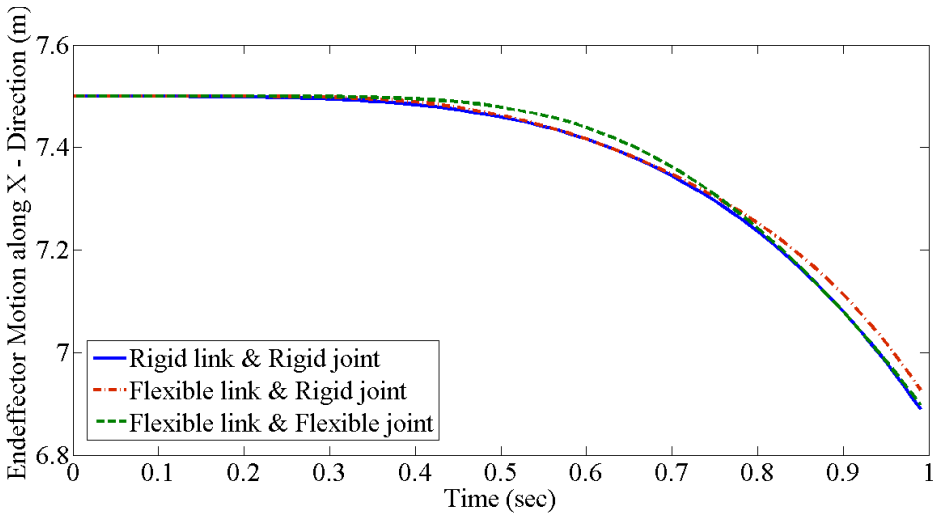

Figure 2.8: *Endeffector $X_4Y_4Z_4$ trajectory response along X-direction.*

Flexible links and Rigid joints, Flexible links and Fleixble joints) with respect to rigid manipulator joint motion is shown in Figure [2.17] - Figure [2.19].

In addition to link flexibility, the flexibility at manipulator joint can significantly alter the motion of the manipulator joint and eventually effects the endeffector $X_4Y_4Z_4$ motion. For the long reach manipulators having link dimensions in the order of few meters, the small elastic deformation at manipulator joint can lead to large endeffector position error. It is particulary evident in Figure [2.11] - Figure [2.13].

The numerical simulation results show that the link and joint flexibility can significantly alter the overall manipulator endeffector motion. Hence it is necessary to include the dynamics of link and joint flexibility to accurately represent the dynamic behaviour of the system.

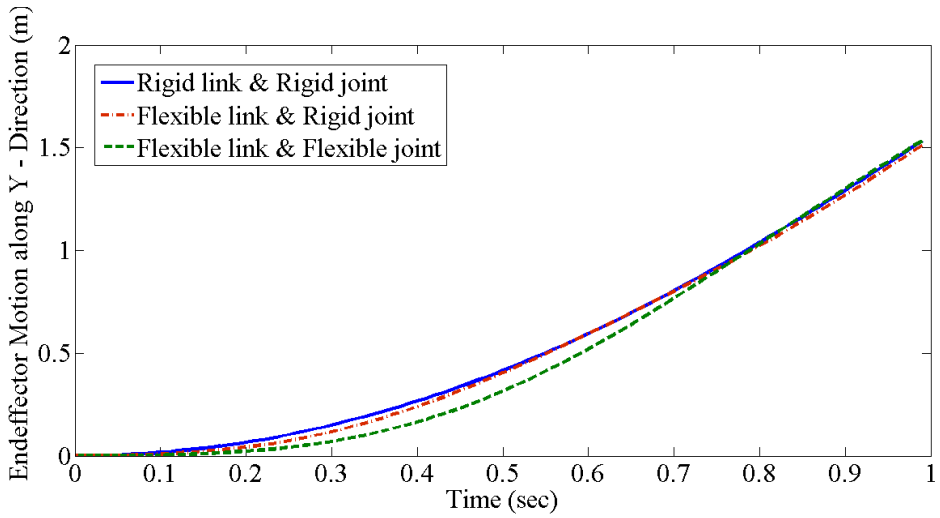

Figure 2.9: *Endeffector $X_4Y_4Z_4$ trajectory response along Y-direction.*

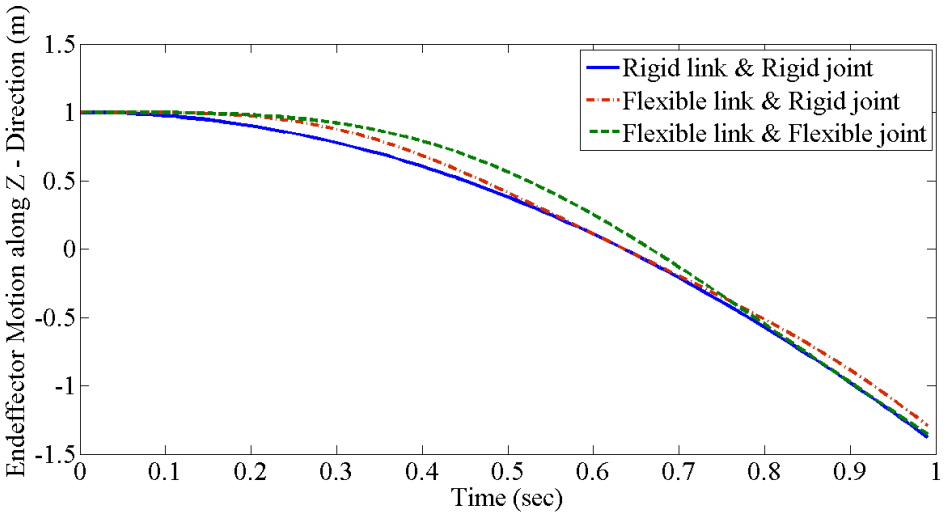

Figure 2.10: *Endeffector $X_4Y_4Z_4$ trajectory response along Z-direction.*

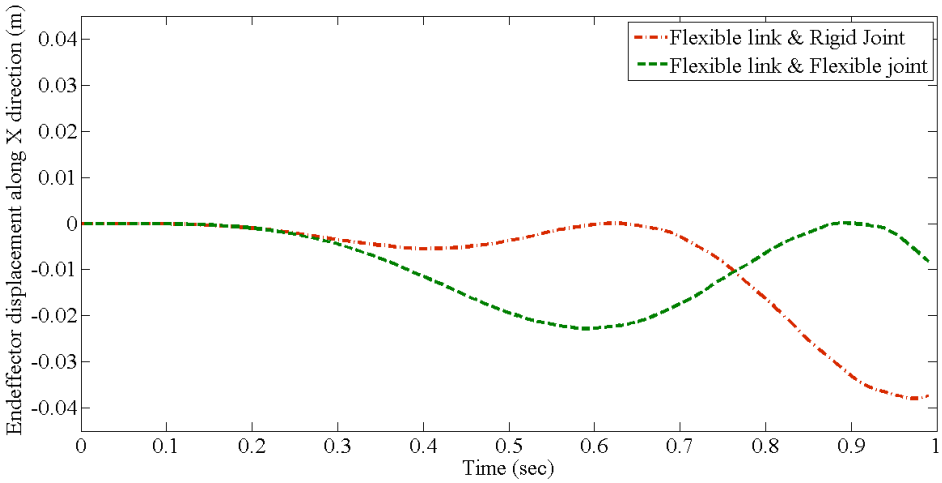

Figure 2.11: *Elastic displacements of endeffector $X_4Y_4Z_4$ with respect to rigid manipulator endeffector motion in X-direction.*

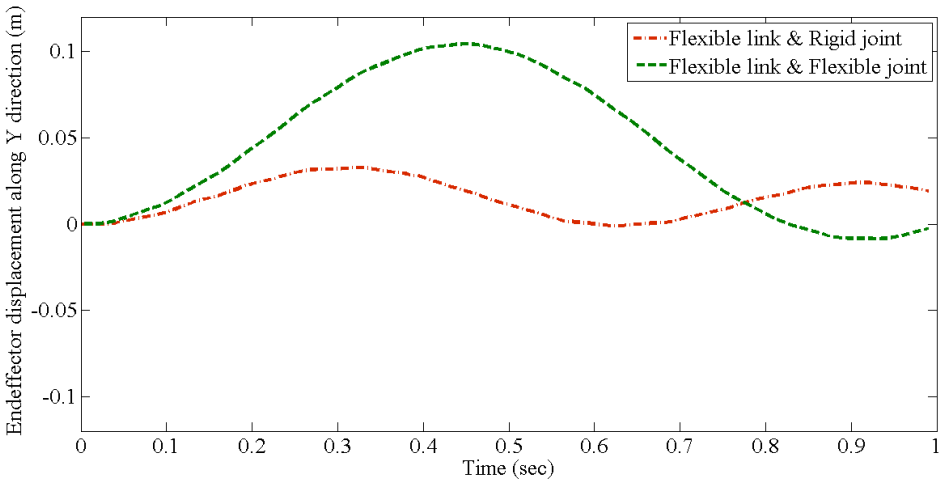

Figure 2.12: *Elastic displacements of endeffector $X_4Y_4Z_4$ with respect to rigid manipulator endeffector motion in Y-direction.*

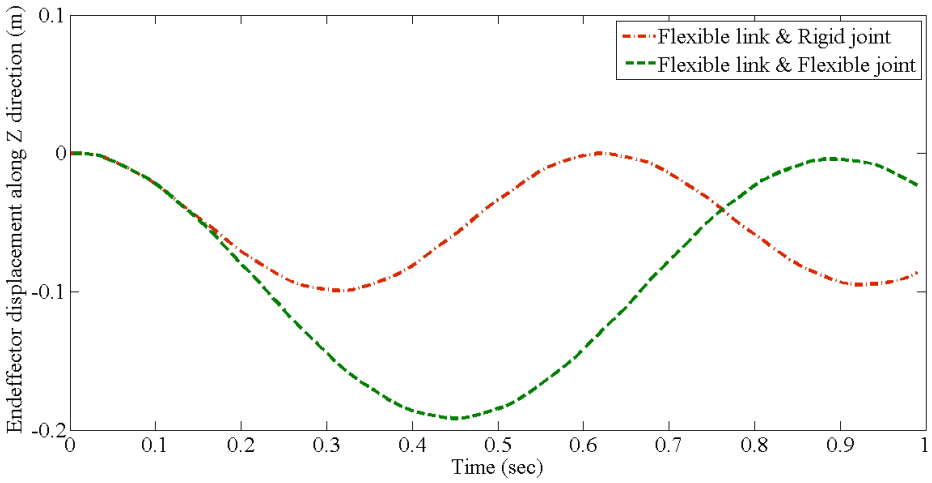

Figure 2.13: *Elastic displacements of endeffector $X_4Y_4Z_4$ with respect to rigid manipulator endeffector motion in Z-direction.*

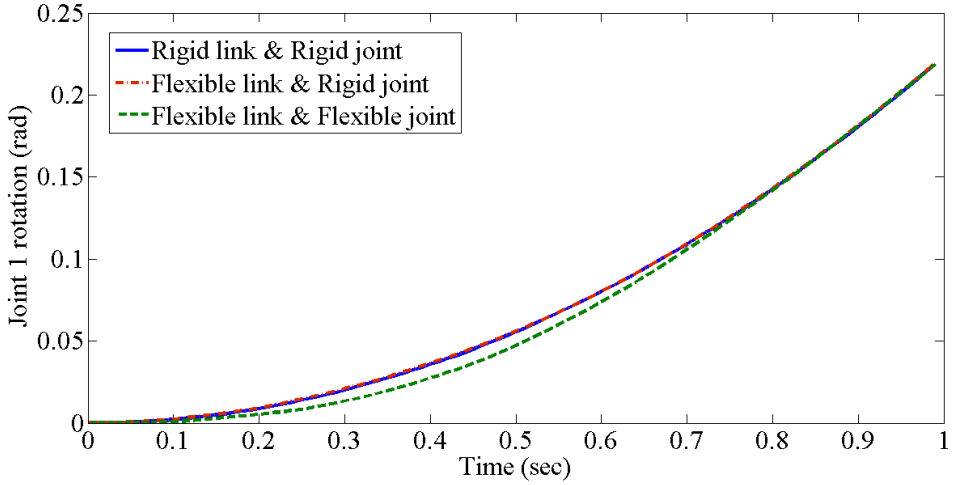

Figure 2.14: *Joint θ_1 trajectory response.*

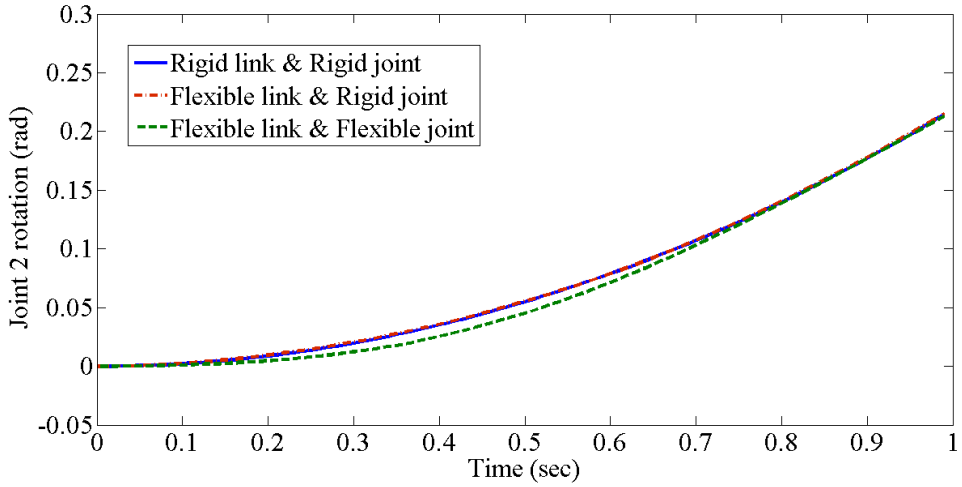

Figure 2.15: *Joint θ_2 trajectory response.*

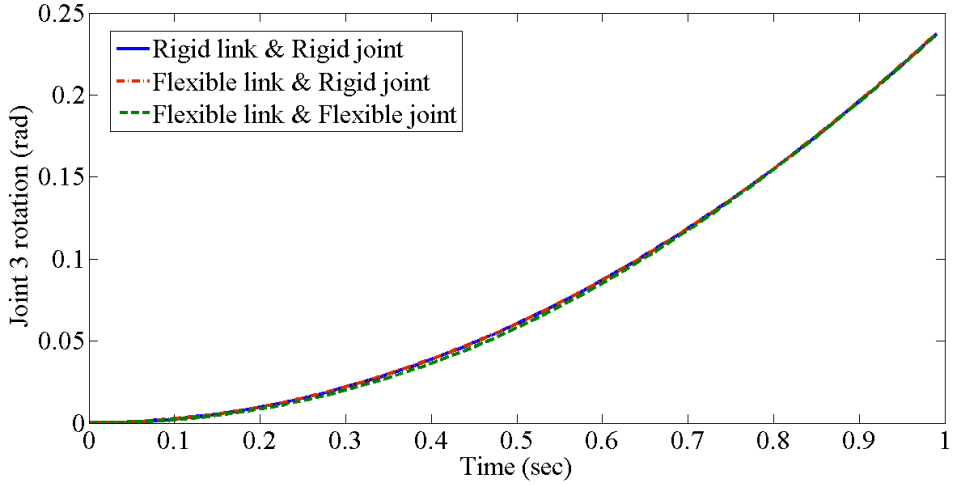

Figure 2.16: *Joint θ_3 trajectory response.*

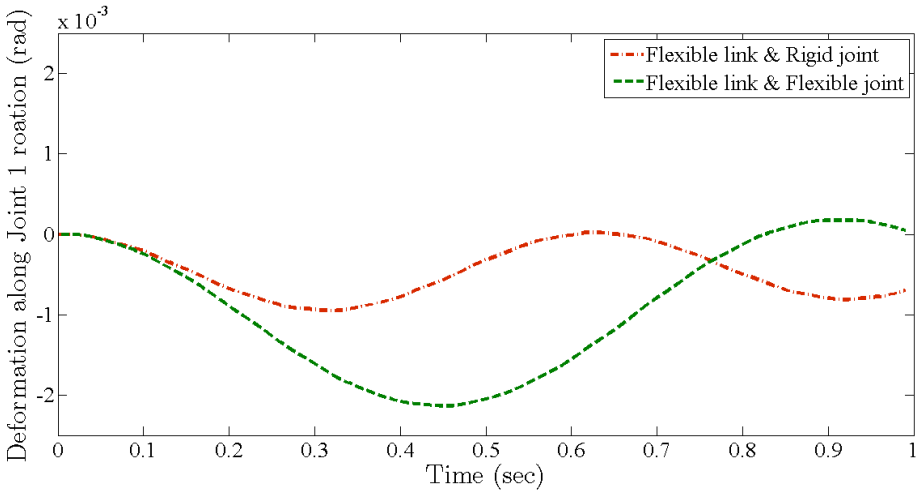

Figure 2.17: *Elastic deformations of joint θ_1 with respect to rigid manipulator joint motion.*

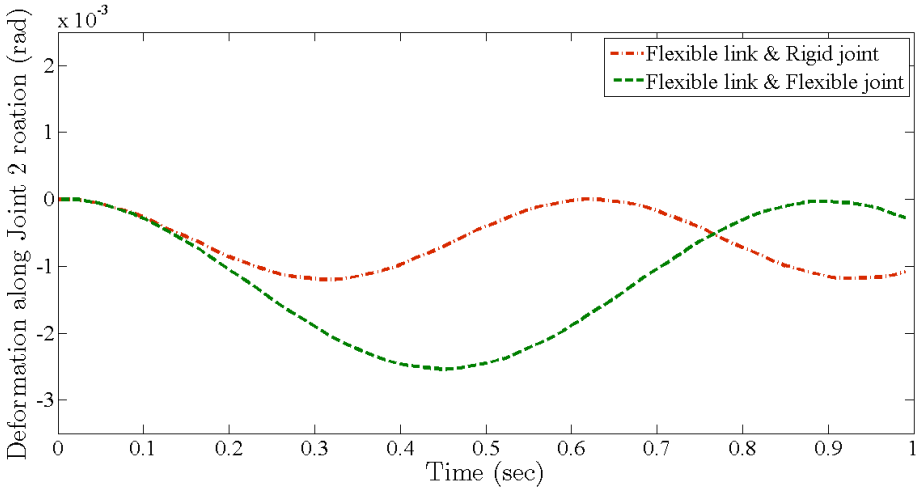

Figure 2.18: *Elastic deformations of joint θ_2 with respect to rigid manipulator joint motion.*

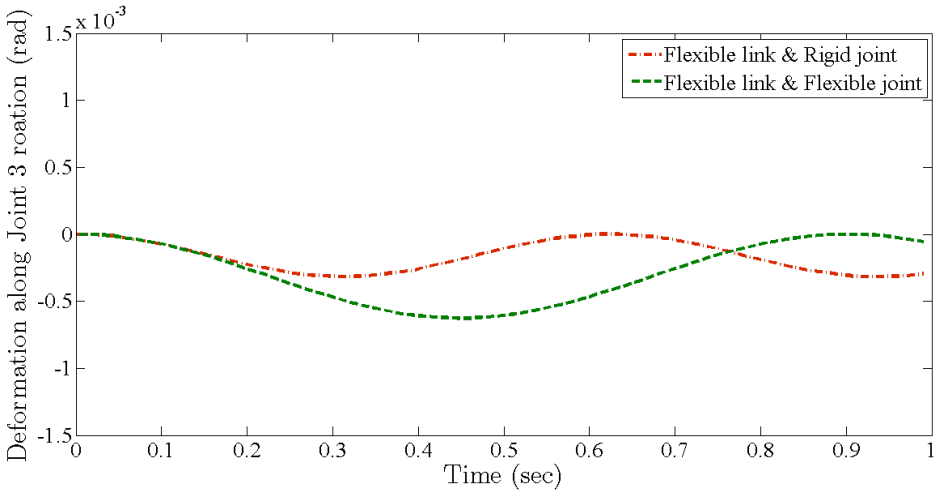

Figure 2.19: *Elastic deformations of joint θ_3 with respect to rigid manipulator joint motion.*

CHAPTER 3

Control of flexible manipulators

Motion control of robot manipulator is important to achieve high speed operations and multi-functionality. In that, dynamic model identification and control are two subsets which are important to attain desired performance. In the previous chapter, the dynamic formulation is presented to identify the dynamic model based on physical parameters of the manipulator. In this chapter, model based control for trajectory tracking of spatial flexible manipulator is presented.

The objective of trajectory tracking control in the case of flexible manipulators is to follow the desired reference trajectory and minimize vibrations of the end-effector along trajectory. The approach to design a model based controller in order to meet the desired performance varies based on the type of flexibility in the system. Link flexibility is considered as most difficult to control because the flexibility is distributed and has infinite degrees of freedom. In this chapter, the controller design is mainly addressed for flexible link manipulators.

The most challenging problem in design of controller for flexible link manipulators is under actuation and non-minimal phase nature. Under actuation is due to finite number of actuators to control infinite degrees of freedom that arise due to link flexibility. Non-minimum phase nature occurs because of non-collocation of actuators and sensors.

Lets consider, the equations of motion of a flexible link manipulator which can be written in explicit form as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D\dot{q} + Kq = B\tau \tag{3.1}$$

where $q = [q_r \;\; q_f]^T$ are rigid and elastic coordinates of the manipulator; q_r is the $n \times 1$ vector that represents rigid body rotations of the n manipulator joints, and q_f is the $m \times 1$ vector that represent elastic coordinates of the link. The number of elastic

coordinates depends on the number of finite elements used to discretize the link. $M(q)$ is the Inertia matrix, $C(q,\dot{q})\dot{q}$ is the Coriolis and centrifugal vector, $D\dot{q}$ is the frictional and damping forces, Kq represents the internal forces due to body elasticity. Input matrix B maps the external torque into generalized forces of the system.

The equations of motion explicitly written in rigid and elastic coordinates are

$$\begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix} \begin{bmatrix} \ddot{q}_r \\ \ddot{q}_f \end{bmatrix} + \begin{bmatrix} C_{rr} & C_{rf} \\ C_{fr} & C_{ff} \end{bmatrix} \begin{bmatrix} \dot{q}_r \\ \dot{q}_f \end{bmatrix} + \begin{bmatrix} D_{rr} & 0 \\ 0 & D_{ff} \end{bmatrix} \begin{bmatrix} \dot{q}_r \\ \dot{q}_f \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_{ff} \end{bmatrix} \begin{bmatrix} q_r \\ q_f \end{bmatrix} = \begin{bmatrix} B_r \\ B_f \end{bmatrix} \tau \tag{3.2}$$

The actuators are assumed to be placed at manipulator joints. Thus, the input matrix B is expressed as $B_r = I_{nXn}$ and $B_f = 0_{mXn}$. The model inversion of equation [3.2], that maps input torque and desired output trajectory, depends on the rigid and elastic coordinates of the system. If the desired output trajectory is the tip trajectory, the system is unstable due to non-minimum phase nature.

The following model based controllers are designed for the trajectory tracking.

1. PD Control
2. Stable Inversion Control
3. A Nonlinear Control
4. Adaptive Control

Among them, PD control and Stable inversion control are derived using feedback linearization technique. A nonlinear control and Adaptive control are derived using sliding mode technique.

3.1 PD Control

The control architecture of PD type control is shown in Figure [3.1]. It consists of feedforward compensator and a PD feedback loop.

Using equation [3.2], the feedforward compensator is defined as

$$\tau = M_{rr}\ddot{q}_r + M_{rf}\ddot{q}_f + C_{rr}\dot{q}_r + C_{rf}\dot{q}_f + D_{rr}\dot{q}_r \tag{3.3}$$

The coupling effect of rigid body motion and elastic deformations of flexible link in equation [3.3] is ignored to study the effect of flexibility in control design. The equation [3.3] is rewritten as

$$\tau = M_{rr}\ddot{q}_r + C_{rr}\dot{q}_r + D_{rr}\dot{q}_r \tag{3.4}$$

A PD type feedback control at joint space is designed to ensure the stability for unmodeled dynamics. It is written as

$$\tau = M_{rr}(\ddot{q}_r + K_p e_r + K_v \dot{e}_r) + C_{rr}\dot{q}_r + D_{rr}\dot{q}_r \tag{3.5}$$

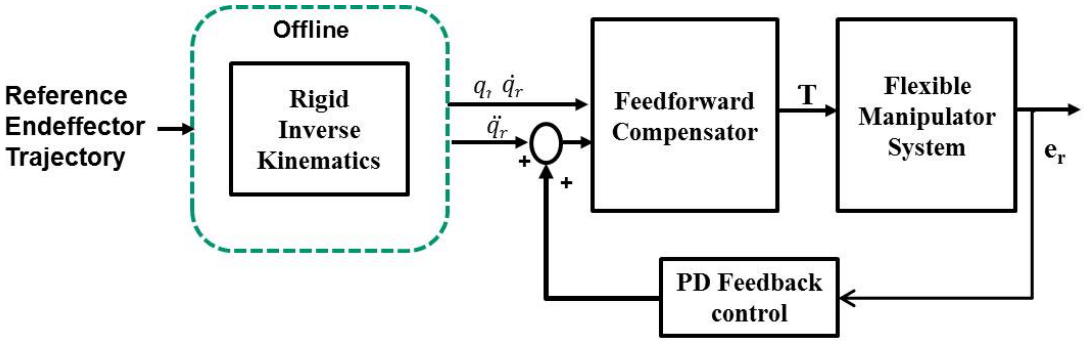

Figure 3.1: *PD control architecture.*

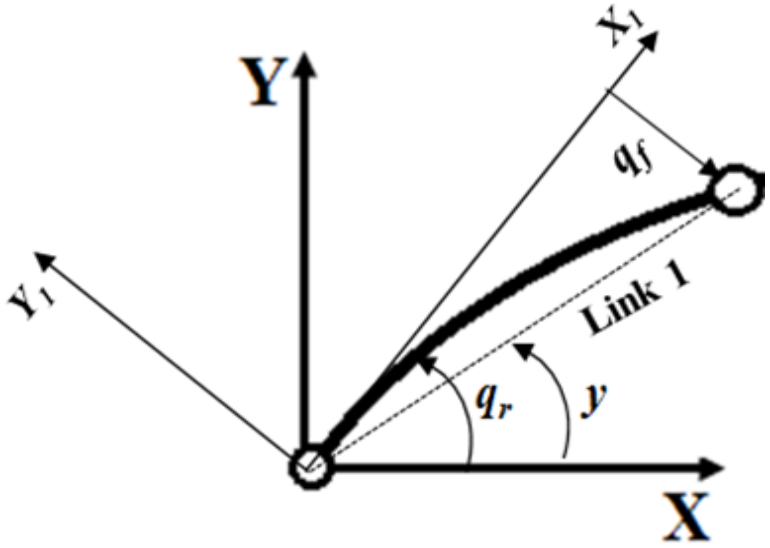

Figure 3.2: *Representation of flexible link tip position at the joint space.*

where, K_p and K_v are the position and velocity error gain, respectively; e_r and $\dot{e}_r$ are the joint trajectory error at position and velocity level, respectively.

The equation [3.5], is exactly similar to computed torque control of rigid link manipulator. The objective of this PD control implementation is to analyse the contribution of link flexibility in control design.

3.2 Singular Perturbation Control

The control architecture of stable inversion control is shown in Figure [3.3]. It consists of a feedforward compensator and a robust feedback control to achieve desired tip trajectory tracking. Feedforward compensator is derived with the help of stable inversion technique. Stable inversion technique solves the non-minimum phase system by pre-computing bounded internal states q_f for the tip position $y(t)$. The representation of link tip position is shown in Figure [3.2].

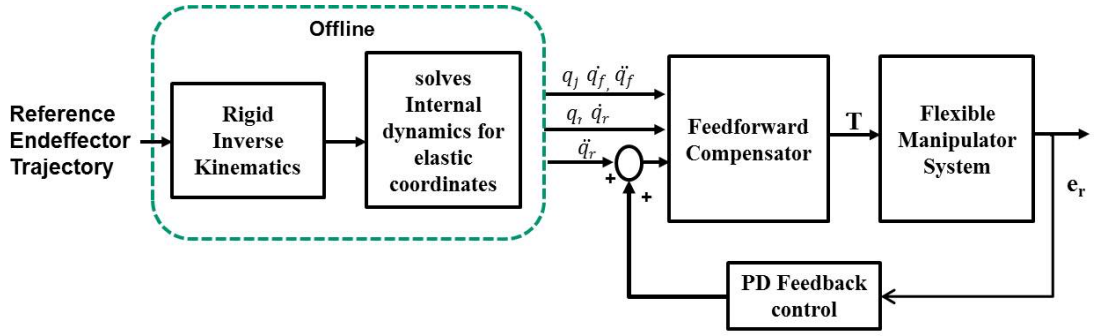

Figure 3.3: *Stable inversion control architecture.*

The actual tip position is defined as a nonlinear function in terms of joint rotation q_r and elastic coordinates q_f.

$$y = q_r + \arctan\left(\frac{q_f}{L}\right) \tag{3.6}$$

Where y, q_r, and q_f are respectively the tip position, joint angle, and link elastic displacement. To simplify the coupling between tip position and link elastic displacement, the actual tip position is approximated using joint angle and elastic displacement by a weighted parameter. Using weighted parameter, the tip position is redefined as a linear combination of joint angle and link tip elastic coordinates

$$y \approx q_r + \left(\frac{q_f}{L}\right) = q_r + \Gamma q_f \tag{3.7}$$

where Γ is the weighted parameter equal to the reciprocal of link length.

In order to compute bounded internal states, the internal dynamics of the system is rewritten in terms of tip position y and elastic coordinates q_f using equation [3.2]

$$M_{fr}(\ddot{y} - \Gamma\ddot{q}_f) + M_{ff}\ddot{q}_f + C_{fr}(\dot{y} - \Gamma\dot{q}_f) + C_{ff}\dot{q}_f + D_{ff}\dot{q}_f + K_{ff}q_f = 0 \tag{3.8}$$

Iterative Learning Method

Internal dynamics of the system is solved using simple PD type Iterative learning method. This method is applied on the nominal dynamic model of the system, since dynamics of the unmodeled payload mass is unknown prior to consider in the dynamic model.

Learning process for the internal dynamics of the system is expressed in terms of deformation torque error,

$$\begin{aligned} e_d(q_f, \dot{q}_f, \ddot{q}_f) = M_{fr}(\ddot{y} - \Gamma\ddot{q}_f) + M_{ff}\ddot{q}_f + C_{fr}(\dot{y} - \Gamma\dot{q}_f) + C_{ff}\dot{q}_f \\ + D_{ff}\dot{q}_f + K_{ff}q_f \end{aligned} \tag{3.9}$$

The bounded elastic coordinates in time domain for the given reference trajectory $y(t)$ is calculated as follows [8]

1. Set initial values $q_f^{(0)}, \dot{q}_f^{(0)}, \ddot{q}_f^{(0)}$ to zero along the trajectory. Set $i = 0$

2. Using equation [3.9], compute $e_d^{(i)}(q_f^{(i)}, \dot{q}_f^{(i)}, \ddot{q}_f^{(i)})$
   If $\|e_d^{(i)}\| < \epsilon_e$, where ϵ_e is error tolerance,
   set $q_{fd} = q_f^{(i)}$ and stop. Else process error using finite impulse response [FIR] filter; $e^{(i)}$ and $\dot{e}^{(i)}$ are thus obtained

3. Update the $q_f^{(i+1)}$ using simple PD type learning rule

$$q_f^{(i+1)} = q_f^{(i)} - K_{LP} e_d^{(i)} - K_{LD} \dot{e}_d^{(i)} \tag{3.10}$$

   with small gains K_{LP} and K_{LD}. Set $i = i + 1$ and go to step 2.

Feedforward Compensator

Feed forward compensator is derived using a stable inversion technique. The bounded elastic states q_f, $\dot{q_f}$, $\ddot{q_f}$ are pre-computed using iterative learning method. Thus, the model inversion is stable for the link tip position $y(t)$. The torque required to drive the system along the tip trajectory is computed using

$$\tau = M_{rr}\ddot{q_r} + M_{rf}\ddot{q_f} + C_{rr}\dot{q_r} + C_{rf}\dot{q_f} + D_{rr}\dot{q_r} \tag{3.11}$$

However, the computed torque shown in equation [3.11] works well for the exact model. A simple PD feedback control with arbitrary gains provides accurate trajectory tracking. The closed loop computed torque control is written as

$$\tau = M_{rr}(\ddot{q_r} + K_p e_r + K_v \dot{e_r}) + M_{rf}\ddot{q_f} + C_{rr}\dot{q_r} + C_{rf}\dot{q_f} + D_{rr}\dot{q_r} \tag{3.12}$$

where, K_p and K_v are the position and velocity error gain, respectively. e_r is the joint trajectory error i.e.

$$e_r = q_{rd} - q_r \tag{3.13}$$

where q_{rd} is the desired joint trajectory, computed using the bounded elastic coordinates q_f and the tip trajectory $y(t)$ from the Iterative learning method. To account for model uncertainties and unknown payload mass M_p, a robust feedback loop is designed based on conventional rigid manipulator dynamics.

Robust feedback control

Lyapunov function is used to design feedback gains to guarantee the stability along the trajectory with uncertainty in the model. Consider a nominal model

$$\overline{M}\ddot{q} + \overline{C}(q, \dot{q})\dot{q} + \overline{D}(q) = B\tau \tag{3.14}$$

A feedback linearization to a nominal model gives the tracking error dynamics for the joint variable q_r as

$$\begin{bmatrix} \dot{e_r} \\ \ddot{e_r} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_r \\ \dot{e_r} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \tag{3.15}$$

Due to the uncertainties present in $M(q)$ and $C(q, \dot{q})$, the tracking error dynamics have an additional nonlinear term η, which is nonlinear function of both e_r and u.

$$\dot{e_r} = Ae_r + B(u + \eta) \tag{3.16}$$

$$\eta = \Delta(u - \ddot{q_r}) + M^{-1}\delta \tag{3.17}$$

where

$$\Delta \ = \ M^{-1}\overline{M} - I_n \tag{3.18}$$

and

$$\delta = C - \overline{C} \tag{3.19}$$

To derive the stability conditions the following assumptions are made with finite constants defining the size of uncertainty [27]

$$\frac{1}{\mu_2} \ \leq \| \ M^{-1} \ \| \leq \ \frac{1}{\mu_1} \tag{3.20}$$

$$\| \ \Delta \ \| \leq \ a \leq \ 1 \tag{3.21}$$

$$\| \ \delta \ \| \leq \ \beta_0 + \beta_1 \| \ e_r \ \| + \beta_2 \| \ e_r \ \|^2 \tag{3.22}$$

$$\| \ \ddot{q_{rd}} \ \| \leq \ c \tag{3.23}$$

Consider a feedback controller

$$u = -Ke_r \tag{3.24}$$

such that

$$\dot{e_r} \ = \ Ae_r + B(u + \eta) \ = \ (A - BK)e_r + B\eta \ = \ A_c + B\eta \tag{3.25}$$

By placing the poles far from the left-half of the plane, the stability of closed loop system in the presence of η is guaranteed. Solving Lyapunov equation

$$A_c{}^T P + PA_c = -Q \tag{3.26}$$

with the choice of

$$Q = \begin{bmatrix} 2K_p^2 & 0 \\ 0 & 2K_v^2 - 2K_p \end{bmatrix} \tag{3.27}$$

and

$$K_v^2 > K_p \tag{3.28}$$

the positive definite solution of equation [3.26] is written as

$$P = \begin{bmatrix} 2K_pK_v & K_p \\ K_p & K_v \end{bmatrix} \tag{3.29}$$

and feedback gains is defined as

$$K = B^T P = \begin{bmatrix} \frac{K_v{}^2}{a} & K_v \end{bmatrix} \tag{3.30}$$

The closed loop system equation [3.25] is uniformly bounded [27] if $e_r(0) = 0, \dot{e_r}(0) = 0$ and

$$a > 1 + \frac{1}{\mu_1}[\beta_0 + 2(\beta_2\beta_0 + \beta_2(\mu_1 + \mu_2)c)^{\frac{1}{2}}] \tag{3.31}$$

where

$$K_v = 2aI \tag{3.32}$$

and

$$K_p = 4aI \tag{3.33}$$

where K_p and K_v are position and velocity error gains, respectively. I is a $n \times n$ identity matrix. In this way, for the positive K_p and K_v gains the closed loop system is asymptotically stable.

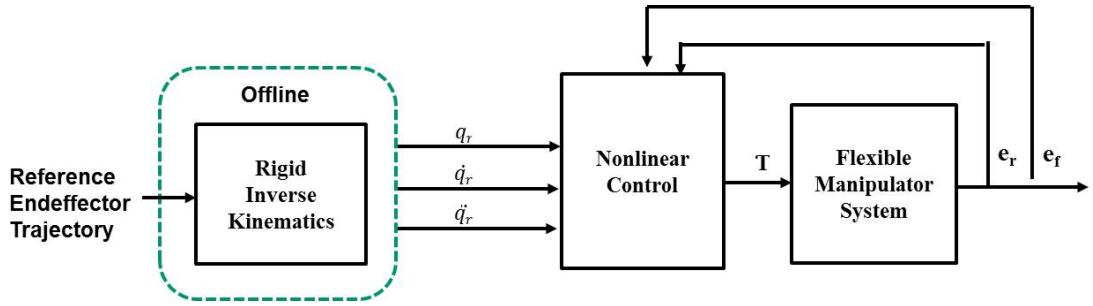

Figure 3.4: *A nonlinear control architecture.*

3.3 A Nonlinear Control

The control architecture of a nonlinear control is shown in Figure [3.4]. The Lyapunov function is used to show the asymptotic stability of closed loop system.

Consider the dynamic model

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D\dot{q} + Kq = B\tau \tag{3.34}$$

Position error along the trajectory is defined as

$$e = \begin{bmatrix} e_r \\ e_f \end{bmatrix} = \begin{bmatrix} q_{rd} - q_r \\ q_{fd} - q_f \end{bmatrix} \tag{3.35}$$

where q_{rd} and q_{fd} are the desired rigid and flexible coordinates. q_{fd} is set to zero to suppress vibrations.

$$e = \begin{bmatrix} q_{rd} - q_r \\ -q_f \end{bmatrix} \tag{3.36}$$

Lets define the sliding surface s as

$$s = \dot{e} + \lambda e = \begin{bmatrix} \dot{e_r} + \lambda_r e_r \\ \dot{e_f} + \lambda_f e_f \end{bmatrix} \tag{3.37}$$

where

$$\lambda = \begin{bmatrix} \lambda_r & 0 \\ 0 & \lambda_f \end{bmatrix} \tag{3.38}$$

The error dynamics of the system with newly defined signal S_r and S_f can be derived as

$$\begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix} \begin{bmatrix} \dot{s_r} \\ \dot{s_f} \end{bmatrix} + \begin{bmatrix} C_{rr} & C_{rf} \\ C_{fr} & C_{ff} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} + \begin{bmatrix} D_{rr} & 0 \\ 0 & D_{ff} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} + \begin{bmatrix} K_{vr} & 0 \\ 0 & K_{vf} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} = \begin{bmatrix} \tau_m + K_{vr}s_r - \tau \\ \tau_a \end{bmatrix} \tag{3.39}$$

where

$$\begin{aligned} \tau_m = {} & M_{rr}(\ddot{q}_{rd} + \lambda_r \dot{e_r}) + M_{rf}(-\lambda_f \dot{e_f}) + C_{rr}(\dot{q}_{rd} + \lambda_r e_r) \\ & + C_{rf}(-\lambda_f e_f) + D_{rr}(\dot{q}_{rd} + \lambda_r e_r) \end{aligned} \tag{3.40}$$

$$\begin{aligned}\tau_a = M_{fr}(\ddot{q}_{rd} + \lambda_r \dot{e_r}) + M_{ff}(-\lambda_f \dot{e_f}) + C_{fr}(\dot{q}_{rd} + \lambda_r e_r) \\ + C_{ff}(-\lambda_f e_f) + D_{ff}(-\lambda_f e_f) + K_{ff} e_f + K_{vf} s_f\end{aligned} \tag{3.41}$$

The following function is considered [37] to analyze the asymptotic stability of error dynamics shown in equation (3.39).

$$\dot{Y} = \begin{cases} 2(\sqrt{Y}a(t) + b(t)) & Y(t) > 0 \\ 2b(t) & Y(t) = 0 \quad b(t) > 0 \\ \delta & Y(t) = 0 \quad b(t) \leq 0 \end{cases} \tag{3.42}$$

where

$$a(t) = \frac{\|s_r\|^2}{\|s_r\|^2 + \epsilon}(s_f^T \tau_a) \tag{3.43}$$

$$b(t) = \frac{-\epsilon}{\|s_r\|^2 + \epsilon}(s_f^T \tau_a) \tag{3.44}$$

and δ is a small positive constant. The above function $Y(t) \geq 0$ for all $t \geq 0$. Let $k = \sqrt{Y}$ or $k^2 = Y$, then $k(t)$ can be checked to satisfy differential equation

$$\dot{k} = \frac{1}{k}\left(\frac{k\|s_r\|^2 - \epsilon}{\|s_r\|^2 + \epsilon}\right)(s_f^T \tau_a), \quad k \neq 0 \tag{3.45}$$

Equation [3.45] is a differential equation to check the variable $Y(t)$ is always positive and $k(t)$ is defined to be its root.

The control law with parameter estimates is chosen as

$$\tau = \tau_m + K_{vr} s_r + \tau_f \tag{3.46}$$

where

$$\tau_f = \frac{(1+k)s_r}{\|s_r\|^2 + \epsilon}(s_f^T \tau_a) \tag{3.47}$$

With this proposed controller the error dynamics of the system is obtained as

$$\begin{aligned}\begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix}\begin{bmatrix} \dot{s_r} \\ \dot{s_f} \end{bmatrix} + \begin{bmatrix} C_{rr} & C_{rf} \\ C_{fr} & C_{ff} \end{bmatrix}\begin{bmatrix} s_r \\ s_f \end{bmatrix} + \begin{bmatrix} D_{rr} & 0 \\ 0 & D_{ff} \end{bmatrix}\begin{bmatrix} s_r \\ s_f \end{bmatrix} \\ + \begin{bmatrix} K_{vr} & 0 \\ 0 & K_{vf} \end{bmatrix}\begin{bmatrix} s_r \\ s_f \end{bmatrix} = \begin{bmatrix} -\tau_f \\ \tau_a \end{bmatrix}\end{aligned} \tag{3.48}$$

The error dynamics written in compact form is

$$M(q)\dot{s} + C(q,\dot{q})s + Ds + K_v s = \begin{bmatrix} -\tau_f \\ \tau_a \end{bmatrix} \tag{3.49}$$

The asymptotic stability of the error dynamics shown in equation [**??**] is analyzed using Lyapunov function [37]

$$\begin{aligned} V &= \frac{1}{2}s^T M(q)s + \frac{1}{2}Y \\ &= \frac{1}{2}s^T M(q)s + \frac{1}{2}k^2 \end{aligned} \tag{3.50}$$

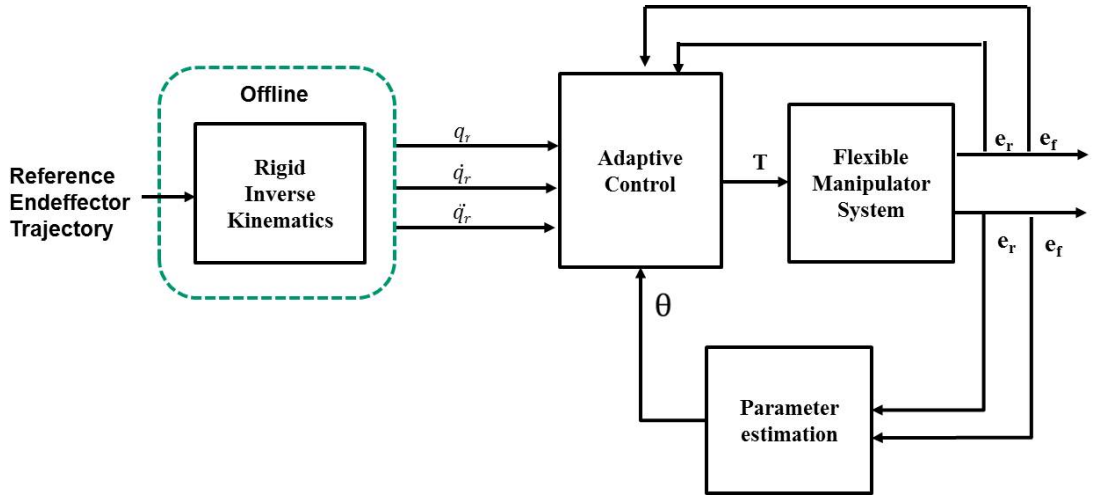

Figure 3.5: *Adaptive control architecture.*

Differentiating equation (3.50) with respect to time yield

$$\begin{aligned}
\dot{V} &= s^T M(q)\dot{s} + \frac{1}{2}s^T\dot{M}(q)s + k\dot{k} \\
&= s^T\left(-Cs - Ds - K_v s + \begin{bmatrix} -\tau_f \\ \tau_a \end{bmatrix}\right) \\
&+ \frac{1}{2}s^T\dot{M}(q)s + \left(\frac{k\|s_r\|^2 - \epsilon}{\|s_r\|^2 + \epsilon}\right)(s_f^T\tau_a) \\
&= -s^T Ds - s^T K_v s + \left(\frac{k\|s_r\|^2 - \epsilon}{\|s_r\|^2 + \epsilon}\right)(s_f^T\tau_a) \\
&- \left(\frac{(1+k)\|s_r\|^2 - \|s_r\|^2 - \epsilon}{\|s_r\|^2 + \epsilon}\right)(s_f^T\tau_a) \\
&= -s^T DS - s^T K_v s \quad for \quad k > 0
\end{aligned} \tag{3.51}$$

Here K_v and D are positive definite, which is clear as $\dot{V} < 0$. Thus, the system is asymptotically stable whenever $k > 0$ and $K_v > 0$.

3.4 Adaptive Control

The control architecture of adaptive control is shown in Figure [3.5]. The Lyapunov function is used to show the asymptotic stability of closed loop system.

Consider a nominal model

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D\dot{q} + Kq = B\tau \tag{3.52}$$

Position error along the trajectory is defined as

$$e = \begin{bmatrix} e_r \\ e_f \end{bmatrix} = \begin{bmatrix} q_{rd} - q_r \\ q_{fd} - q_f \end{bmatrix} \tag{3.53}$$

where q_{rd} and q_{fd} are the desired rigid and flexible coordinates. q_{fd} is set to zeros to suppress vibrations.

$$e = \begin{bmatrix} q_{rd} - q_r \\ -q_f \end{bmatrix} \tag{3.54}$$

Lets define the sliding surface s as

$$s = \dot{e} + \lambda e = \begin{bmatrix} \dot{e_r} + \lambda_r e_r \\ \dot{e_f} + \lambda_f e_f \end{bmatrix} \tag{3.55}$$

where

$$\lambda = \begin{bmatrix} \lambda_r & 0 \\ 0 & \lambda_f \end{bmatrix} \tag{3.56}$$

The error dynamics of the system with newly defined signal S_r and S_f can be derived as

$$\begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix} \begin{bmatrix} \dot{s_r} \\ \dot{s_f} \end{bmatrix} + \begin{bmatrix} C_{rr} & C_{rf} \\ C_{fr} & C_{ff} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} + \begin{bmatrix} D_{rr} & 0 \\ 0 & D_{ff} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} + \begin{bmatrix} K_{vr} & 0 \\ 0 & K_{vf} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} = \begin{bmatrix} \tau_m + K_{vr} s_r - \tau \\ \tau_a \end{bmatrix} \tag{3.57}$$

where

$$\begin{aligned} \tau_m = {} & M_{rr}(\ddot{q}_{rd} + \lambda_r \dot{e_r}) + M_{rf}(-\lambda_f \dot{e_f}) + C_{rr}(\dot{q}_{rd} + \lambda_r e_r) \\ & + C_{rf}(-\lambda_f e_f) + D_{rr}(\dot{q}_{rd} + \lambda_r e_r) \end{aligned} \tag{3.58}$$

$$\begin{aligned} \tau_a = {} & M_{fr}(\ddot{q}_{rd} + \lambda_r \dot{e_r}) + M_{ff}(-\lambda_f \dot{e_f}) + C_{fr}(\dot{q}_{rd} + \lambda_r e_r) \\ & + C_{ff}(-\lambda_f e_f) + D_{ff}(-\lambda_f e_f) + K_{ff} e_f + K_{vf} s_f \end{aligned} \tag{3.59}$$

The dynamics of the flexible manipulator is expressed in terms of linear type parametric model as

$$\begin{aligned} W_1 \Theta_1 = {} & M_{rr}(\ddot{q}_{rd} + \lambda_r \dot{e_r}) + M_{rf}(-\lambda_f \dot{e_f}) \\ & + C_{rr}(\dot{q}_{rd} + \lambda_r e_r) + C_{rf}(-\lambda_f e_f) + D_{rr}(\dot{q}_{rd} + \lambda_r e_r) \end{aligned} \tag{3.60}$$

$$\begin{aligned} W_2 \Theta_2 = {} & M_{fr}(\ddot{q}_{rd} + \lambda_r \dot{e_r}) + M_{ff}(-\lambda_f \dot{e_f}) + C_{fr}(\dot{q}_{rd} + \lambda_r e_r) \\ & + C_{ff}(-\lambda_f e_f) + D_{ff}(-\lambda_f e_f) + K_{ff} e_f + K_{vf} s_f \end{aligned} \tag{3.61}$$

where W_1 and W_2 are $n \times r_1$, $m \times r_2$ regression matrix for appropriate $r_1, r_2 > 0$; and Θ_1, Θ_2 are unknown constant parameters.

To design adaptive controller for the error dynamics show in equation [3.57], the following function is considered [37]

$$\dot{Y} = \begin{cases} 2(\sqrt{Y} a(t) + b(t)) & Y(t) > 0 \\ 2b(t) & Y(t) = 0 \quad b(t) > 0 \\ \delta & Y(t) = 0 \quad b(t) \leq 0 \end{cases} \tag{3.62}$$

where

$$a(t) = \frac{\|s_r\|^2}{\|s_r\|^2 + \epsilon} (s_f^T W_2 \hat{\Theta}_2 + s_f^T K_{vf} s_f) \tag{3.63}$$

$$b(t) = \frac{-\epsilon}{\|s_r\|^2 + \epsilon}(s_f^T W_2 \hat{\Theta}_2 + s_f^T K_{vf} s_f) \tag{3.64}$$

and δ is a small positive constant. The above function $Y(t) \geq 0$ for all $t \geq 0$. Let $k = \sqrt{Y}$ or $k^2 = Y$, then $k(t)$ can be checked to satisfy differential equation

$$\dot{k} = \frac{1}{k}\left(\frac{k\|s_r\|^2 - \epsilon}{\|s_r\|^2 + \epsilon}\right)(s_f^T W_2 \hat{\Theta}_2 + s_f^T K_{vf} s_f), \quad k \neq 0 \tag{3.65}$$

Equation [3.65] is a differential equation to check the variable $Y(t)$ is always positive and $k(t)$ is defined to be its root.

The control law with parameter estimates is chosen as

$$\tau = W_1 \hat{\Theta}_1 + K_{vr} s_r + \tau_f \tag{3.66}$$

where

$$\tau_f = \frac{(1+k)s_r}{\|s_r\|^2 + \epsilon}(s_f^T W_2 \hat{\Theta}_2 + s_f^T K_{vf} s_f) \tag{3.67}$$

In which, $\hat{\Theta}_1$ and $\hat{\Theta}_2$ are the estimates of Θ_1 Θ_2 respectively.

$$\begin{aligned} W_1\hat{\Theta}_1 = {} & \hat{M}_{rr}(\ddot{q}_{rd} + \lambda_r \dot{e_r}) + \hat{M}_{rf}(-\lambda_f \dot{e_f}) \\ & + \hat{C}_{rr}(\dot{q}_{rd} + \lambda_r e_r) + \hat{C}_{rf}(-\lambda_f e_f) + \hat{D}_{rr}(\dot{q}_{rd} + \lambda_r e_r) \end{aligned} \tag{3.68}$$

$$\begin{aligned} W_2\hat{\Theta}_2 = {} & \hat{M}_{fr}(\ddot{q}_{rd} + \lambda_r \dot{e_r}) + \hat{M}_{ff}(-\lambda_f \dot{e_f}) \\ & + \hat{C}_{fr}(\dot{q}_{rd} + \lambda_r e_r) + \hat{C}_{ff}(-\lambda_f e_f) + \hat{D}_{ff}(-\lambda_f e_f) + \hat{K}_{ff} e_f \end{aligned} \tag{3.69}$$

With this proposed controller the error dynamics of the system is obtained as

$$\begin{aligned} & \begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix} \begin{bmatrix} \dot{s_r} \\ \dot{s_f} \end{bmatrix} + \begin{bmatrix} C_{rr} & C_{rf} \\ C_{fr} & C_{ff} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} + \begin{bmatrix} D_{rr} & 0 \\ 0 & D_{ff} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} \\ & + \begin{bmatrix} K_{vr} & 0 \\ 0 & K_{vf} \end{bmatrix} \begin{bmatrix} s_r \\ s_f \end{bmatrix} = \begin{bmatrix} W_1\tilde{\Theta}_1 - \tau_f \\ W_2\tilde{\Theta}_2 \end{bmatrix} \end{aligned} \tag{3.70}$$

The error dynamics written in compact form is

$$M(q)\dot{s} + C(q,\dot{q})s + Ds + K_v s = \begin{bmatrix} W_1\tilde{\Theta}_1 - \tau_f \\ W_2\tilde{\Theta}_2 \end{bmatrix} \tag{3.71}$$

where $\tilde{\Theta}_1 = \Theta_1 - \hat{\Theta}_1$ and $\tilde{\Theta}_2 = \Theta_2 - \hat{\Theta}_2$ are the parameter estimation error.

The adaptation law is derived as

$$\dot{\hat{\Theta}}_1 = -K_1 W_1^T s_r \tag{3.72}$$

$$\dot{\hat{\Theta}}_2 = -K_2 W_2^T s_f \tag{3.73}$$

where K_1, K_2 are positive definite matrices. The asymptotic stability of the error dynamics is derived using Lyapunov function

$$\begin{aligned} V &= \frac{1}{2}s^T M(q)s + \frac{1}{2}\tilde{\Theta}_1^T K_1^{-1}\tilde{\Theta}_1 + \frac{1}{2}\tilde{\Theta}_2^T K_2^{-1}\tilde{\Theta}_2 + \frac{1}{2}Y \\ &= \frac{1}{2}s^T M(q)s + \frac{1}{2}\tilde{\Theta}_1^T K_1^{-1}\tilde{\Theta}_1 + \frac{1}{2}\tilde{\Theta}_2^T K_2^{-1}\tilde{\Theta}_2 + \frac{1}{2}k^2 \end{aligned} \tag{3.74}$$

Differentiating equation [3.74] with respect to time yields

$$\begin{aligned}
\dot{V} &= s^T M(q)\dot{s} + \frac{1}{2}s^T\dot{M}(q)s + \dot{\tilde{\Theta}}_1^T K_1^{-1}\tilde{\Theta}_1 + \dot{\tilde{\Theta}}_2^T K_2^{-1}\tilde{\Theta}_2 + k\dot{k} \\
&= s^T\left(-Cs - Ds - K_v s + \begin{bmatrix} W_1\tilde{\Theta}_1 - \tau_f \\ W_2\tilde{\Theta}_2 \end{bmatrix}\right) \\
&+ \frac{1}{2}s^T\dot{M}(q)s - s_r^T W_1\tilde{\Theta}_1 - s_f^T W_2\tilde{\Theta}_2 \\
&+ \left(\frac{k\|s_r\|^2 - \epsilon}{\|s_r\|^2 + \epsilon}\right)(s_f^T W_2\hat{\Theta}_2 + s_f^T K_{vf} s_f) \\
&= -s^T Ds - s^T K_v s + \left(\frac{k\|s_r\|^2 - \epsilon}{\|s_r\|^2 + \epsilon}\right)(s_f^T W_2\hat{\Theta}_2 \\
&- \left(\frac{(1+k)\|s_r\|^2 - \|s_r\|^2 - \epsilon}{\|s_r\|^2 + \epsilon}\right)(s_f^T W_2\hat{\Theta}_2 + s_f^T K_{vf} s_f) + s_f^T W_2\hat{\Theta}_2 \\
&- s_f^T W_2\hat{\Theta}_2 \\
&= -s^T DS - s^T K_v s \quad for \quad k > 0
\end{aligned} \tag{3.75}$$

Here K_v and D are positive definite, which is clear as $\dot{V} < 0$. Thus, the system is asymptotically stable whenever $k > 0$ and $K_v > 0$.

3.5 Simulation Results

A spatial RRR manipulator shown in Figure [3.6], is considered to demonstrate the performance of model based controllers. The manipulator have three flexible links and three rigid revolute joints. Each flexible link is discretized using two finite element beams with six degrees of freedom on each node and one degree of freedom for rigid body rotations i.e. θ_i where $i = 1, 2, 3$. The damping effects on links and joints are ignored. The physical parameters of a RRR spatial manipulator are presented in Table [4.1]. Uniform cross-section and material properties are assumed on each link.

Table 3.1: *The physical parameters of a RRR flexible manipulator.*

Parameter	Link 1	Link 2	Link 3
Link Length (m)	1	4.0	3.5
C/s Area (m^2)	0.028	0.0020	0.0008
Moment of Inertia (m^4)	8.33×10^{-7}	6.24×10^{-7}	5.37×10^{-7}
Polar moment of Inertia (m^4)	1.66×10^{-6}	1.24×10^{-6}	1.07×10^{-6}
Tensile Modulus (MPa)	206000		
Shear Modulus (MPa)	79300		
Density (Kg/m^3)	8253		

The reference trajectories of joint 1, joint 2, and joint 3 are shown in Figure [3.7]. The simulation results of PD controller, Stable inversion controller, nonlinear controller, and Adaptive controller are compared. Figure [3.8], Figure [3.9], and Figure [3.10] show the error along joint trajectory θ_1, θ_1, and θ_1 respectively. Figure [3.11],

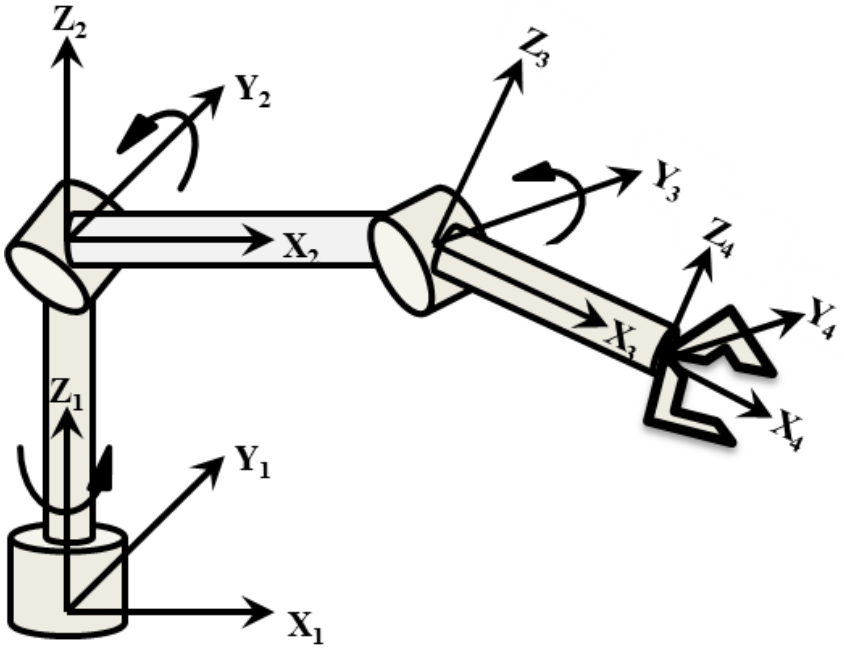

Figure 3.6: *Spatial RRR flexible manipulator with three flexible links and three rigid joints.*

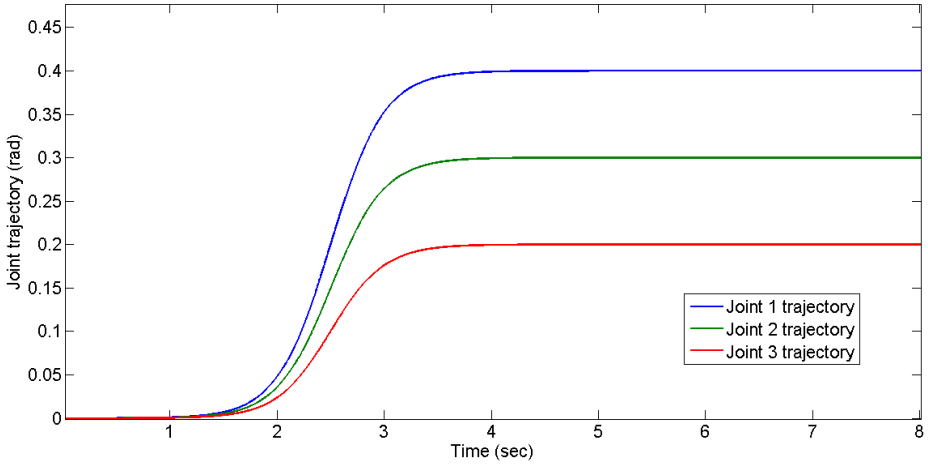

Figure 3.7: *Reference trajectory.*

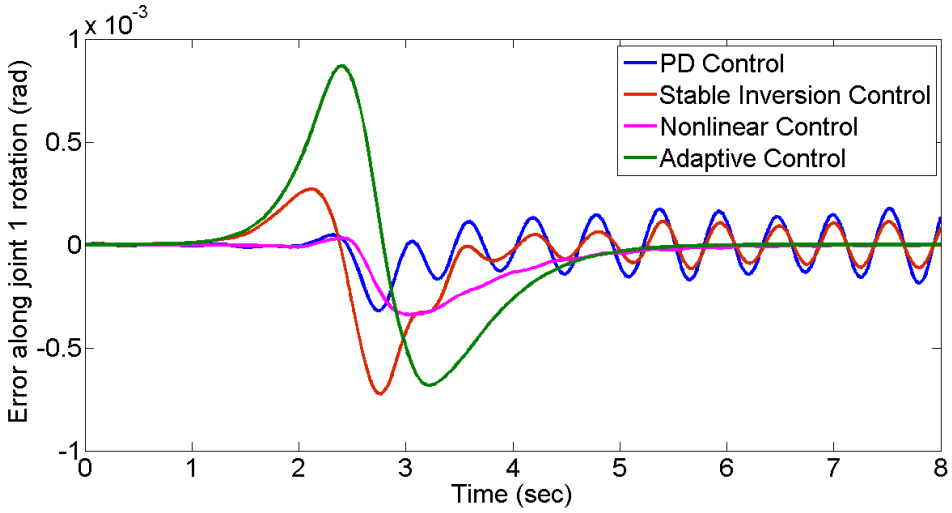

Figure 3.8: *Error along joint θ_1 trajectory.*

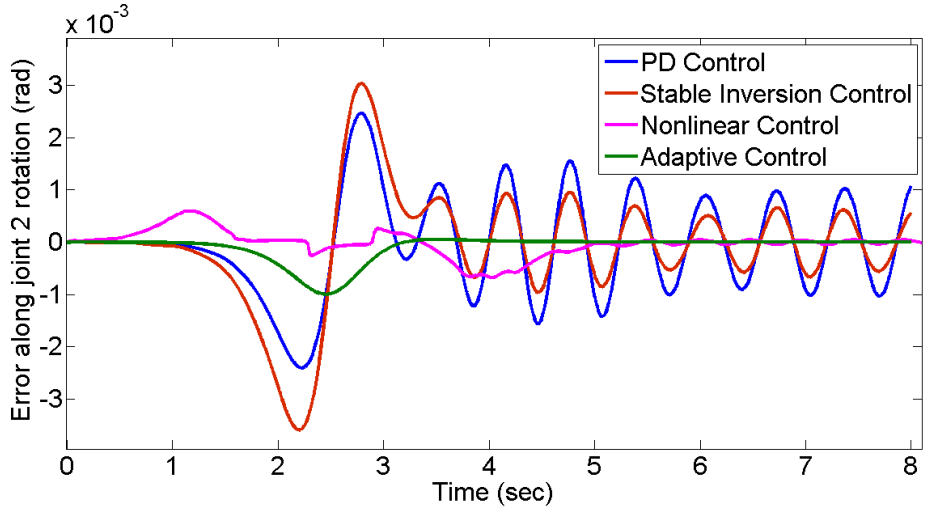

Figure 3.9: *Error along joint θ_2 trajectory.*

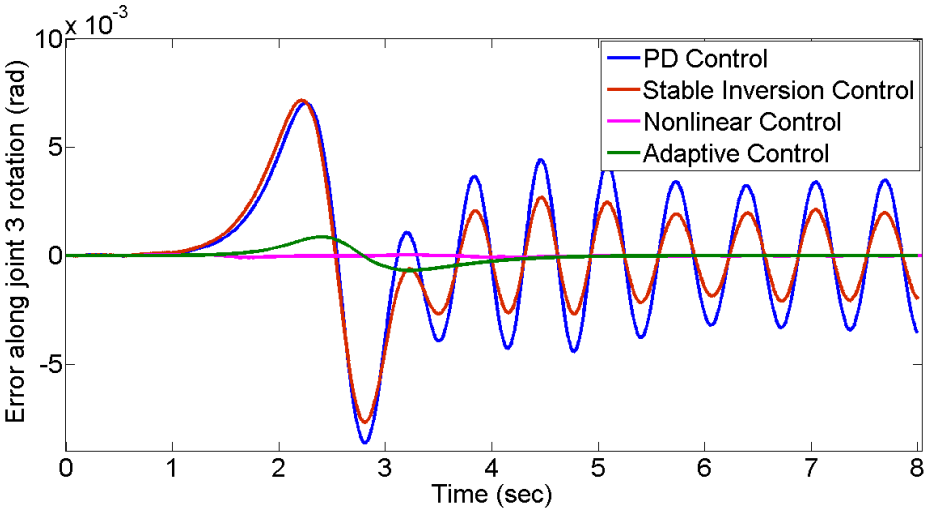

Figure 3.10: *Error along joint θ_3 trajectory.*

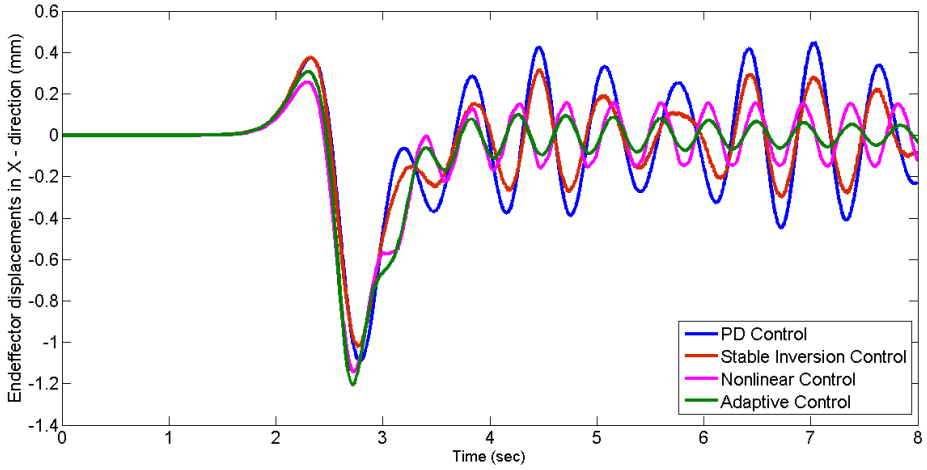

Figure 3.11: *Elastic displacements of endeffector $X_4Y_4Z_4$ along X-Direction.*

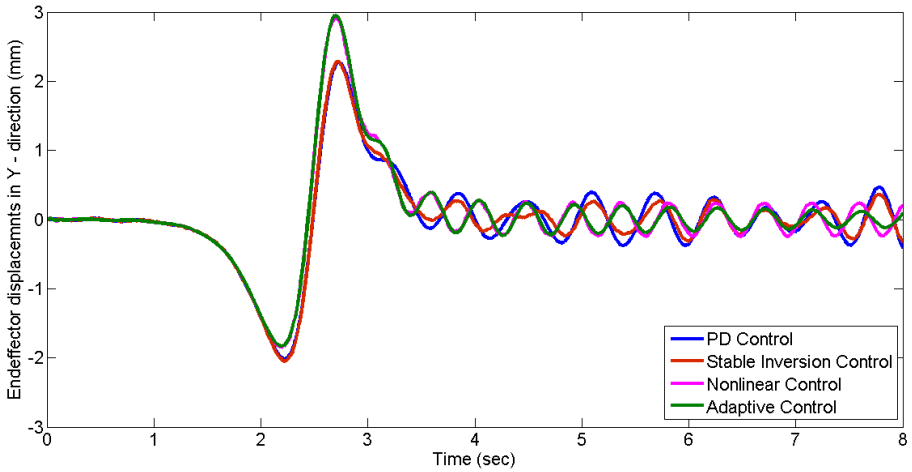

Figure 3.12: *Elastic displacements of endeffector $X_4Y_4Z_4$ along Y-Direction.*

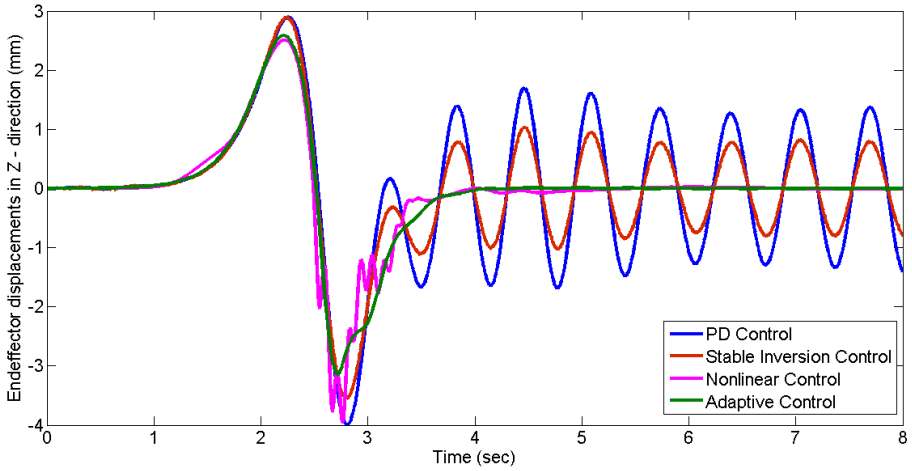

Figure 3.13: *Elastic displacements of endeffector $X_4Y_4Z_4$ along Z-Direction.*

Figure [3.12], and Figure [3.13] show the endeffector $X_4Y_4Z_4$ elastic displacements along X, Y, and Z direction respectively.

The PD controller is designed based on rigid link manipulator dynamics to study the effect of link flexibility in control design. Thus, the simulation results of PD controller showed good trajectory tracking at joint space but it is not very efficient to damp the endeffector vibrations. It can be seen in Figure [3.11] - [3.13].

The stable inversion control showed better trajectory tracking compared to PD control, however this method is incapable of damping the vibrations in case of unmodeled dynamics such as unknown payload mass. Moreover, stable inversion control have good performance for planar flexible link manipulators compared to spatial flexible link manipulators. These results are shown in chapter [4].

Nonlinear control and adaptive control showed good trajectory tracking at the joint space and also efficient to damp the endeffector $X_4Y_4Z_4$ vibrations compared to PD controller and stable inversion control as it can be seen in Figure [3.11] - Figure [3.13]. The endeffector vibrations along Z-direction is damped quickly where as the endeffector vibrations along X-direction and Y-direction are taking long time to damp the vibrations because these vibrations are out-of-plane with respect to actuators at the manipulator joints. To improve the damping properties of the out-of-plane bending vibration additional control effort is required along this direction.

Overall, the adaptive control showed better performance compared to PD controller, stable inversion control and nonlinear control.

CHAPTER 4

Experimental Results

4.1 Introduction

A single link flexible manipulator shown in Figure [4.1] is designed to demonstrate the performance of the model based controllers that are developed for the trajectory tracking. The experimental setup consists of Quanser SRV02 rotary servo plant, Quanser Q8 terminal board, universal power module (UPM), single link flexible manipulator, and a strain guage. The schematic layout of experimental setup is shown in Figure [4.3].

A strain gauge is mounted close to clamped end of flexible link to measure tip displacements. It is shown in Figure [4.2]. The physical parameters of single link manipulator are presented in Table [4.1].

The Quanser SRV02 rotary servo plant consists of actuator and external load gear. The actuator consists of DC motor equipped with internal planetary gearbox. The internal planetary gearbox is connected to external load gear. The assembly of DC motors, internal planetary gear box and external load gear introduce friction and damping at the joint.

The friction model and damping properties are identified using experimental method. The coulomb and viscous Friction model is considered to accurately model the friction behaviour at the joint. The offset value and gain for friction model are computed based on series of experiments by measuring constant angular velocities for given input voltage.

4.2 Model Validation

Lets consider the equations of motion of a flexible link manipulator which can be written as

Figure 4.1: *Experimental setup of a single link flexible manipulator*

Table 4.1: *The physical parameters of single link planar manipulator.*

Parameter Name	Nominal Value
Motor Inertia (I_m)	2.08e-3 Kgm^2
Motor Viscous Damping (B_m)	4e-3 $Nm/(rad/s)$
Payload Mass (M_p)	0.1 kg
Dimension of Link (LxHxW)	0.52x0.045x0.002 m
Tensile Modulus	206000 MPa
Density (ρ)	8253 Kg/m^3

$$\begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix} \begin{bmatrix} \ddot{q}_r \\ \ddot{q}_f \end{bmatrix} + \begin{bmatrix} C_{rr} & C_{rf} \\ C_{fr} & C_{ff} \end{bmatrix} \begin{bmatrix} \dot{q}_r \\ \dot{q}_f \end{bmatrix} + \begin{bmatrix} D_{rr} & 0 \\ 0 & D_{ff} \end{bmatrix} \begin{bmatrix} \dot{q}_r \\ \dot{q}_f \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_{ff} \end{bmatrix} \begin{bmatrix} q_r \\ q_f \end{bmatrix} = \begin{bmatrix} B_r \\ B_f \end{bmatrix} \tau \tag{4.1}$$

where q_r and q_f are the rigid and elastic coordinates.

Using the equation [4.1] the feedforward compensator is defined as

$$\tau = M_{rr}\ddot{q}_r + M_{rf}\ddot{q}_f + C_{rr}\dot{q}_r + C_{rf}\dot{q}_f + D_{rr}\dot{q}_r \tag{4.2}$$

For the given reference trajectories q_r, $\dot{q}_r$ and $\ddot{q}_r$, the input torque applied at manipulator joint can be computed by equation [4.2]. The values of elastic coordinates q_f, $\dot{q}_f$ and $\ddot{q}_f$ is set to zero because these values are not known apriori.

The out response of the feedforward compensator is shown

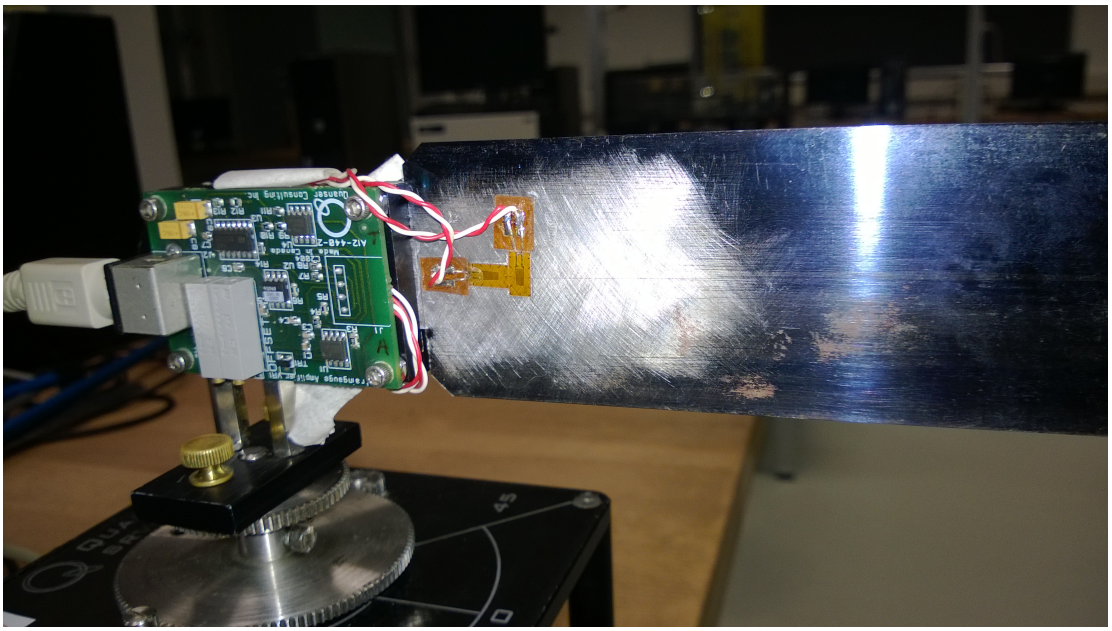

Figure 4.2: *Strain guage mounted on flexible manipulator*

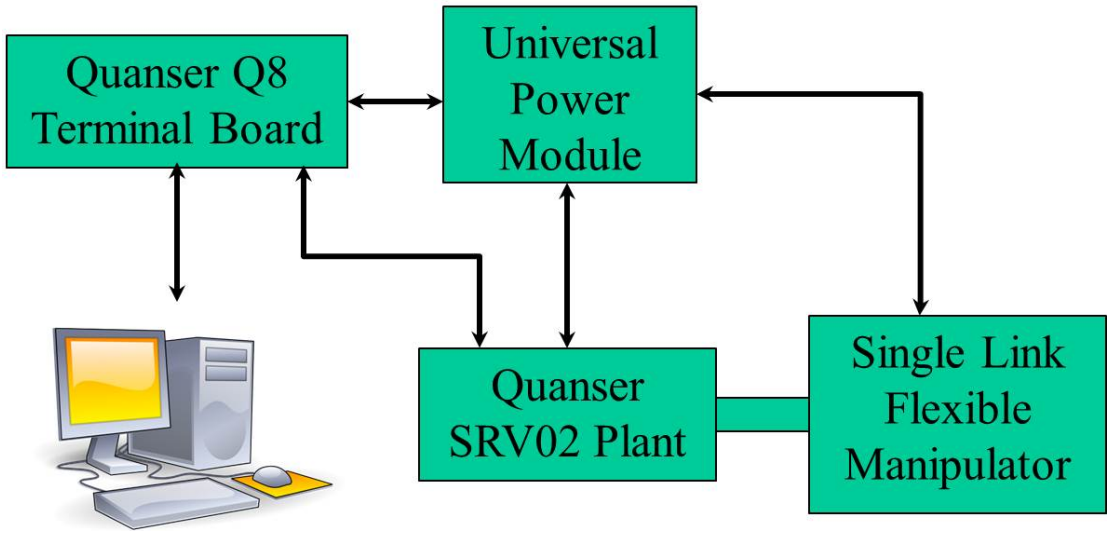

Figure 4.3: *Experimental setup of a single link flexible manipulator*

4.3 Friction Compensator

The friction at the manipulator joint plays a major role in the position control. The friction model can be identified using the Newtons first law of motion. The equations of motion of a Qunaser servo plant is defined as

$$T_m(t) - T_f = J_m\dot{\omega}(t) \tag{4.3}$$

where T_m is the motor torque, and T_f is the friction torque. J_m is the motor inertia, and $\omega(t)$ is the angular velocity at the load shaft.

From equation [4.3], if the angular velocity is constant then friction torque is equal to motor torque i.e.

$$T_m(t) = T_f \tag{4.4}$$

The torque applied by the servo motor at the manipulator joint can be defined as

$$T_m(t) = \frac{\eta_g \kappa_g \eta_m \kappa_t (V_m - \kappa_g \kappa_m \dot{\theta})}{R_m} \tag{4.5}$$

where V_m is the input voltage and $\dot{\theta}$ is the angular velocity of load shaft. The Quanser servo plant parameters are listed in Table [4.2]

Table 4.2: *Quanser servo plant parameters.*

Parameter	Description	Value
κ_t	Motor torque constant	7.68E-3 Nm
η_m	Motor efficiency	0.69
κ_g	Total gearbox ratio	70
η_g	Gearbox efficiency	0.90
κ_m	Back-emf constant	7.68E-3 V/(rad/s)
R_m	Motor armature resistance	2.6 Ω

From the equation [4.3] the friction torque is equal to motor torque when the angular velocity is constant. Hence, the frictional torque can be computed using equation [4.5]. The angular velocity $\dot{\theta}$ is measured at different constant angular velocities by increasing the input voltage V_m. The measured angular velocity for the given input voltage is shown in Table [4.3].

Table 4.3: *The experimental data: angular velocity measurement vs input voltage.*

Voltage (volts)	Angular velocity (rad/s)	Motor torque (Nm)
0.0	0	0
0.1	0	0.0128
0.2	0.095	0.0257
0.3	0.171	0.0385
0.4	0.253	0.0514
0.5	0.331	0.0642
1.0	0.756	0.1284
1.5	1.182	0.1926
2.0	1.614	0.2568
2.5	2.053	0.3210
3.0	2.489	0.3852
3.5	2.924	0.4494
4.0	3.372	0.5136
4.5	3.815	0.5778
5.0	4.243	0.6420

The coulomb and viscous friction model is considered to fit the measured data.

$$F_f(\dot{\theta}) = F_c sgn(\dot{\theta}) + \beta\dot{\theta} \tag{4.6}$$

where F_f is the total frictional torque, F_c is the coulomb friction torque, β is the viscous friction coefficient and $\dot{\theta}$ is the angular velocity.

From Table [4.3] the motor begins to rotate between the input voltage 0.1 V and 0.2 V. To find out the precise value of input voltage where the motor begins to rotate, the angular velocity $\dot{\theta}$ is measured for very small increments of voltage. It is noticed that at input voltage 0.11 V the motor begins to rotate. Thus the coulomb friction torque is defined as this point.

The coulomb frictional torque value is computed as

$$F_c = 0.0141Nm \tag{4.7}$$

The viscous friction coefficient β is computed using the slope of frictional torque plotted in Figure [4.4]. It is defined as

$$\beta = \frac{change\ in\ the\ friction\ torque}{change\ in\ the\ angular\ velocity} = 0.1496 \tag{4.8}$$

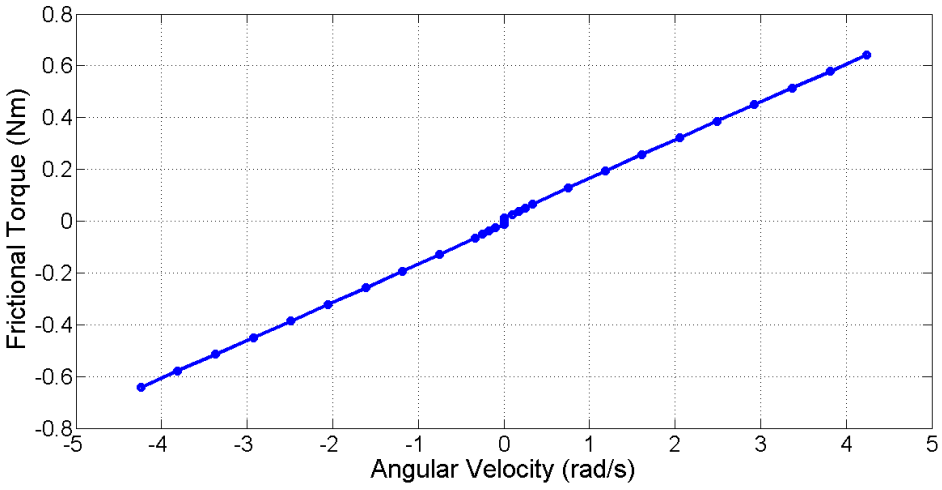

Figure 4.4: *Frictional torque F_f vs angular velocity plot using experimental data.*

4.4 Experimental Results

The following model based controllers are experimentally verified on single link flexible manipulator.

1. PD Control
2. Stable Inversion Control
3. Adaptive Control

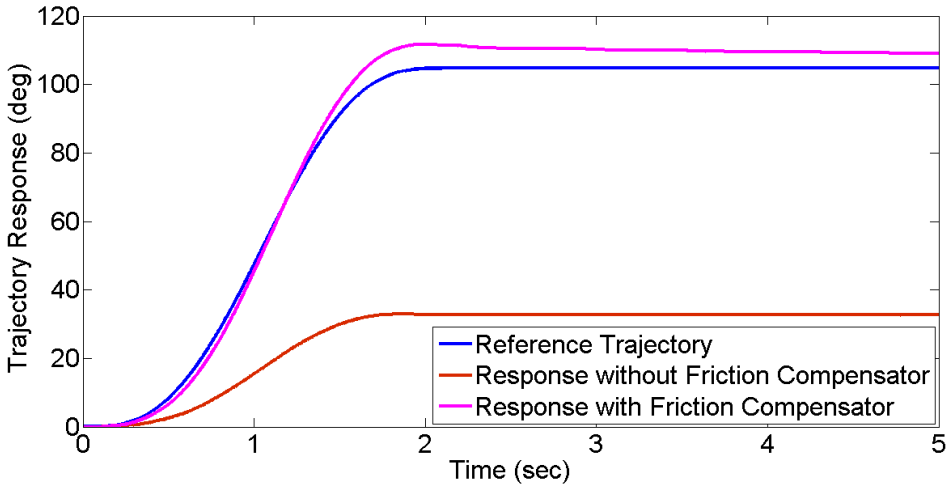

Figure 4.5: *The joint trajectory response of open loop system.*

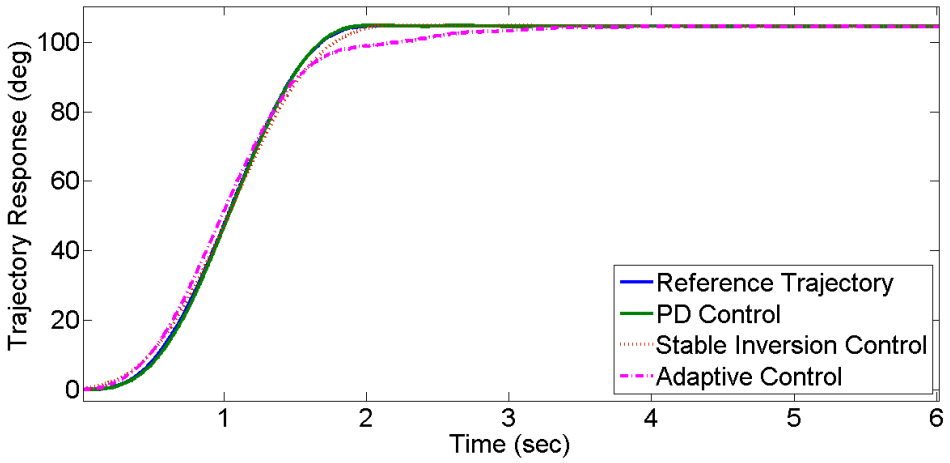

Figure 4.6: *The joint trajectory response.*

The performance of the model based controllers are tested in the presence of addition payload mass on the tip.

Figure [4.6] show the comparison between PD controller, stable inversion controller, and adaptive controller without payload mass. Figure [4.7] show the trajectory tracking error. Figure [4.8] show the tip displacements along the trajectory.

Figure [4.9] show the comparison between PD controller, stable inversion controller, and adaptive controller in the presence of additional payload mass. Figure [4.10] show the trajectory tracking error. Figure [4.11] show the tip displacements along the trajectory.

The experimental results show that stable inversion controller and adaptive controller have accurate trajectory tracking without payload mass on tip. In the presence of additional payload mass M_p = 0.1 Kg, adaptive control showed better tip trajectory tracking compared to PD controller, and stable inversion controller.

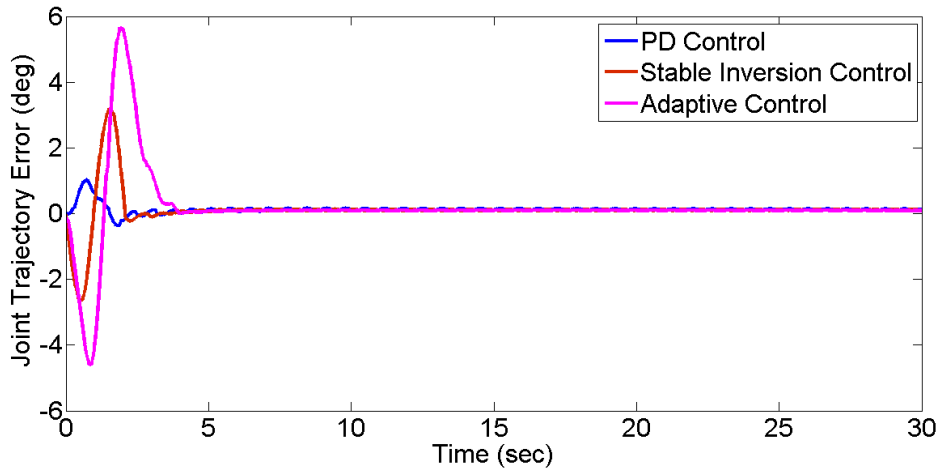

Figure 4.7: *The error along joint trajectory.*

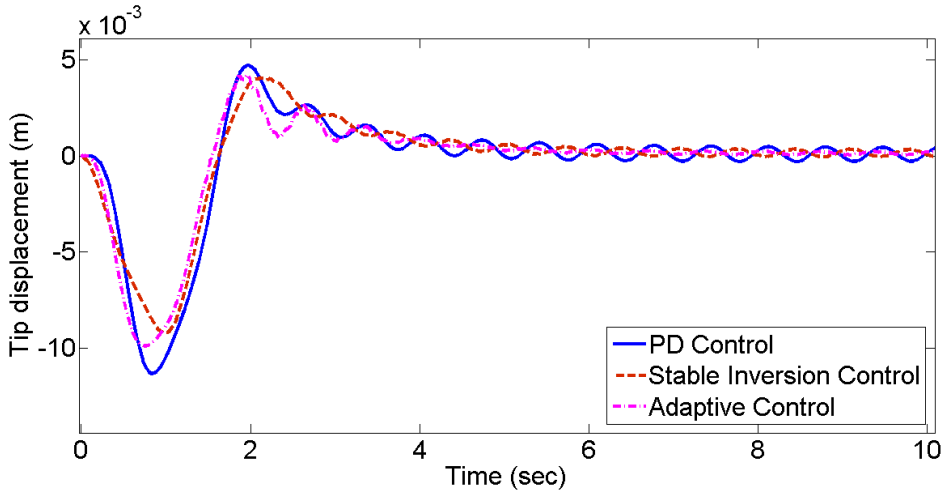

Figure 4.8: *Tip displacement along the trajectory.*

The advantage of the stable inversion control is that it does not require measurement of the tip displacement because these values are estimated off-line using Iterative learning method. The stable inversion control showed good trajectory tracking and minimized vibrations without payload mass as it is seen in Figure [4.7] and Figure [4.8], but for additional payload mass the stable inversion control is not efficient to damp the vibrations because the estimated off-line elastic displacements heavily depend on the accuracy of the dynamic model which is unknown a priori in case of unknown payload mass.

The adaptive control measures tip displacement along the trajectory and minimizes the tip vibrations. It has tip displacements and joint trajectory error as a feedback in the control. Thus, adaptive control showed better performance in trajectory tracking and vibration suppression in the presence of additional payload mass.

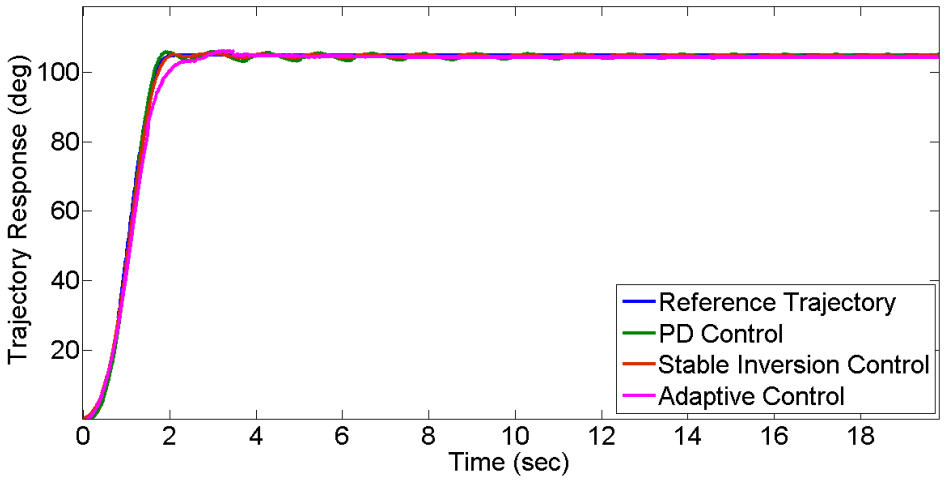

Figure 4.9: *The joint trajectory response in the presence of additional payload mass.*

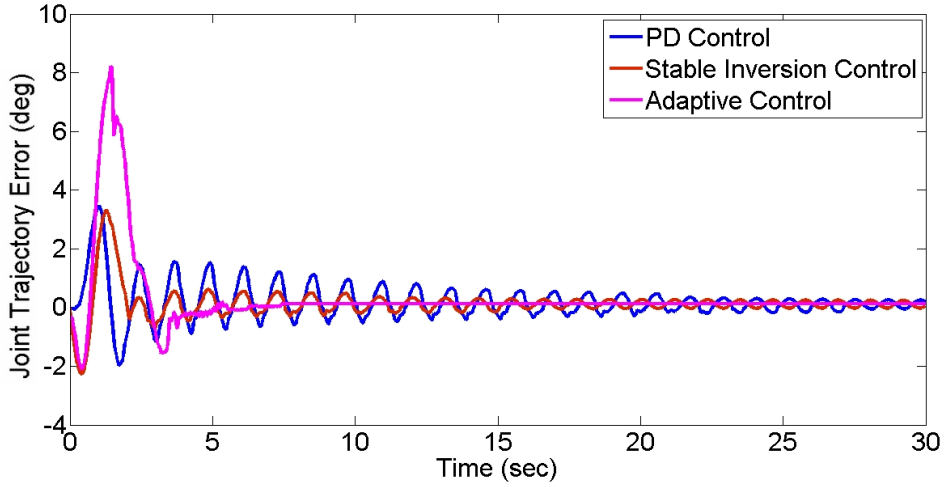

Figure 4.10: *The error along joint trajectory in the presence of additional payload mass.*

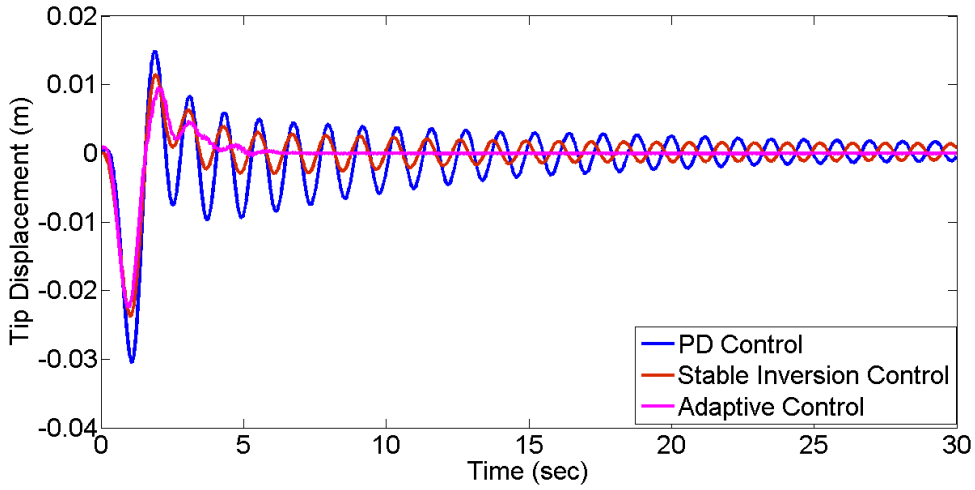

Figure 4.11: *Tip displacement along the trajectory in the presence of additional payload mass.*

CHAPTER 5

Conclusions

The goal of this thesis was to develop systematic approach for dynamic modeling and control of spatial flexible manipulators. The thesis work on flexible manipulators was divided into two parts. The first part was focused on dynamic modelling of spatial flexible manipulators while the second part was focused on control design of spatial flexible manipulators for trajectory tracking.

A general purpose multi-body code has been developed to obtain a nonlinear dynamic model of spatial flexible manipulators for model based control design and simulation purposes. Both link and joint flexibilities can be included in the dynamic modeling. The flexible links are discretized to get a finite dimensional dynamic model.

The deformation of each link is assumed to be due to both bending and torsion. The deformation of the joints is assumed to be due to pure torsion. The deformation of each link is assumed to be small relative to the rigid body motion. Thus, the configuration of each link is defined as the sum of rigid and elastic coordinates using a floating reference frame. The dynamic model is first derived using the principle of virtual work along with finite element method in generalized coordinates for general purpose implementation. Then, the system of equations in generalized coordinates is converted into independent coordinate form using a recursive kinematic formulation based on the topology of a manipulator.

The advantage of general purpose multi-body code is that it uses minimum set of equations that define the dynamics of flexible manipulator, which is required in control design to reduce computation cost. In addition, it allows the dynamic modeling of any arbitrary manipulator configuration that consists of rigid links, flexible links and flexible joints.

Numerical simulation results of an open chain RRR spatial manipulator with flexible links and flexible joints was presented to show the effects of flexibility on robot manip-

ulator dynamics. The simulations results showed that the link and joint flexibility can alter the motion of endeffector in workspace. Thus, Ignoring the link or joint flexibility can cause poor estimation of dynamic parameters and, eventually, poor performance of the control design.

Model based controllers were developed for trajectory tracking and vibration suppression of spatial flexible link manipulators. The following model based controllers were designed for an open chain RRR spatial flexible link manipulator:

1. PD Control
2. Stable Inversion Control
3. A Nonlinear Control
4. Adaptive Control

Among them, PD control and Stable inversion control are derived using feedback linearization technique. A nonlinear control and Adaptive control are derived using sliding mode technique.

The simulation results of PD controller showed good trajectory tracking at joint space. However, it is not very efficient to damp the endeffector vibrations. The stable inversion control showed better trajectory tracking compared to PD control, however this method is incapable of damping the vibrations in case of unmodeled dynamics such as unknown payload mass. Moreover, stable inversion control can have good performance for planar flexible link manipulators compared to spatial flexible link manipulators.

Nonlinear controller and adaptive controller showed good trajectory tracking at the joint space and also efficient to damping the endeffector vibrations compared to PD controller and stable inversion controller.

Experimental results on a single link flexible manipulator showed that the adaptive controller has better trajectory tracking and vibration suppression compared to PD control and stable inversion control in the presence of additional unknown payload mass on the endeffector.

Through the present work, the state of the art developed for planar flexible links manipulators was extended to a spatial flexible link manipulators. It is learnt that the spatial flexible link manipulators require additional control effort to damp the out of plane vibration modes. These vibration modes are due to out of plane bending effect. The future work on spatial flexible link manipulators can be focused to improve the damping properties of out-of-plane vibration modes. Further studies can be focused on the additional actuator and its positions to damp the out-of-plane bending modes.

Bibliography

[1] A. Akira, and K. Komuro. An energy saving open-loop control technique for flexible manipulators. *International Conference on Mechatronics and Automation*, pages 416–421, 2011.

[2] A. Albu-Schaffer, O. Eiberger, M. Grebenstein, S. Haddadin, C. Ott, T. Wimbock, S. Wolf, and G. Hirzinger. Soft robotics: From torque feedback controlled lightweight robots to intrinsically compliant systems. *IEEE Robotics and Automation Mag.*, 15(3):20–30, 2008.

[3] A. Daryabor, A.H. Abolmasoumi, and H. Momeni. . Robust pole placement control of flexible manipulator arm via lmis. *41st Southeastern Symposium on System Theory*, pages 274–279, 2009.

[4] A. De Luca, B. Siciliano. Closed-form dynamic model of planar multi-link lightweight robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(4):826–839, 1991.

[5] A. De Luca, S. Iannitti, R. Mattone and G. Oriolo. Control problems in underactuated manipulators. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2:855–861, 2001.

[6] A. Konno, L. Deman, and M. Uchiyama. A singularly perturbed method for pole assignment control of a flexible manipulator. *Robotica*, 20(6):637–651, 2002.

[7] A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, 2005.

[8] A.De. Luca. Trajectory control of flexible manipulators. *Control Problems in Robotics and Automation*, pages 83–104, 1998.

[9] A.De. Luca, and B. Siciliano. Joint-based control of a nonlinear model of a flexible arm. *Proceedings of IEEE in American Control Conference*, pages 935–940, 1988.

[10] A.De. Luca and B. Siciliano. Trajectory control of a non-linear one-link flexible arm. *International Journal of Control*, 50(5):1699–1715, 1989.

[11] A.De. Luca, and B. Siciliano. Inversion-based nonlinear control of robot arms with flexible links. *Journal of guidance, control, and dynamics*, 16(6):1169–1176, 1993.

[12] A.De. Luca, F. Nicolo, and G. Ulivi. Trajectory tracking in flexible robot arms. *System Models and Feedback: Theory and Applications*, pages 17–34, 1992.

[13] A.De. Luca, S. Panzieri, and G. Ulivi. Stable inversion control for flexible link manipulators. *IEEE International Conference on Robotics and Automation*, 1:799–805, 1998.

[14] A.H. Abolmaasoumi, H.R. Momeni, and A. Daryabor. Robust $\hat{\mathrm{I}}\frac{1}{4}$-synthesis control of a flexible manipulator arm. *41st Southeastern Symposium on System Theory*, pages 330–334, 2009.

[15] Atsushi Konno and Masaru Uchiyama. Modeling of a flexible manipulator dynamics based on the holzers method. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:223–229, 1996.

[16] B. Siciliano, and W.J. Book. A singular perturbation approach to control of lightweight flexible manipulators. *The International Journal of Robotics Research*, 7(4):79–90, 1988.

[17] C. Sallaberger, SPT Force. Canadian space robotic activities. *Acta astronautica*, 41(4):239–246, 1997.

[18] C.J. Damaren. Adaptive control of flexible manipulators carrying large uncertain payloads. *Journal of robotic systems*, 13(4):219–228, 1996.

[19] C.L. Damaren. Approximate inverse dynamics and passive feedback for flexible manipulators with large payloads. *IEEE Transactions on Robotics and Automation*, 12(1):131–138, 1996.

[20] D. Xilun, and J. M. Selig. Lumped parameter dynamic modeling for the flexible manipulator. *Fifth World Congress on Intelligent Control and Automation*, 1:280–284, 2004.

[21] D.S. Kwon and W.J. Book. A time-domain inverse dynamic tracking control of a single-link flexible manipulator. *J. Dynamic Systems, Measurement, and Control*, 116:193–200, 1994.

[22] E. Barbieri and U. Ozguner. Unconstrained and constrained mode expansions for a flexible slewing link. *IEEE in American Control Conference*, pages 83–88, 1988.

[23] E. Bayo. A finite element approach to control the end point motion of a single link flexible robot. *Journal of Robotic Systems*, 4(1):63–75, 1987.

[24] E. Bayo and H. Moulin. An efficient computation of the inverse dynamics of flexible manipulators in the time domain. *Proceedings of IEEE Conterence on Robotics and Automation*, pages 710–715, 1989.

[25] E.G. Christoforou and C.J. Damaren. The control of flexible link robots manipulating large payloads: Theory and experiments. *Journal of robotic systems*, 17(5):255–271, 2000.

[26] F. Resta, F. Ripamonti, G. Cazzulani, and M. Ferrari. Independent modal control for nonlinear flexible structures: an experimental test rig. *Journal of Sound and Vibration*, 329(8):961–972, 2010.

[27] F.L. Lewis, D.M. Dawson and C.T. Abdallah. *Robot Manipulator Control: Theory and Practice*. CRC Press, 2003.

[28] G. Hirzinger, M. Fischer, B. Brunner, R. Koeppe, M. Otter, M. Grebenstein. Advances in robotics: The dlr experience. *The International Journal of Robotics Research*, 18(11):1067–1087, 1999.

[29] G. OKE and Y. ISTEFANOPULOS,. End-effector trajectory control in a two-link flexible manipulator through reference joint angle values modification by neural networks. *Journal of Vibration and Control*, 12(2):101–117, 2006.

[30] Gutierrez, Luis Benigno, F. L. Lewis, and J. Andy Lowe. Implementation of a neural network tracking controller for a single flexible link: comparison with pd and pid controllers. *IEEE Transactions on Industrial Electronics*, 45(2):307–318, 1998.

[31] G.V. Kostin, M.C. Steinbach, H.G. Bock, R.W. Longman. Modeling of dynamics of industrial robots with flexible electric drives. *Proceedings of International Conference on Control of Oscillations and Chaos*, 2:275–278, 1997.

[32] I.H. AKYUZ, Z. BINGUL, and S. KIZIR. Cascade fuzzy logic control of a single-link flexible-joint manipulator. *Turk J Elec Eng and Comp Sci*, 20:713–726, 2012.

[33] J.B. Jonker. *A Finite Element Dynamic Analysis of Flexible Spatial Mechanisms and Manipulators.* Delft University Press, 1988.

[34] J.F. Besseling, and D. G. Gong. Numerical simulation of spatial mechanisms and manipulators with flexible links. *Finite elements in analysis and design*, 18(1):121–128, 1994.

[35] J.H. Ryu, D.S. Kwon, and Y. Park. A robust controller design method for a flexible manipulator with a time varying payload and parameter uncertainties. *Proceedings of IEEE Conterence on Robotics and Automation*, 1:413–418, 1999.

[36] J.H. Yang, F.L. Lian, and L.C. Fu. Adaptive robust control for flexible manipulators. *Proceedings of IEEE Conterence on Robotics and Automation*, 1:1223–1228, 1995.

[37] J.H. Yang, F.L. Lian, and L.C. Fu. Nonlinear adaptive control for flexible-link manipulators. *IEEE Transactions on Robotics and Automation*, 13(1):140–148, 1997.

[38] K. Yuan, and L.Y. Liu. Achieving minimum phase transfer function for a noncollocated single link flexible manipulator. *Asian Journal of Control*, 2(3):179–191, 2000.

[39] K. Yuan, L. Lin. Motor-based control of manipulators with flexible joints and links. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1809–1814, 1990.

[40] K.Y. Kuo and J. Lin. Fuzzy logic control for flexible link robot arm by singular perturbation approach. *Applied Soft Computing*, 2(1):24–38, 2002.

[41] L. Tian and C. Collins. Adaptive neuro-fuzzy control of a flexible manipulator. *Mechatronics*, 15(10):1305–1320, 2005.

[42] L. Tian, Lianfang, and Z. Mao. Fuzzy neuro controller for a two-link rigid-flexible manipulator system. *Proceedings of the 9th International Conference on Neural Information Processing*, 4:1867–1871, 2002.

[43] L.Le. Tien, A.A. Schaffer, G. Hirzinger. Mimo state feedback controller for a flexible joint robot with strong joint coupling. *IEEE International Conference on Robotics and Automation*, pages 3824–3830, 2007.

[44] M. Balas. Trends in large space structure control theory: fondest hopes, wildest dreams. *IEEE Transactions on Automatic Control*, 27(3):522–535, 1982.

[45] M. Benosman and G. Le. Vey. Stable inversion of siso nonminimum phase linear systems through output planning: an experimental application to the one-link flexible manipulator. *IEEE Transactions on Control Systems Technology*, 11(4):588–597, 2003.

[46] M. Benosman and G.Le. Vey. Model inversion for a particular class of nonlinear non-minimum phase systems: an application to the two-link flexible manipulator. *Proceedings of IEEE International Conference on Decision and Control*, 2:1174–1180, 2001.

[47] M. Farid, S.A. Lukasiewicz. Dynamic modeling of spatial manipulators with flexible links and joints. *Computers and Structures*, 75(4):419–437, 2000.

[48] M. Isogai, F. Arai, and T. Fukuda. Modeling and vibration control with neural network for flexible multi-link structures. *Proceedings of IEEE Conterence on Robotics and Automation*, 2:1096–1101, 1999.

[49] M. Karkoub, G. Balas, K. Tamma, and M. Donath. Robust control of flexible manipulators via $\hat{\mathbf{I}}\frac{1}{4}$-synthesis. *Control engineering practice*, 8(7):725–734, 2000.

[50] M. Khairudin. Dynamic modeling of a flexible link manipulator robot using amm. *TELKOMNIKA*, 6(3):185–190, 2008.

[51] M.A. Ahmad, Z. Mohamed, and Z.H. Ismail. Experimental investigation of feedforward control schemes of a flexible robot manipulator system. *Elektrika*, 10(2):28–35, 2008.

[52] M.A. Ahmad, Z. Mohamed, N. Hambali. Dynamic modeling of a two-link flexible manipulator system incorporating payload. *IEEE Conference on Industrial Electronics and Applications*, pages 96–101, 2008.

[53] M.H.F. Dado, A.H. Soni. Dynamic response analysis of 2-r robot with flexible joints. *Proceedings of the IEEE International Conference on Robotics and Automation*, 4:479–483, 1987.

[54] M.J. Balas. Active control of flexible systems. *Journal of Optimization Theory and Applications*, 25(3):415–436, 1978.

[55] M.O. Tokhi, and A. K. M. Azad. Real-time finite difference simulation of a single-link flexible manipulator system incorporating hub inertia and payload. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 209(1):21–33, 1995.

[56] M.O. Tokhi, and A. K. M. Azad. Finite difference and finite element approaches to dynamic modelling of a flexible manipulator. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 211(2):145–156, 1997.

[57] M.O. Tokhi, Z. Mohamed, S.H.M. Amin, et al. Dynamic characterization of a flexible manipulator system, theory and experiments. *Proceedings of TENCON*, 3:167–172, 2000.

[58] M.W. Spong. Modeling and control of elastic joints robots. *ASME Journal of Dynamic Systems, Measurement, and Control*, 109(6):310–319, 1987.

[59] Pradhan, Santanu Kumar, and Bidyadhar Subudhi. Real-time adaptive control of a flexible manipulator using reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 9(2):237–249, 2012.

[60] R. Bischoff, et al . The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing. *41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–8, 2010.

[61] R. Fareh and M. Saad. Adaptive control for a single flexible link manipulator using sliding mode technique. *6th International Multi-Conference on Systems, Signals and Devices*, pages 1–6, 2009.

[62] R. Seifried, M. Burkhardt and A. Held. Trajectory control of flexible manipulators using model inversion. *ECCOMAS Thematic Conference on Multibody Dynamics*, 2011.

[63] R.H. Cannon, and E. Schmitz. Initial experiments on the end-point control of a flexible one-link robot. *The International Journal of Robotics Research*, 3(3):62–75, 1984.

[64] R.J. Theodore, A. Ghosal. Comparison of the assumed modes and finite element models for flexible multilink manipulators. *The International Journal of Robotics Researches*, 14(2):91–111, 1995.

[65] R.J. Theodore and A. Ghosal. Robust control of multilink flexible manipulators. *Mechanism and machine theory*, 38(4):367–377, 2003.

[66] S. Moberg, and S. Hanssen. A dae approach to feedforward control of flexible manipulators. *IEEE International Conference on Robotics and Automation*, pages 3439–3444, 2007.

[67] S.J. Huang, and T.Y. Wang. Structural dynamics analysis of spatial robots with finite element approach. *Computers and structures*, 46(4):703–716, 1993.

[68] S.S. Ge, T. H. Lee, and G. Zhu. A new lumping method of a flexible manipulator. *Proceedings of IEEE American Control Conference*, 3:1412–1416, 1997.

[69] T. Yoshikama and K. Hosoda. Modeling of flexible manipulators using virtual rigid links and passive joints. *The International Journal of Robotics Research*, 15(3):290–299, 1996.

[70] V. A. Spector and H. Flashner. Modeling and design implications of noncollocated control in flexible systems. *Journal of Dynamic Systems, Measurement and Control*, 112(2):186–193, 1990.

[71] V. Etxebarria, A. Sanz, and I. Lizarraga. Control of a lightweight flexible robotic arm using sliding modes. *arXiv preprint cs/0601054*, 2006.

[72] W. Chen. Dynamic modeling of multi-link flexible robotic manipulators. *Computers and Structures*, 79(2):183–195, 2001.

[73] W.J. Book. Recursive lagrangian dynamics of flexible manipulator arms via transformation matrices. 1983.

[74] W.J. Book. Modeling, design, and control of flexible manipulator arms: A tutorial review. *IEEE International Conference on Decision and Control*, pages 500–506, 1990.

[75] Yang, H., H. Krishnan, and M. H. Ang Jr. A simple rest-to-rest control command for a flexible link robot. *IEEE International Conference on Robotics and Automation*, 4:3312–3317, 1997.

[76] Y.L. Hwang. Recursive newton-euler formulation for flexible dynamic manufacturing analysis of open-loop robotic systems. *The International Journal of Advanced Manufacturing Technology*, 29(5-6):598–604, 2006.

[77] Z. Mohamed, and M.O. Tokhi. Command shaping techniques for vibration control of a flexible robot manipulator. *Mechatronics*, 14(1):69–90, 2004.

[78] Zhao, Hongchao, and Degang Chen. Tip trajectory tracking for multilink flexible manipulators using stable inversion. *Journal of Guidance Control and Dynamics*, 21(2):314–320, 1998.