POLITECNICO
DI TORINO

Master’s Degree in Mechatronic Engineering

Thesis

Dynamic Modelling and Control of Flexible-link Multibody Systems

Supervisors

Alessandro Rizzo

Giovanni G. Muscolo

Candidate

Chukwudi David Nchekwube

Academic year 2018-2019

Abstract

Multibody systems are mechanisms of interconnected bodies with application in many areas of engineering. There is a need to optimize such systems for, among others, the minimization of the torque demand of the system motion, and also for maximization of the velocity of the systems. The integration of lightweight flexible bodies in multibody systems makes sense as the lower weight would imply an overall lighter systems and faster system; but this also comes with vibrations that in turn affect the system dynamics. It is necessary to have a control system for the motion of the system such that the torque and velocity targets are met and keeping vibrations low. This work focuses on a four-bar linkage multibody system. Moreover, the four-bar linkage system in this case includes two extra links in the middle of the structure symmetrically dividing the system. This architecture is based on the horizontal-to-vertical and vertical-to-horizontal target motion of the system's coupler link. For the study, different configurations of lightweight flexible body integration are modeled and observed for nature of t he system performance in terms of the speed, torque and vibration. From the foregoing, a most suitable configuration is selected and following this, a control architecture is proposed for the system such that the torque demand is kept below the set limit, link vibrations are lowered and the system is fast. In this work, Matlab/SimscapeMultibody is used to model and study the system. Using the Simulink environment, the controller is designed to satisfy the performance requirement and to follow the reference motion path. Simulation results are presented, which show that the controller can control the system motion and keep vibrations low. The dynamic modeling and control approach presented can be applied to other cases of multibody systems other than the four-bar linkage mechanism.

Acknowledgements

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multibody systems in the most basic words consists of interconnected bodies. The system can be comprised of the only rigid bodies, only flexible bodies, or a mix of rigid and flexible bodies. Multibody systems applies to many engineering fields and largely seen in such areas as robotics, aeronoutics,etc. To handle these systems it is required to study how the respective bodies link to and affect each other and in turn affect the overall system composition. This is an analysis of the movement and force dynamics of the system. In another breadth, however still in the same light, multibody systems represents a formalism used to model the dynamic behaviour of interconnected rigid or flexible bodies, each of which may undergo large rotational and translational displacements.

Nowadays there is need for light and high-speed mechanisms and one that consumes the least amount of energy as possible. In this light, in the application of the multibody systems such as the *n-bar linkage mechanism* (where n represents the number links that make up the system), there is need for such mechanisms that adhere to the aforementioned demand; to reduce inertia forces and driving torque requirement. Hence, research have been made with regard to the inclusion of flexible materials in the mechanical system with different works involving different levels of flexibility inclusion; from a single flexible link to cases of an all-link-flexible system.

Multibody systems undergoing motion, which is the focus in this work, of course have a different and more complex behaviour than single-body setups. The

Dynamic Modeling and control of multibody systems is somewhat of intertwined process which simply implies respectively first the engineering study of the motion of the multibody system in question to understand the behaviour of the system and consequently designing of a control protocol, based on some set requirements, to have the system run in a desired manner. In engineering mechanics, while kinematic study of systems, or simply ’kinematics’, deals with the motion of bodies or systems without considering forces responsible, ’dynamics’, deals with the connection between body or system motion and the responsible forces acting on it. As would be expected, the complexity of the multibody system, affected by factors such as number of links, size, shape, etc, proportionally affects the complexity of the modeling process. The design of multibody systems is needed to be efficient. Essentially the system has to be well designed and the design engineer has to understand the kinemnatic and dynamic behaviour of the system beforehand so the system motion runs according to the design intent.

Several approaches have been taken toward the study of the dynamics of multibody systems amounting to formalisms such as Newton and Euler’s and Lagrange’s. The mathematical equations for these multibody system formalisms may differ in appearance but the physics concept behind them all remains the same.

Solving multibody system dynamics problems using analytical methods is difficult. Fundamentally, the dynamics of a system implies the formulation of the equations of motion and there are methods and principles such as Newtons’s law, principle of conservation of motion, etc. Usually, in the process of solving these systems, the equations of motion are too complex to solve by hand. As a result of the nature of multibody system dynamics there is no singular method so powerful and general such that it would be always applied to the resolution of multibody dynamics tasks in general. This is the basis for the turn to physical modeling of these systems using computer-aided approach where the such laws as Newton’s equations of motion is employed to model the multibody system in a uniform and consistent manner and where the powerful computational ability of the computer is harnessed to solve the equations using the numerical methods implemented on computers. A number of computer programs exist for solving multibody systems with preferences made in a case by case basis and with regard to the respective ability and flexibility of each package. In [10] co-simulation is presented where

CAD application is used for dynamics solving. The dynamic model is then exported to a control design software for the development of the anticipated control law. In most cases, an 'external' application, different from the control design one is used for dynamics modeling. In this work, the application for modeling and control design is somewhat integrated; SimScape Multibody^{TM} (formerly aliased 'SimMechanics') and Simulink^{T}M, both of the Mathworks platform are employed for the dynamic modeling and control architecture design respectively.

1.0.1 Dynamics of flexible-link multibody systems

Light bodies have been required in the implementation of systems for the inertial implications but come with deflection of the body under the application of a force or torque. To utilize these bodies properly and accurately, they need to be modelled appropriately for use in the light of application of controls. Works have investigated the modeling of multibody systems with flexible parts/links. The dynamics of flexible multibody sytems was analyzed in [1] and [2] by a finite segement approach. The work presented a theoritical basis for the modeling approach based on *Kane's* equation modeling flexibility using spring and dampers at the joints of the system. In [3], the geometric constraint at the end of a flexible link was derived and introduced into Hamiltons principle to formulate the governing equations of the connecting rod which is modelled by Timoshenko beam theory. In [5], finite element analysis was used for a high-speed manipulator while in one of his earlier works, Giovagnoni [6] used the finite element approach for dynamics modelling of flexible planar linkages with 1 d.o.f. and treating elastic and rigid body degree of freedom in identical manner, and obtained an equilibrium equation involving all the mechanisms members by means of sensitivity coefficients. A detailed review of the dynamic analysis of flexible links is presented in the work of Santosha et al [4], detailing the methods of assumed mode, finite element method and lumped parameter methods applied to single-link multibody system in the form of manipulators, as well as two-link manipulators and flexible link manipulators. [8] and[9] also presents developments in flexible multibody dynamics.

1.0.2 Control of flexible-link multibody systems

In the actuation of the linkage system, different works have featured different utilization of actuation. Some case have feautured two actuators on the fixed bar, others have used a single actuator. In [11],[12] an innovative actuation approach is adopted including a link connected centrally to the default coupler link of a four-bar linkage system and incorporating only one actuator. This novel approach handles the problems of interference while minimizing system complexity. Many bar linkage mechanism literature have focused on area of trajectory algorithm optimization and dynamic balancing. A review on this is in the work of [11].

The use of lightweight flexible links to reduce torque and increase speed introduces elastodynamic issues. There is need for control systems for the flexible four-bar linkage system to suppress vibrations, maintain as low as possible torque with a fast system speed. Zhang et al [13] investigated the control of a high-speed flexible linkage mechanism by a H_∞ active vibration controller, using a pair of piezoelectric sensor and actuator on each link. The controller presented was to provide robustness, handling uncertainty at high frequencies. In [14], a similar approach was employed to a different kind of structure but showing H_∞-based control's able to handle various vibration modes demonstrating robustness to up to 5^{th} mode. In [15] an inverse dynamics controller is proposed for four bar linkage system with a single flexible link. Using a zero dynamics stabilizer it tracked tip trajectory for the link system. However, the exact approach presented is not practical for elastic coordinates and speeds because an observer would be needed to get the state of the dynamic model.

Essentially, to apply control theory to systems, they have to be linear systems. The dynamic model of the flexible four-bar linkage mechanism is non-linear. Thus, there is need to linearise the system for the applying control laws. In [16], and in [17], is presented a modelling of a flexible-link planar mechanism based on finite element method (FEM) and decomposing the overall motion using equivalent rigid link system (ERLS) into rigid body motion and elastic motion. Gasparetto [17] synthesized a non-linear model, linearised it about an equilibrium point. Both models were compared showing the linear model as a good model approximation of the non-linear model. Caracciolo and Trevisani [18] using similar approach, presented a simultaneous rigid-body motion PID controller and proportional vibration controller. Wang et al. [19] from the FEM method presented an LQR controller for simultaneous motion and vibration control of a multi flexible-link system. The controller used a strain rate feedback, used an adaptive disturbance estimator and Luenberger observer to estimate state variables. Trevisani [20] also presented a feedback control for flexible four-bar linkages that featured a position and vibration controller working simultaneously; PID control for position and P-control for vibration. Incorporating a gravity compensator the controller suppresses amplitude of link oscillation associated with first natural frequencies. Boscariol et al.[21] showed that a model-based predictive control approach can be applied to systems with non-linear and fast-changing dynamics. Implemented on a single-link mechanism, the control approach estimated the nodal displacements and allowed the reduction of spring-back effects.

Trevisani and Valcher [22] remarking that the complexity of fully coupled analytical methods makes real-time model-basd control a difficult task, proposed an energy-based control using simple PD controllers and proportional controllers, and showed that applying the proposed control law, the resulting control system is stable. Caracciolo et al [23] presented delayed reference control (DRC) which reduces link deformation by suitably shaping the time history of the position reference on the basis of sensed link strains. A mixed-norm approach was introduced in [24]. It proposed a synthesis of mixed H_2/H_∞ regulators for position and vibration control of flexible link mechanisms. Rong et al. [25] utilising piezoelectric actuators and strain gauge sensors presented a control scheme for flexible four-bar linkage modelling dynamics via a discrete time transfer matrix method of multibody system (MS-DTTMM) and then proposed a non-linear fuzzy neural network controller to suppress vibration. Shawky et al [26] used state dependent Riccatti equation (SDRE) technique to present a control strategy for a flexible link manipulator requiring only measurements of the link's free-end displacement. In [27] Ulbrich and Von Stein proposed a feedforward-feedback combination control strategy to improve the dynamics of a flexible mechanism. Defining a regulation and a tracking problem, a control law was derived that mitigated deviations of a link system's tracer point. In [28], Hill presented a modified *Kane's* equation of motion approach to controlling the motion of a rigid and flexible four bar coupler. Lyapunov methods was used to derive a control law to provide tracking and vibration damping of the coupler.

A four-bar system developed by Giovanni [11] which includes two centrally positioned links is used in the study. A physical model of the system is made using Matlab Simscape Multibody and control system is applied. In the physical model it is convenient to make quick changes and observe effects. Lumped parameter approach is used for the flexible body modeling. In the foregoing review, research have sort to incorporate flexible bodies to effect lower energy consumption and thus implying smaller actuators and also higher speed of the mechanical system. On the basis of the aforementioned four-bar linkage, this work aims at dynamic modelling and control approach for the minimization of torque, maximizing of speed and suppression of vibration. Different flexible link incorporation are made in the four-bar linkage mechanism, in each case investigating the impact on the torque, speed, and vibrations. A proposed control scheme is presented for the mechanism. The PD-LQR control structure is presented and the Linear quadratic optimal control structure is presented. Numerical simulations are performed using Matlab Simulink.

1.0.3 Thesis Organization

The rest of the thesis is structured as follows. Chapter two deals with the analytical model of multibody systems with flexible links focusing on the four-bar linkage mechanism used for the body and implementing a good formulation for analysis. In chapter three physical modeling approach is discussed and the analysis that can be done thereby is presented. Chapter four focuses on the control design and presents the important 'sub-steps' that are involved in the designing of a controller for the modelled multibody system. Chapter five presents the simulation of the system and the results. The last chapter presents the conclusion.

Chapter 2

Analytical Modeling of Flexible-link Multibody Systems

The analytical modeling of the multibody systems with flexible links can take different approaches. The Finite element approach has been choosen to model the body and the analysis is presented in this section. In this section the finite element method (FEM) is application to a multibody system is presented with the sample four-bar linkage mechanism. Flexibility is decomposed into rigid body motions and elastic deformations in the light of the concept of equivalent rigid link system (ERLS), and with the ensuing mathematical analysis a state-space model of the system is obtained. As this approach is not the main focus of the work, it would not be completely elaborated but would still be clear enough. However it is interesting to observe the cumbersome-ness as compared to the adopted physical modeling process presented in the following chapter.

2.0.1 Dynamic Model based on FEM method and ERLS

The dynamic model of the flexible-link multibody system is modelled by beam elements following the finite element method, decomposing the motion of the system into a rigid body motion of an equivalent rigid link system (ERLS) and an elastic deformation. Next, the motion equations of the mechanism is written through the application of the principle of virtual work. The virtual work principle leads to the system of equations providing accelerations of the free coordinates of the mechanism and the accelerations at the nodes.

The model outcome is a non-linear one as would be shown due the quadratic components existing in the differential equation relating the node accelerations and the free coordinates velocities. A linearization process for the model is necessary as the system theory requires linear models for application. The linearization is made about an operating point and the linear model outcome can be used to represent the state-space form of the system for instance and thus control structures can be applied on it. [**?**] shows that the for a planar flexible-link mechanism the linearization is valid. In Fig.1, **u** is the nodal displacement, **r** is the position of the nodes, both in the ERLS; **p** is the vector of position of the generic point of the finite element obtained by adding the vector of the corresponding point in the ERLS, **w** to the elastic displacement, **v**:

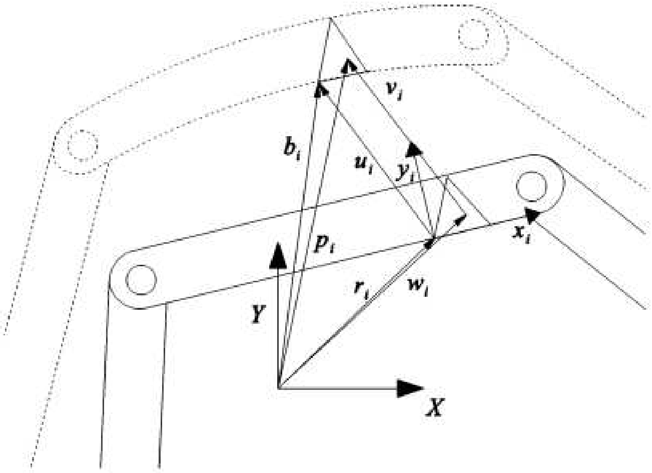


Figure 2.1. Dynamic system model

$$p_i = w_i + v_i. \tag{2.1}$$

The sum of the ERLS position and the elastic displacements, give the displacements and the rotations at the nodes:

$$b_i = u_i + r_i. \tag{2.2}$$

i representing the ith finite element. Each finite element has a local co-ordinate following the ERLS. A transformation matrix $T_i(q)$ and $R_i(q)$ are needed respectively for transformations from the global co-ordinate frame to the local co-ordinate frame and vice-versa.Using these matrices, equation 2.0.1 can be written as

$$p_i = w_i + R_i(q)N_i(x_i, y_i, z_i)T_i(q)u_i, \tag{2.3}$$

where $N_i(x_i, y_i, z_i)$ is the shape function for the interpolation of the ith finite element in the local co-ordinate frame. We can then have

$$\delta p_i = \delta w_i + \delta v_i \tag{2.4}$$

where

$$\delta w_i = R_i(q)N_i(x_i, y_i, z_i)T_i(q)\delta r_i \tag{2.5}$$

Considering virtual nodal displacements δu_i and virtual displacements $\delta(q)$ of

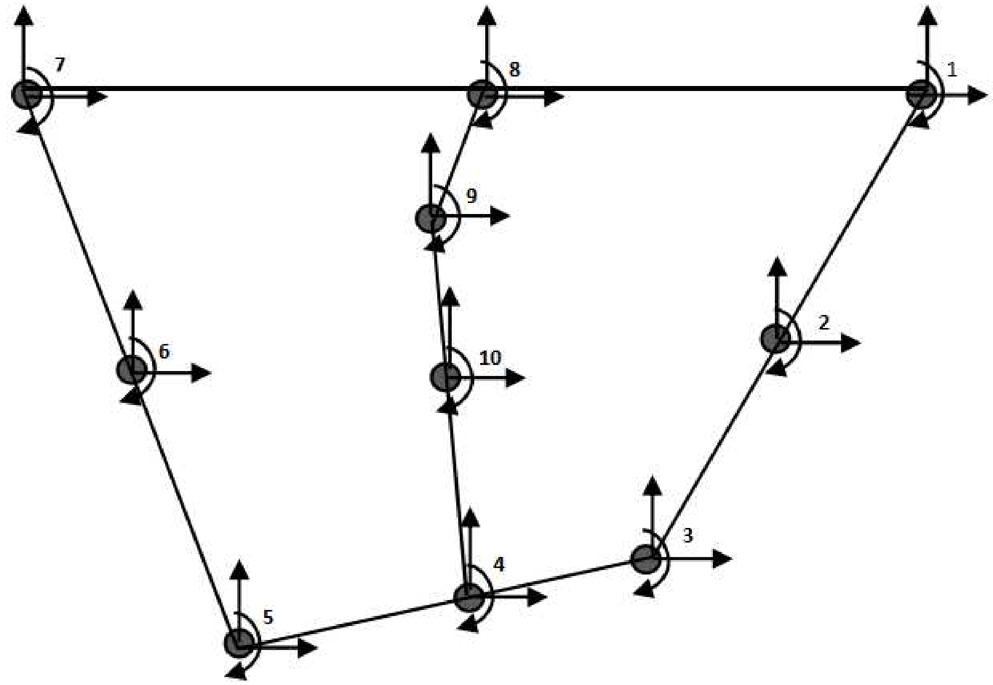


Figure 2.2. nodes of the four-bar linkage system

the generalized co-ordinaes, the virtual displacement in the fixed reference frame becomes

$$\begin{aligned} \delta p_i = R_i(q)N_i(x_i, y_i, z_i)T_i(q)\delta r_i + \delta R_i(q)N_i(x_i, y_i, z_i)T_i(q)u_i \\ + R_i(q)N_i(x_i, y_i, z_i)\delta T_i(q)u_i + R_i(q)N_i(x_i, y_i, z_i)T_i(q)\delta u_i \end{aligned} \tag{2.6}$$

Focusing on only higher order magnitudes, equation 2.6 simplifies to

$$deltap_i = R_i(q)N_i(x_i, y_i, z_i)T_i\delta r_i + R_i(q)N_i(x_i, y_i, z_i)T_i(q)\delta u_i \tag{2.7}$$

To obtain the related acceleration we have to differentiate equation 2.7 three times. This is the acceleration of a generic point in the ith finite element. Thus:

$$\begin{aligned} \ddot{p}_i = R_i(q)N_i(x_i, y_i, z_i)T_i(q)\delta\ddot{r}_i + R_i(q)N_i(x_i, y_i, z_i)T_i(q)\ddot{u}_i \\ + 2(\dot{R}_i(q)N_i(x_i, y_i, z_i)T_i(q) + R_i(q)N_i(x_i, y_i, z_i)\dot{T}_i(q))\dot{u}_i \\ + (R_i\ddot{}(q)N_i(x_i, y_i, z_i)T_i(q) + 2\dot{R}_i(q)N_i(x_i, y_i, z_i)\dot{T}_i(q) \\ + R_i(q)N_i(x_i, y_i, z_i)\ddot{T}_i(q))u_i \end{aligned} \tag{2.8}$$

Focusing on the higher order of magnitude as above, a simplified expression becomes

$$\begin{aligned} \ddot{p} = R_iN_i(x_i, y_i, z_i)T_i\ddot{r}_i + R_iN_i(x_i, y_i, z_i)T_i\ddot{u}_i \\ + 2(\dot{R}_iN_i(x_i, y_i, z_i)T_i + R_iN_i(x_i, y_i, z_i)\dot{T}_i)\dot{u}_i \end{aligned} \tag{2.9}$$

Now by grouping all the kinematic entities of every finite element in the system into a vector, equation 2.2 differentiated will be

$$db = du + dr. \tag{2.10}$$

Position, velocity and acceleration of ERLS depend on the generalized co-ordinate(s) and are expressed as thus:

$$dr = S(q)dq, \tag{2.11}$$

$$\dot{r} = S(q)\dot{q}, \tag{2.12}$$

$$\ddot{r} = S(q)\ddot{q} + \dot{S}(q,\dot{q})\dot{q} = S(q)\ddot{q} + \left(\sigma_k \dot{q}_k \frac{\partial S}{\partial q_k}\right), \tag{2.13}$$

where $\mathbf{S(q)}$ represents the sensitivity coefficients matrix for all nodes. This matrix's elements represent the rigid-body velocities at the nodes per unit velocity of the generalized co-ordinates of the ERLS. Substituting equation 2.11 into equation 2.10, it is obtained thus

$$db = \begin{bmatrix} I & S \end{bmatrix} \begin{bmatrix} du \\ dq \end{bmatrix} \tag{2.14}$$

In equation 2.14, a square coefficient matrix is necessary such that when db has a minute nodal displacement, it does not imply many sets of increments of the generalized co-ordinates of the system. As suggested in [**?**], the easiest way to eliminate this redundancy is to force to zero a number of elements of du equal to the number of generalized co-ordinates of the ERLS. Partitioning du and S are partitioned into independent and zeroed parts, signified by subscripts *in* and *0* respectively, the element forced to zero could be eliminated as thus

$$db = \begin{bmatrix} I & S_{in} \\ 0 & S_0 \end{bmatrix} \begin{bmatrix} du_{in} \\ dq \end{bmatrix} \tag{2.15}$$

In the coefficient matrix, S_0 should have a non-zero determinant so that the matrix is non-singular; and the generalized coordinates of the ERLS should be chosen such that singularities are not in the motion of the system. Following the kinematic definitions, the principle of virtual work is applied to obtain the dynamic equations:

$$dW^{inertia} + dW^{elastic} + dW^{external} = 0 \tag{2.16}$$

expressing it explicitly,

$$\sum_i \int_{vi} P_i^T \ddot{p}_i \rho_i dv + \sum_i \int_{vi} \delta\varepsilon_i^T D_i \varepsilon_i dv = \sum_i \int_{vi} \delta p_i^T g p_i dv + (\delta u^T + \delta r^T) f, \quad (2.17)$$

where the first and second terms on the left hand side respectively are $\delta W^{inertia}$ and $\delta W^{elastic}$, and the right hand side is $-\delta Wexternal$. D_i, ε_i, ρ_i respectively signify stress-strain matrix, strain vector, and mass density for the ith element. g, f are gravity acceleration vector and external forces and torques vector, respectively. vi is the ith element volume. The following terms can be introduced,

$$\int_{vi} T_i^T N_i^T R_i N_i T_i \rho dv = M_i, \quad (2.18)$$

$$\int_{vi} T_i^T B_i^T D_i B_i T_i dv = K_i, \quad (2.19)$$

$$\int_{vi} T_i^T N_i^T R_i^T g \rho_i dv = f_g, \quad (2.20)$$

$$\int_{vi} T_i^T N_i^T R_i^T \dot{R}_i N_i T_i \rho_i dv = M_{G1i}, \quad (2.21)$$

$$\int_{vi} T_i^T N_i^T R_i^T R_i N_i \dot{T}_i \rho_i dv = M_{G2i}, \quad (2.22)$$

$$\delta T_i^T = \delta\phi_i T_i^T, \quad (2.23)$$

The the virtual work equation above can be put in the form

$$\begin{aligned} &\sum_i \delta u_i^T M_i(\ddot{r}_i + \ddot{u}_i) + 2\sum_i \delta u_i^T (M_{G1i} \\ &\quad + M_{G2i})\dot{u}_i + \sum_i \delta r_i^T M_i(\ddot{r}_i + \ddot{u}_i) + 2\sum_i \delta r_i^T (M_{G1i} + M_{G2i})\dot{u}_i \\ &\quad + \sum_i \delta u_i^T K_i u_i + \sum_i u_i^T \delta\phi_i K_i u_i = \\ &\quad \sum_i (\delta u_i^T + \delta r_i^T) f_{gi} + (\delta u^T + \delta r^T) f. \quad (2.24) \end{aligned}$$

From the foregoing is obtained two dynamic equations for the motion of the node and the elastic displacement:

$$du^T M(\ddot{r} + \ddot{u}) + 2du^T M_G \dot{u} + du^T K u = du^T (f_g + f) \tag{2.25}$$

$$dr^T M(\ddot{r} + \ddot{u}) + 2dr^T M_G \dot{u} + du^T K u = dr^T (f_g + f) \tag{2.26}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{M}_G = M_{G1} + M_{G2}$ represents the Coriolis matrix, K the stiffness matrix of the system; $\mathbf{f}_g$ is the gravity vector and $\mathbf{f}$ represents the vector of external loads applied to the system. The infinitesimal displacements of the ERLS can be expressed by means of the sensitivity coefficient matrix, so that $\delta\mathbf{u}$'s and $\delta\mathbf{r}$'s can be eliminated from equations (2.25) and (2.26). Moreover, simple Rayleigh damping could be introduced, so that the following system of differential equations is obtained:

$$M(\ddot{r} + \ddot{u}) + 2M_G \dot{u} + \alpha M \dot{u} + \beta K \dot{u} + K u = f_g + f \tag{2.27}$$

$$S^T M(\ddot{r} + \ddot{u}) + 2S^T M_G \dot{u} + \alpha S^T M \dot{u} = S^T (f_g + f) \tag{2.28}$$

Equations (2.27) and (2.28) can be rearranged in matrix form discarding the equations of the elastic DOF's forced to zero in order to get an explicit integration scheme:

$$\begin{bmatrix} M & MS \\ S^T M & S^T M S \end{bmatrix} \begin{bmatrix} \ddot{u} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} t(u, \dot{u}, q, \dot{q}) \\ S^T t(u, \dot{u}, q, \dot{q}) \end{bmatrix}. \tag{2.29}$$

2.0.2 Model Linearization

The nonlinear model above needs to be linearized to get a state-space linear model capable of reproducing the dynamic behaviour of a flexible link mechanism about an equilibrium point[17] so that system theory can be applied to it. Defining the state vector of the system as $x = \begin{bmatrix} \dot{u} & \dot{q} & u & q \end{bmatrix}^T$, the system dynamic equation model (2.29) can be put into the following state space representation:

$$\begin{bmatrix} M & MS & 0 & 0 \\ S^T M & S^T MS & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{u} \\ \ddot{q} \\ \dot{u} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -2M_G - \alpha - \beta K & -M\dot{S} & -K & 0 \\ S^T(-2M_G - \alpha M) & S^T MS & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{q} \\ u \\ q \end{bmatrix} + \begin{bmatrix} M & I \\ S^T M & S^T \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_g \\ f \end{bmatrix}. \tag{2.30}$$

In a more compact form:

$$A(x)\dot{x} = B(x)x + C(x)v. \tag{2.31}$$

Matrices A,B, and C depend on x, while the state vector x and the input v depend on time. The system of differential equations showed in (2.30) is nonlinear due to the quadratic term in S matrix $(\dot{q}^2)$. Linearization is done choosing an equilibrium point such that $x = x_e$, $v = v_e$ and $\dot{x_e} = 0$, and we can write: $x(t) = x_e + \Delta x(t), v(t) = v_e + \Delta v(t)$. Adopting the approximation $A(x_e + \Delta x)\Delta\dot{x} \cong A(x_e)\Delta\dot{x}$, Equation (2.31) becomes:

$$A(x_e)\Delta\dot{x} = B(x_e + \Delta x)(x_e + \Delta x) + C(x_e + \Delta x)(v_e + \Delta v). \tag{2.32}$$

Considering the ith row,

$$\begin{aligned} &\sum_{j=1}^{n} A_{i,j}(x_{e1} + \Delta x_1, x_{e2} + \Delta x_2, ..., x_{en} + \Delta x_n) * \Delta\dot{x}_j \\ &\qquad = \sum_{j=1}^{n} A_{i,j}(x_{e1}, x_{e2}, ..., x_{en}) * \Delta\dot{x}_j + \sum_{k=1}^{n} [\sum_{j=1}^{n} \frac{\partial A_{i,j}}{\partial x_k}|_{x=x_e} \Delta x_k \Delta\dot{x}_j] \\ &\qquad\qquad = \sum_{j=1}^{n} A_{i,j}(x_{e1}, x_{e2}, ..., x_{en}) * \Delta\dot{x}_j \end{aligned} \tag{2.33}$$

where the higher order term $\sum_{k=1}^{n}[\sum_{j=1}^{n}\frac{\partial A_{i,j}}{\partial x_k}|_{x=x_e}\Delta x_k \Delta\dot{x_j}]$ is negligible. Similar considerations is applied to the right hand side terms of 2.32 relating to B_i and C_i. For B_i-related term, a more detailed expression is obtained as

$$\sum_{k=1}^{n}\Big[\sum_{j=1}^{n}\frac{\partial B_{i,j}}{\partial x_k}|_{x=x_e}x_{ej}\Big]\Delta x_k = \left[\sum_{j=1}^{n}\frac{\partial B_{i,j}}{\partial x_1}|_{x=x_e}x_{ej}\cdots\sum_{j=1}^{n}\frac{\partial B_{i,j}}{\partial x_n}|_{x=x_e}x_{ej}\right]\begin{bmatrix}\Delta x_1\\ \cdots\\ \Delta x_n\end{bmatrix}$$
$$= \left[[\frac{\partial B_{i,1}}{\partial x_1}\cdots\frac{\partial B_{i,n}}{\partial x_1}]_{x=xe}\begin{bmatrix}x_{e1}\\ \cdots\\ x_{en}\end{bmatrix}\cdots\left[\tfrac{\partial B_{i,1}}{\partial x_n}\quad\cdots\quad\tfrac{\partial B_{i,n}}{\partial x_n}\right]\begin{bmatrix}x_{e1}\\ \cdots\\ x_{en}\end{bmatrix}_{x=xe}\right]\begin{bmatrix}\Delta x_1\\ \cdots\\ \Delta x_n\end{bmatrix}$$
$$= \left[\left[\left[\tfrac{\partial Bi,1}{\partial x_1}\quad\cdots\quad\tfrac{\partial B_{i,n}}{\partial x_1}\right]\quad\cdots\quad\left[\tfrac{\partial Bi,1}{\partial x_n}\quad\cdots\quad\tfrac{\partial B_{i,n}}{\partial x_n}\right]\right]_{x=xe}\otimes\begin{bmatrix}x_{e1}\\ \cdots\\ x_{en}\end{bmatrix}\right]\begin{bmatrix}\Delta x_1\\ \cdots\\ \Delta x_n\end{bmatrix} \tag{2.34}$$

A similar way is taken to explicitly express also the C_i-related term. As equilibrium point implies

$$B(x_e)S_e + C(x_e)v_e = A(x_e)\dot{x_e} = 0, \tag{2.35}$$

After some algebra it is obtained:

$$A(x_e)\Delta\dot{x} = \left[B(x_e) + (\frac{\partial B}{\partial x}|_{x=x_e}\otimes x_e) + (\frac{\partial C}{\partial x}|_{x=x_e}\otimes v_e)\right]\Delta x + C(X_e)\Delta v$$
$$= \tilde{B}(x_e, v_e)\Delta x + C(x_e)\Delta v, \tag{2.36}$$

where the symbol "$\otimes$" indicates the inner product of $[\partial B_{i,1}/\partial x_j \ldots \partial B_{i,n}/\partial x_j]_{x=x_e}]$ and $[\partial C_{i,1}/\partial x_j \ldots \partial C_{i,n}/\partial x_j]_{x=x_e}]$ with, respectively, x_e and v_e, for all the subscripts 'i' and 'j'. After setting an operating point x_e, the matrices $A(x_e), B(x_e)$ and $C(x_e)$ are calculated and the terms

$$\left(\frac{\partial B}{\partial x}\Big|_{x=xe}\otimes x_e\right) \text{ and } \left(\frac{\partial C}{\partial x}\Big|_{x=xe}\otimes v_e\right)$$

By setting $F = A^{-1}\tilde{B}$ and $G = A^{-1}C$ a standard form for the linearized equation can be written as:

$$\Delta\dot{x} = F\Delta x + G\Delta v. \tag{2.37}$$

Chapter 3

Physical Modeling and Analysis

Pbysical Modeling is a method of modeling systems with the use real physical components. Physical modeling allows for effective faster design and testing of engineering systems more than before. The physical modeling approach allows one to describe the physical structure of a system instead of the inherent mathematics. Moreover, in some cases, the analytical equations modeling is cumbersome and thus physical modeling could be very pertinent in this case.

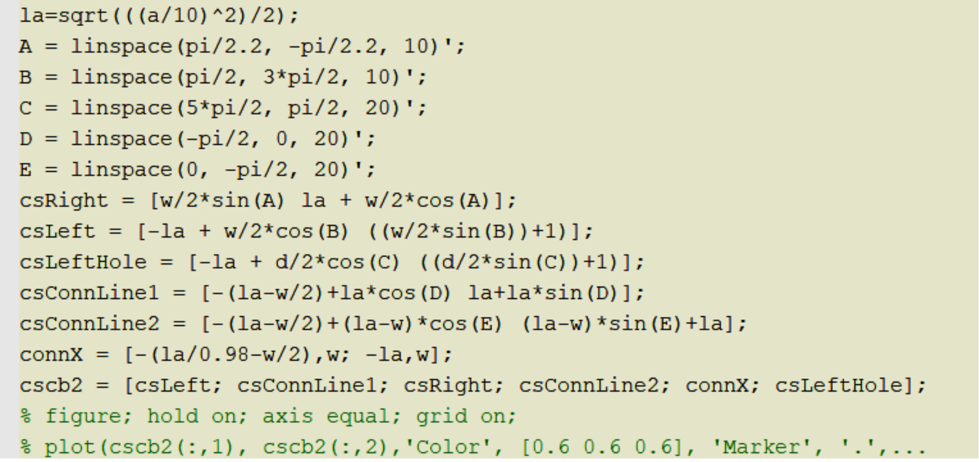
Physical models of systems can be implemented to study system dynamics and for control system design. The physical model represents the system under study with respect to its physical parts and assembles these different parts to form the complete system; then the system behaviour is simulated. A number of packages exist that are equipped for physical modeling of systems. In this work, the Mathworks package is choosen and used. Specifically, the Simscape Multibody^{TM} toolbox is used since what is dealt with in the work is multibody systems. Mathworks Simscape Multibody, formerly SimMechanics, is used for physical modeling of systems. It is also flexible as it allows the import of CAD designs in an external software into the Simscape environment for cases where the user prefers to make a CAD design on another platform or also to avoid repetition of the modeling process if it happens that the user already has a CAD model. The modeled system can be integrated with control systems design in Simulink^{TM}. In this section, the physical modeling process of the case study four-bar linkage system with Simscape, building part dimensions and assembling them to form a unit, is presented. Moreover, the main focus is on the *flexibility modeling* with regard to the flexible-links of multibody systems.

Mathworks physical modeling tools enables to:

- Assemble system-level models that span multiple physical domains and include the control system in a single environment.
- Create reusable models of physical system with physical ports, in addition to input and output signals.
- Model custom physical components (mechanical, electrical, and other physical domains) using a MATLAB based physical modeling language.
- Extend analysis with 3D visualization and additional simulation methods. [31]

3.0.1 Multibody Physical Modeling in Simscape

The dynamic modeling of a multibody system by a Simscape physical model involves the description of the actual physical characteristics of the system from which the full 3-dimensional body is built as it would appear in reality. At this level, the description of the 2-D section of individual bodies of the system are made with a Matlab code and stored as a script. In figure 3.1, a snippet of the matlab description of a physical body's dimensions is shown alongside the import of this description to Simscape to shape a generic body mass, called *general extrusion*, into the exact parameters contained in the description. After the coding of the body dimension, it is observed using a plot to verify the correctness of the design. Figure 3.2 shows a sample plot of a curved beam section with finite segements. In figure 3.3 the two 2-D dimension is imported to shape a general extrusion mass block to define the final body. The material properties are also set by inserting the density of the choice material, for instance Aluminum, the mass and moment properties are updated. This process is repeated for the other individual bodies that make-up the multibody system. In the end, the bodies are connected using appropriate joints as necessary.



```
la=sqrt(((a/10)^2)/2);
A = linspace(pi/2.2, -pi/2.2, 10)';
B = linspace(pi/2, 3*pi/2, 10)';
C = linspace(5*pi/2, pi/2, 20)';
D = linspace(-pi/2, 0, 20)';
E = linspace(0, -pi/2, 20)';
csRight = [w/2*sin(A) la + w/2*cos(A)];
csLeft = [-la + w/2*cos(B) ((w/2*sin(B))+1)];
csLeftHole = [-la + d/2*cos(C) ((d/2*sin(C))+1)];
csConnLine1 = [-(la-w/2)+la*cos(D) la+la*sin(D)];
csConnLine2 = [-(la-w/2)+(la-w)*cos(E) (la-w)*sin(E)+la];
connX = [-(la/0.98-w/2),w; -la,w];
cscb2 = [csLeft; csConnLine1; csRight; csConnLine2; connX; csLeftHole];
% figure; hold on; axis equal; grid on;
% plot(cscb2(:,1), cscb2(:,2),'Color', [0.6 0.6 0.6], 'Marker', '.',...
```

Figure 3.1. Physical model sample code snippet

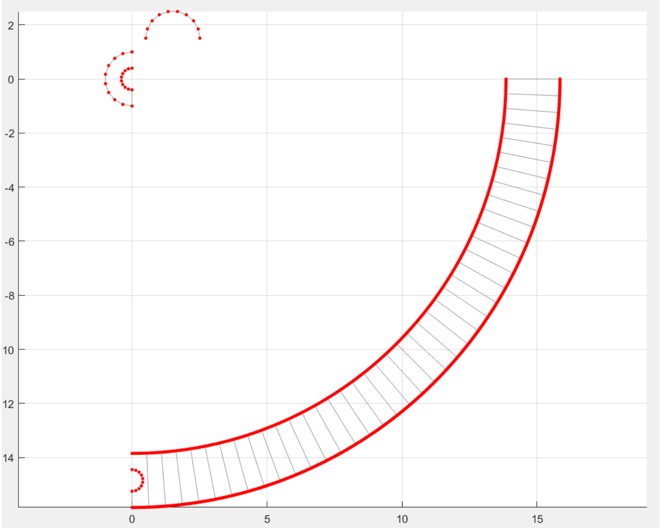
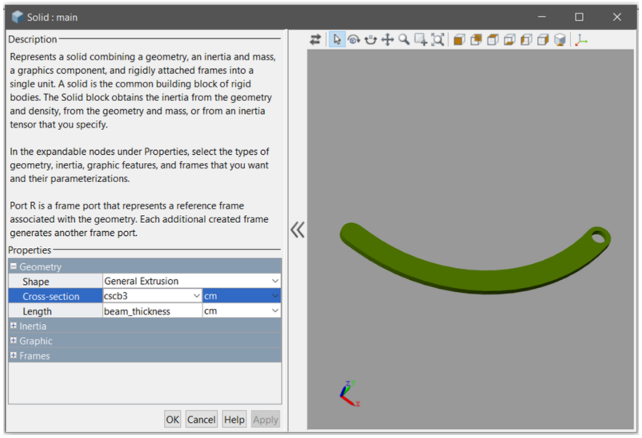


Figure 3.2. Beam section plot

3.0.2 Modeling Flexible Bodies in SimMechanics and Simulink

Rigid-body motion as well as flexible body motion is simulated combining Simscape and Simulink forming an efficient tool for control systems applications. SimMechanics simulates flexibility with a fidelity that is sufficient for many applications such as control design. Two most common method for flexible beam modeling are: lumped-parameter method and Finite-element import method. Both capture



Description

Represents a solid combining a geometry, an inertia and mass, a graphics component, and rigidly attached frames into a single unit. A solid is the common building block of rigid bodies. The Solid block obtains the inertia from the geometry and density, from the geometry and mass, or from an inertia tensor that you specify.

In the expandable nodes under Properties, select the types of geometry, inertia, graphic features, and frames that you want and their parameterizations.

Port R is a frame port that represents a reference frame associated with the geometry. Each additional created frame generates another frame port.

Properties

Geometry

Shape	General Extrusion	
Cross-section	cscb3	cm
Length	beam_thickness	cm

Inertia

Graphic

Frames

Figure 3.3. cross section build

deformations that are small, linear and elastic.[33]

Lumped-parameter method treat the flexible body as a collection of discrete flexible units. Each flexible unit comprises two or more rigid mass elements coupled by joints with internal springs and dampers. The joints provide the degrees of freedom required for deoformation to occur. The mass, spring, and damper elements provide the inertial, restorative, and dissipative forces that collectively account for deformation.

Finite-element import method treat the flexible body as the superposition of distinct rigid-body and deformation models. The rigid-body model captures the motion of the body as though it were incapable of deforming. The deformation model captures the deflections with the body as though it were fixed in place. The process here involves the use of an external finite-element analysis software to to perform finite-element analysis and then the results are imported into Simulink-Simscape.

A detailed description for both methods can be seen in [29]. For the work presented, the lumped-parameter method (LPM) is used. Some description, therefore, is provided about the modeling using this method. The lumped-parameter method is suited for bodies with slender geometries, such as rods and beams.[29] This method approximates a flexible body as a collection of interconnected discrete flexible units; with each unit comprising two mass elements coupled by a joint having internal springs and dampers. The deformation modes of the flexible unit are captured by the joint degrees of freedom while the stiffness and damping characteristics are captured by the springs and dampers. Figure (3.4) shows the element discretization according to the lumped-parameter method. In each flexible beam unit, there are two mass elements (m), spring (k), and damper (b). The derivation of the parameters for the system blocks used in the LPM is shown in the following. Consider a generic beam under longitudinal forces. With one translational degree of freedom, a flexible beam follows a damped harmonic oscillator motion with the motion equation:

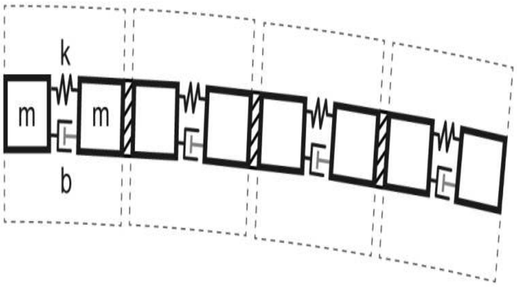


Figure 3.4. Lumped-parameter method structure for flexible body

$$m\ddot{u} + b_T\dot{u} + k_T u = f, \tag{3.1}$$

where f is the net external force action on the translating mass element and u is the translational offset of the mass element from its equilibrium position. m,b_T, and K_T are mass of a mass element, joint translational damping coefficient, and joint translational spring coefficient. If the translational degree of freedom is replaced with a rotational one, the equation becomes

$$I\ddot{\theta} + b_R\theta + K_R\theta = \tau, \tag{3.2}$$

where τ is the net external torque acting on the rotating mass element and θ is the rotational offset of the element equilibrium position. I, b_R, and k_R are the moment of inertia of a mass element about the rotation axis, the rotational damping coefficient of the joint, and the rotational spring coefficient of the joint, respectively. In terms of damping ratio (ζ) and natural frequency (w), the rotational equation is,

$$I(\ddot{\theta} + 2\zeta w\dot{\theta} + w^2\theta) = \tau, \tag{3.3}$$

where

$$\zeta = \frac{br}{2\sqrt{Ik_R}} \text{ and } w = \sqrt{\frac{k_R}{I}} \tag{3.4}$$

In a beam such as the cantilever beam, subjected to bending, spring coefficient follows from the equality between the spring torque at the joint and the bending moment on a continuous version of the flexible beam. From Hooke's law,

$$\tau_k = k_r\theta, \tag{3.5}$$

where τ_k is the spring torque, k_R is the rotational spring constant, and θ is the deflection angle. The bending moment on a continuous beam unit is

$$M = \frac{EI_A}{R}, \tag{3.6}$$

where M is the bending moment, E is Young's modulus of elasticity, I_A is the second moment of area, and R is the bending radius of curvature. For small deflections, θ reduces to l/R, where l is the undeformed length of a flexible beam, thus spring coefficient is

$$k_R = \frac{EI_A}{l}. \tag{3.7}$$

For split beam with N flexible beam units and length L, l is then

$$l = \frac{L}{N} \tag{3.8}$$

In figure (3.5) is shown the radius of curvature (R) due to bending and angle (θ); and mass element is half the length of the flexible beam. To define damping coefficient, it is assumed that damping is linear and bound by a constitutive law

$$\tau_b = b_R\dot{\theta} \tag{3.9}$$

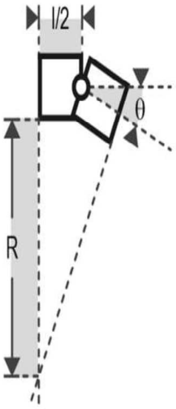


Figure 3.5. Geometry of flexible beam unit with one rotational degree of freedom

with τ_k as the magnitude of the damping torque between the two mass elements coupled by a rotational degree of freedom. As a first-order approximation, the damping coefficient is taken to be proportional to the spring coefficient:

$$b_R = \alpha k_R \tag{3.10}$$

The proportionality constant (α) is an empirically set damping factor. The damping factor can be set by matching the lumped-parameter deformations to reliable benchmark data.[29][34] The discretization level of the beam scales the damping coefficient.

3.0.3 Model of Four-bar Linkage System

The Simscape model of the FBL system starts with the defining and creating the cross-sections of the different parts of the system and then connection the system with joints. Figure (3.6) shows the model of the system. For each link in the system a flexible beam equivalent is created for different simulation configuration studies. The parts (rigid and flexible) are modeled in the FBL system. In figure (3.8), the LPM modeling flexible beam is shown. The parameters for the mass elements and couplings are defined. For the LPM, the number of mass elements affect the quality of the approximation of the material flexibility behaviour. However, at a

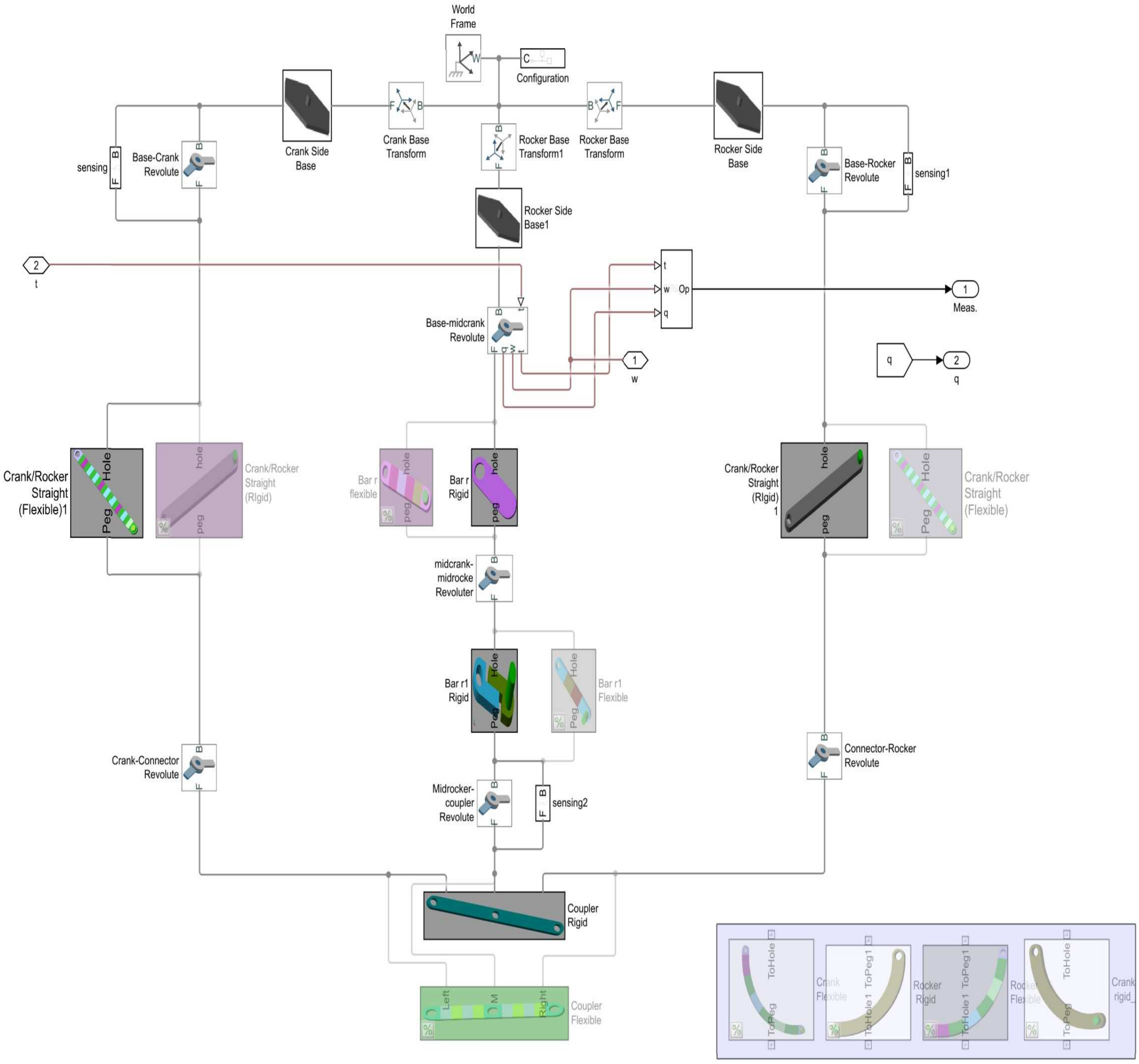


Figure 3.6. FBL plant model

certain amount, the effect of increasing the mass element does not make changes or improvement.[33] For the longest length beams, the number of mass element was increased to 20, with the lowest being 16; except for the shortest beam, link r, the actuated link, which had 10 mass elements. As referenced also in following section, the stiffness of the flexible beam was reduced to study the impact of flexible

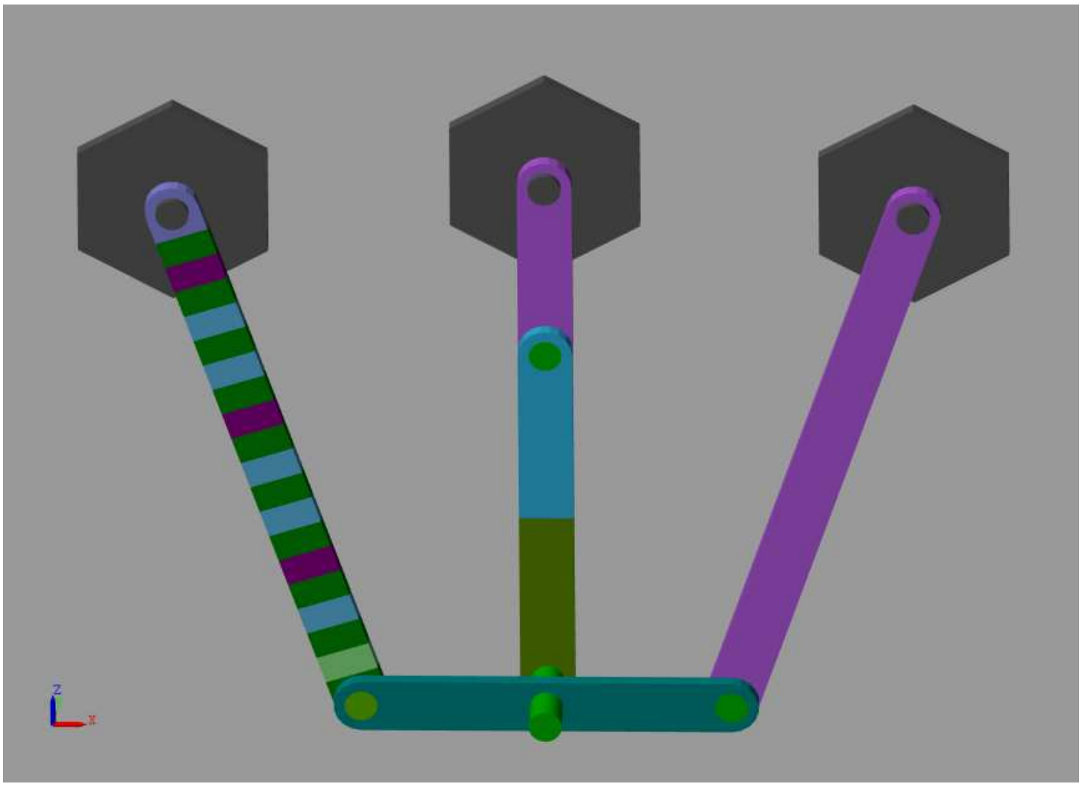


Figure 3.7. Model Simscape 3D Visualization

beam.png

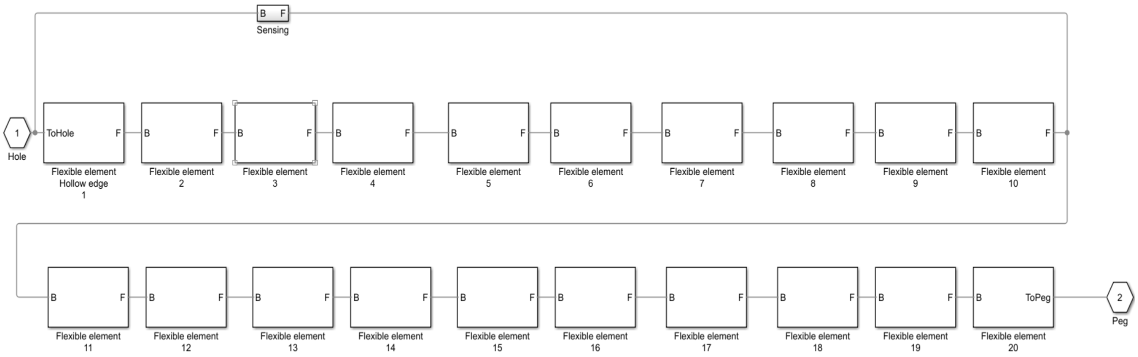


Figure 3.8. Flexible beam model

beams for the novel FBL system design. Each link is made up of two mass element and a joint. This implies that there is in effect 40 discretized mass elements for instance for the the longest beams. 3.10 shows the mass element definition. The

beam2.png

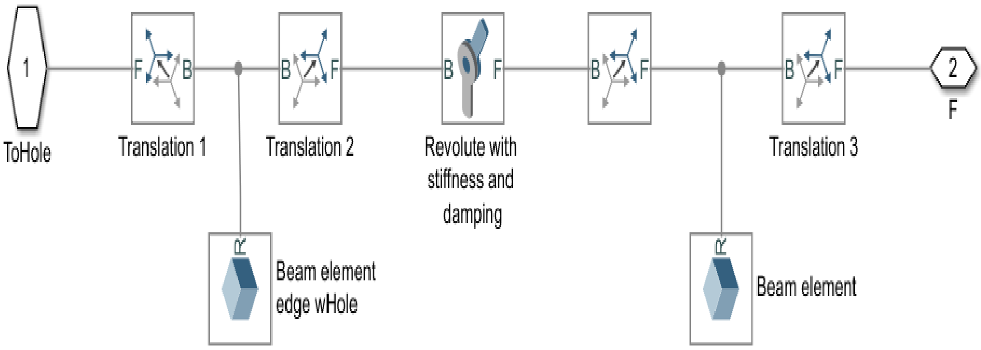
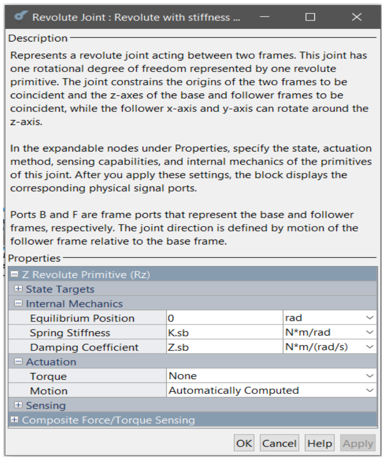


Figure 3.9. Flexible beam model

revolute joint that connects each of the mass element couple is parameterized with the result of the calculated damping coefficient and stiffness values based on the number of the mass element and material property used. To observe the deflections



Revolute Joint : Revolute with stiffness ...

Description

Represents a revolute joint acting between two frames. This joint has one rotational degree of freedom represented by one revolute primitive. The joint constrains the origins of the two frames to be coincident and the z-axes of the base and follower frames to be coincident, while the follower x-axis and y-axis can rotate around the z-axis.

In the expandable nodes under Properties, specify the state, actuation method, sensing capabilities, and internal mechanics of the primitives of this joint. After you apply these settings, the block displays the corresponding physical signal ports.

Ports B and F are frame ports that represent the base and follower frames, respectively. The joint direction is defined by motion of the follower frame relative to the base frame.

Properties

⊟ Z Revolute Primitive (Rz)		
⊞ State Targets		
⊟ Internal Mechanics		
Equilibrium Position	0	rad
Spring Stiffness	K.sb	N*m/rad
Damping Coefficient	Z.sb	N*m/(rad/s)
⊟ Actuation		
Torque	None	
Motion	Automatically Computed	
⊞ Sensing		
⊞ Composite Force/Torque Sensing		

Figure 3.10. Flexible beam model

of the flexible links the *transform sensor*3.11 block is used. In the block properties of the transform sensor, the axis to is desired to be recorded are selected and thus the block provides output channels to record them. The simulink signals are not the same as the physical signals directly measured from the physical model and therefore in every case the physical signals have to be converted with signal transform block. The signal transform block is of two modes. One is the PS-Simulink block, and the other is Simulink-PS block. Respectively they are used to convert physical signals directly from physical modeling blocks to format that can connected to Simulink, and to convert Simulink signals to physical model signals. It is crucial to use this block and to configure each blocks unit specification with respect to the signal connected. For linear analysis process as shown in the later part, the signals selected for input and output must be Simulink signals and not physical signals.

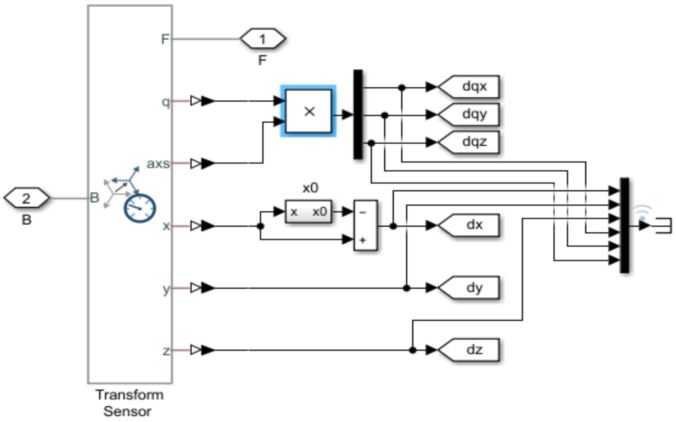


Figure 3.11. Flexible beam model

Actuator model

Figure (3.12) shows the actuator model integrated to the FBL system model. In Simulink, the parameters of the actuator can be set and used to represent the transfer function; but also another option is to use Simscape to make an equivalent physical model of the plant. The later is implemented, driven by a PWM signal and H-bridge.

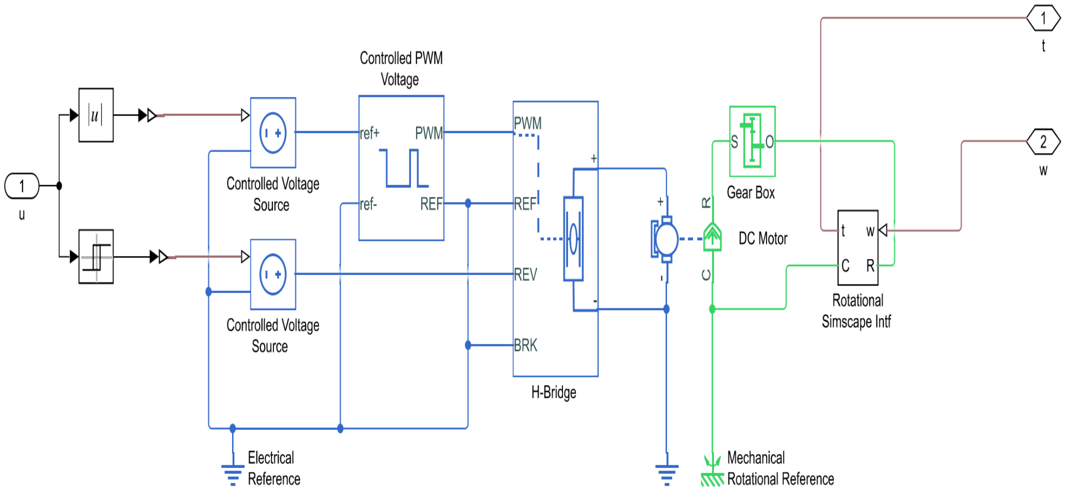


Figure 3.12. integrated actuator model

3.0.4 Visualization

The Simscape tool is equipped with a 3D visualization medium, the mechanics explorer. Simscape Multibody is based on Simulink hence graphs are available for simulation outcomes but these graphs are rather abstract. With the animation property it could help to interpret the results faster and understand them better. The visualization property gets even more interesting with the increase in the complexity of the modeled system.

3.0.5 Pre-control Analysis

Having a physical model of the dynamic behaviour of the multibody system, desired analysis can be performed through the simulation of the system. At this point, a control system has not been designed and connected. The modelled system can be analysed to observe visually (through the 'mechanics explorer' feature of Simscape) or by signal plots the behaviour of the system. For example, it could be desired or observe the effect of gravity on the four-bar linkage mechanism with the purpose to gain insight into the torque requirements to balance gravity force, etc. To get an idea of the torque demand of the mechanism in order to choose the appropriate actuator system, a simple reverse analysis is performed where the angle of the actuator is system input and the torque required to make the demanded transition is recorded. For analysis like the trimming of the model, forward dynamics is performed where external torques and forces are applied and the formulation and integration of the equations of motion and solution for the system motion is made.

Chapter 4

Control Design

For multibody systems and systems in general, different control architectures could be used as seen in the literature review. With a physical model approach to the dynamic model, the control of flexible link multibody systems is not limited to any method(s) of control. In this section, it is presented the control of the multibody system of the FBL mechanism via two schemes.

Proportional-Integral-Derivative (PID) Control

Here the intent is to use '*PID*' to represent not just the literal PID system but the other combination structures namely P, PI, PD, and PID. For the FBL mechanism, the different structures are tested to choose the best-performing. PID controllers are largely in use today. It is interesting to note that most industrial controllers in use today are PID controllers or modified PID controllers. The usefulness of PID controls lies in their general applicability to most control systems.[35]. The error signal which is difference between desired state and the feedback state is passed through, as the configuration case may be, a proportional action, an integral action and/or derivative action, with the sum of the effect(s) producing the control signal then feed to the plant system[37].

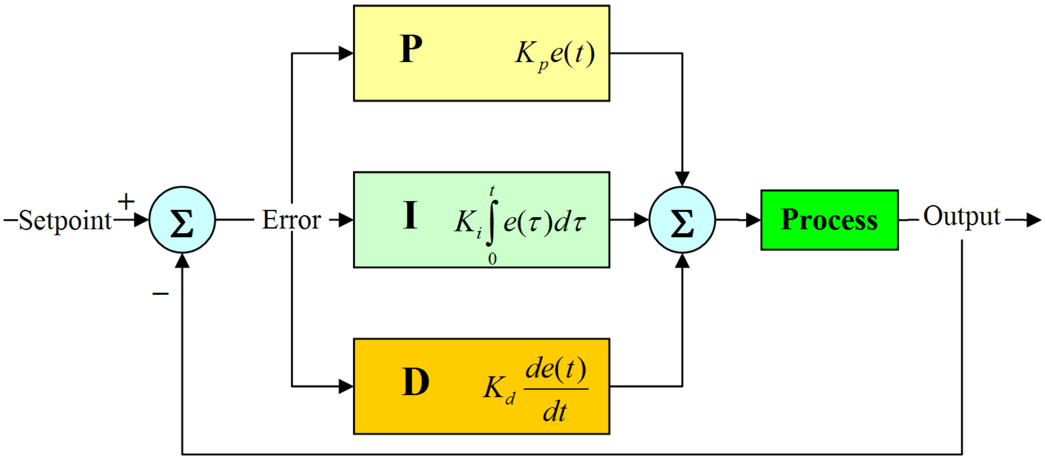


Figure 4.1. Generic PID architecture

Linear Quadratic Optimal Control

Linear quadratic optimal control is a feedback controller that provides an algorithm to minimize a quadratic cost function - such that dynamic system controlled operates at a minimum cost as specified by the cost function. For a system with the dynamics $\dot{x} = Ax + Bu$, the Linear quadratic (LQ) scheme calculates an optimal control gain matrix, K, such that the state feedback law $u = -Kx$ minimizes the quadratic cost function

$$J(u) = \int_0^\infty (x^T Qx + u^T Ru + 2X^T Nu)dt \tag{4.1}$$

where Q and N are symmetric positive semi-definite nxn matrices, R is a symmetric positive definite mxm matrix.

Kalman Estimator

Given the state-space model of the system and the process and measurement noise covariance data, the Kalman estimator provides an optimal solution to the continuous-time or discrete-time estimation problems. In this case we look at the

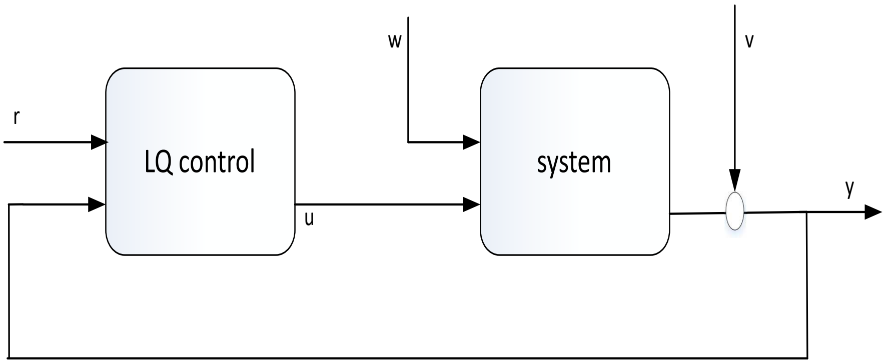


Figure 4.2. LQ control architecture

continuous-time estimation. Given the system equations for state and output

$$\dot{x} = Ax + Bu + Gp \tag{4.2}$$

$$y = Cx + Du + Hp + m \tag{4.3}$$

with inputs u known, and measurement noise, m and process noise, p white and such that it has zero mean, $E(p) = E(m) = 0$ and we also have variance and covariance $E(pp^T) = Q, E(mm^T) = R, E(pm^T) = N$ and it is to be constructed a state estimate $\hat{x}(t)$ such that minimizes covariance of steady-state error

$$P = \lim_{t\to\infty} E(x - \hat{x}x - \hat{x}^T) \tag{4.4}$$

The optimal solution is the Kalman filter with equations

$$\hat{x} = A\hat{x} + Bu + K(y - C\hat{x} - Du) \tag{4.5}$$

$$\begin{bmatrix} \hat{y} \\ \hat{x} \end{bmatrix} = \begin{bmatrix} C \\ I \end{bmatrix} \hat{x} + \begin{bmatrix} D \\ 0 \end{bmatrix} u \tag{4.6}$$

The filter gain K is determined by solving the algebraic Riccati equation

$$K = (PC^T + \bar{N})\bar{R}^{-1} \tag{4.7}$$

where

$$\bar{R} = R + HN + N^T H^T + HQH^T \tag{4.8}$$
$$\bar{N} = G(QH^T + N) \tag{4.9}$$

The output estimate $\hat{y}$ and state estimate $\hat{x}$ are generated. As may be required the output estimate can be corrected using the actual plant output. In figure 4.3, v_1 and v_2 the white process and measurement noises.

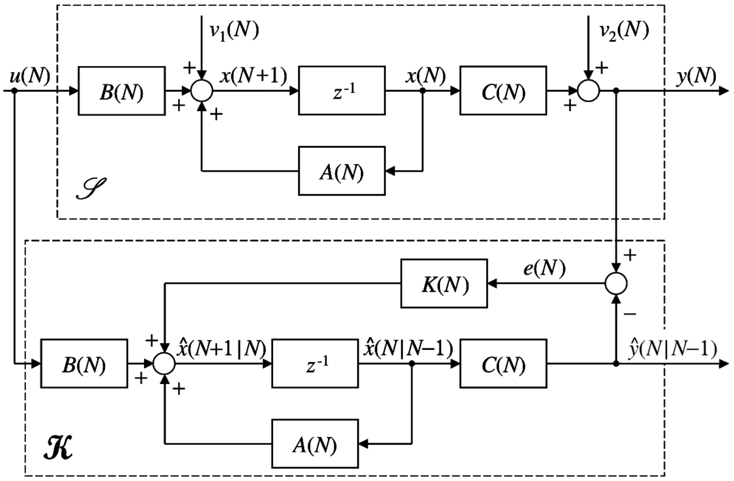


Figure 4.3. Kalman estimator

4.0.1 Control of the FBL System

PID scheme

The control system for the FBL system is shown in (4.4). The use of flexible, light, materials imply an overall lighter mechanism which can thus run at faster speed and take less torque. A PD-based closed-loop control is implemented. The close-loop feedback is the the position of the coupler link which is the angle it makes with the horizontal. With regard the intention of the novel FBL architecture, the coupler link moves from a horizontal to vertical position and vice-versa while the mechanism is mounted on an second structure like a robot to pick and place

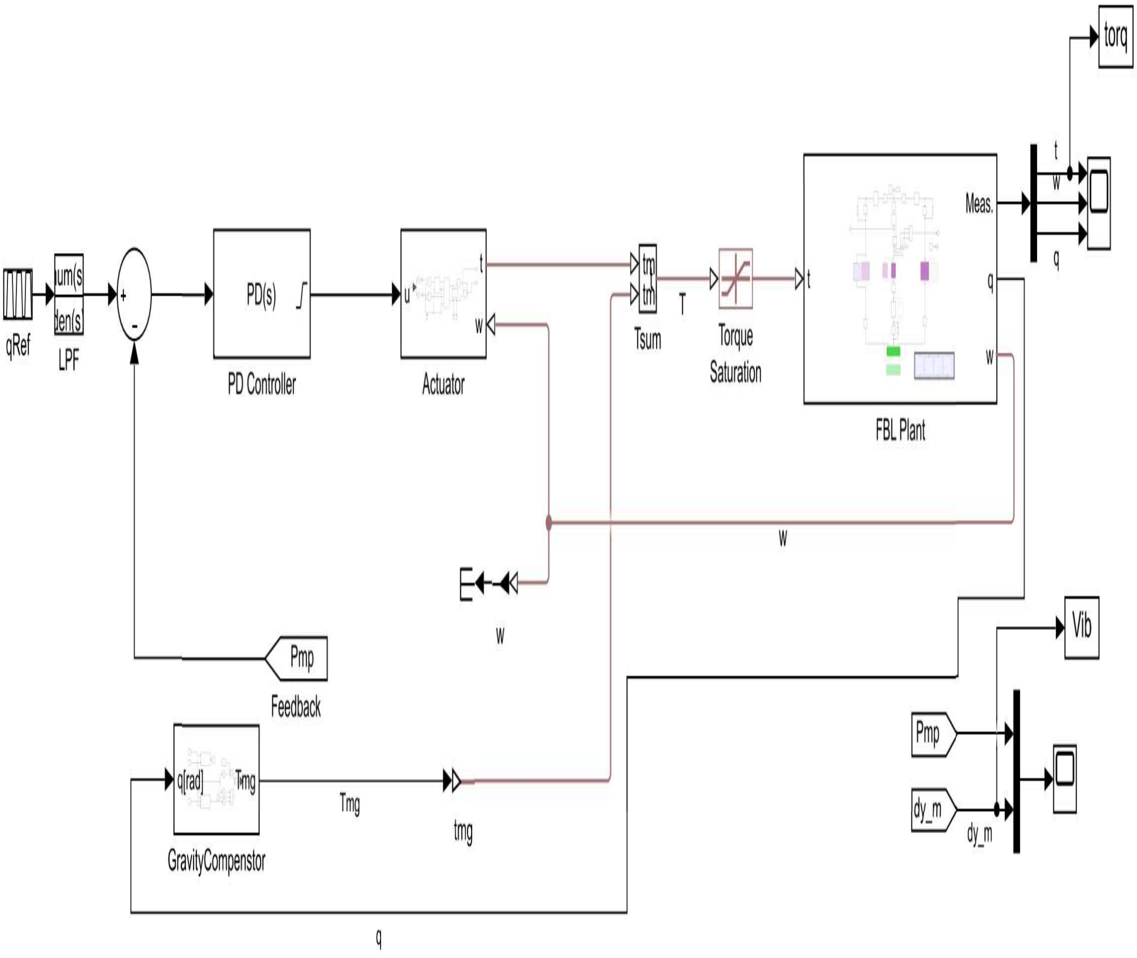


Figure 4.4. PD Control System

bottles. The controller includes a compensator for the gravity effect. The plant is linearized for control selecting linear analysis points of 'input perturbation' from the reference input and 'output measurement' from actuator position. If the error signal e_q is equal to $q_{ref} - q$, where q_{ref} is the reference input and q is the coupler position feedback, the control law is

$$T_p = K_p.e_q + K_d.e_q.\frac{N}{1 + \frac{N}{s}} \tag{4.10}$$

where K_p, K_d, N are proportional gain, derivative gain, and derivative filter coefficient.

The overall control action, adds a gravity compensation

$$T = T_p + g \tag{4.11}$$

where T is the actuator torque, and g is the compensation with regard to gravity action.

Figure (4.5) shows the convention used in calculating the gravity compensation torque. The reference frame chosen for the link angles measurement is the vertical frame as shown. T_{mg} here is defined as the torque due to gravity, M_a, M_b, M_c, M_1, M_2 are links loads, P_a, P_b, P_c, P_d are joints loads, a, b, r, r_1 are link lengths, $q, \theta_1, \theta_2, \theta_3, \theta_4$ are the angles between the reference frame and the links or the equivalent rigid link system. Thus

$$T_{mg} = r\Big[\big(\frac{M_1}{2} + y_8\big)\sin q + A\frac{\sin\theta_4}{\cos\theta_4}\cos q\Big], \tag{4.12}$$

where

$$y_8 = P_a + P_b + P_c + P_d + M_a + M_b + M_c + M_2, \text{ and,} \tag{4.13}$$

$$A = P_a + P_b + P_c + M_a + M_b + M_c + 3M_2/2 \tag{4.14}$$

Measuring angles at crank and follower joints, angles θ_2, θ_4 are calculated as

$$\theta_2 = asin\Big[\frac{(a\cos\theta_3 - a\cos\theta_1)}{b}\Big], \text{ and,} \tag{4.15}$$

$$\theta_4 = acos\Big[\frac{(a\cos\theta_3 - b/2\cos\theta_2 - r\cos q)}{r_1}\Big] \tag{4.16}$$

In inverse dynamics, a reference path is simulated and the torques are observed.

LQR scheme

The LQR control scheme is shown in 5.5 consisting of the optimal feedback gain K_{lqr} multiply the input action and in positive feedback to generate the final input to the plant. The the LQR optimal control implementation anticipates that all

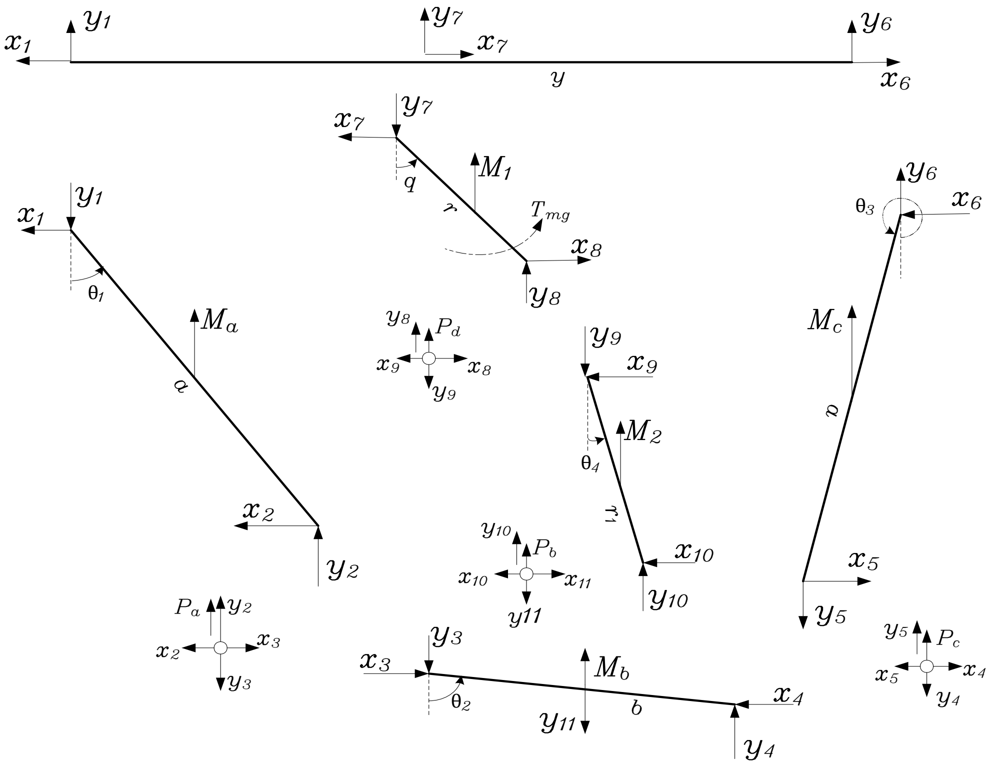


Figure 4.5. convention used to calculate T_{mg}

states are measured. However, practically, all the states of the system cannot be measured. Thus the is need for an observer to estimate the states of the system. The observer implemented is a Kalman observer as presented previously. The Kalman filter takes the input signal and the recorded output to and predicts the states.

Chapter 5

Numerical Example Simulations

The dimensions used for building the physical model and the material properties used are as shown in table 5.1.

Table 5.1. Specifications for model

link a	link b	link c	link r	link r_1	link 0 (base)
Link lengths (m)					
0.231	0.135	0.231	0.135	0.068	0.270
Link masses(kg)					
0.060	0.036	0.060	0.021	0.039	
Number of mass element					
20	16	20	16	10	16
Link width: 0.02m link thickness: 0.005m Material density: 2700 kg/m Elastic modulus: 70e9 N/m^2					

Table 5.2 shows the characteristics used as actuator

Table 5.2. Actuator properties

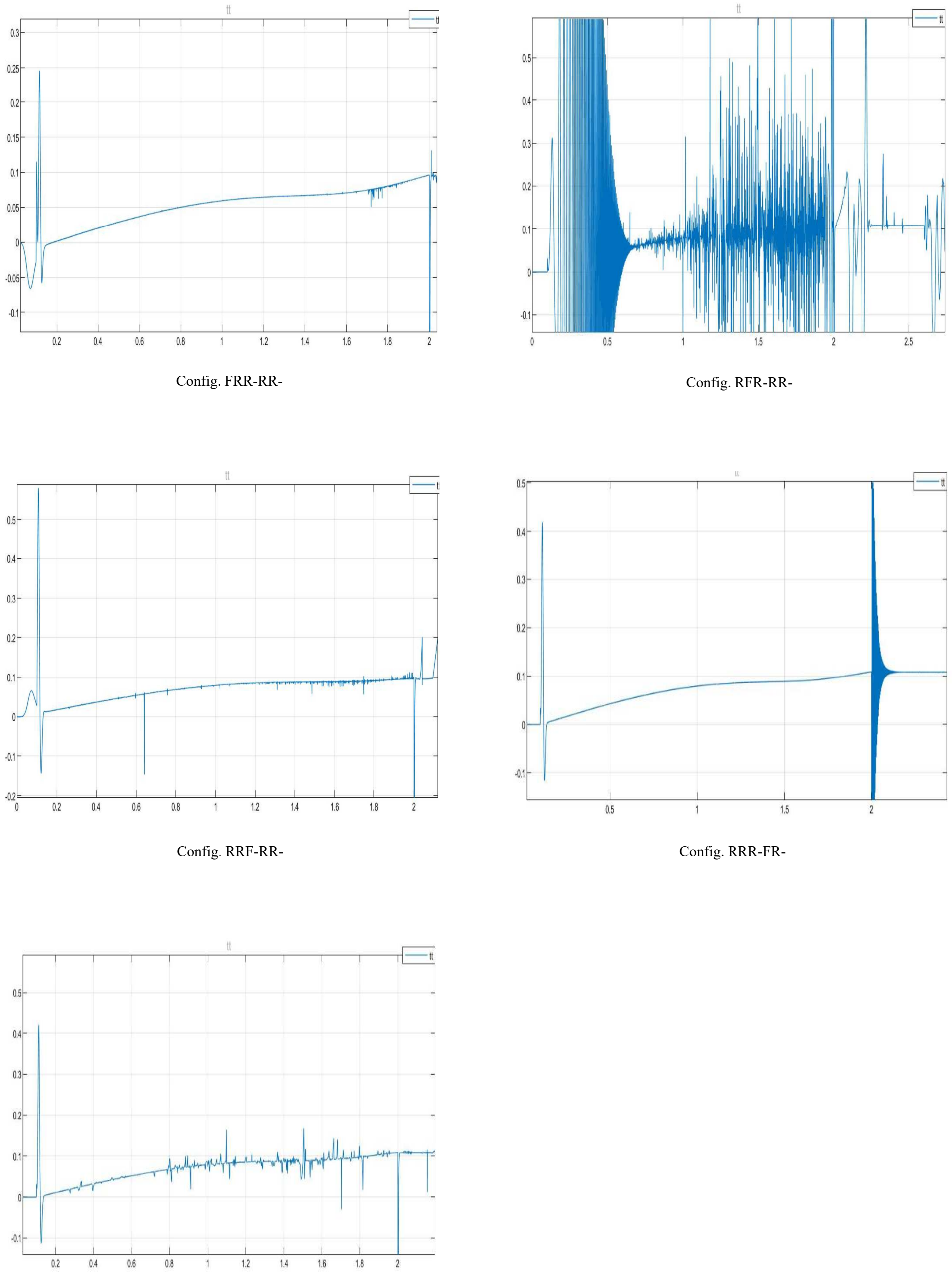
k_t: 18.4e^{-3}Nm/A; K_v: 520rpm/V; La:1.2e^{-5}; Ra=1Ω;
J: 1$e^{-6}gcm^{-2}$; time constant: 6.07e^{-}3s; Max. torque: 0.45Nm

The torque saturation reported on different actuator data sheet sources varied, but the value reported in table 5.2 was selected. After setting up the model, *ode23t* being the best for the type of mechanism modeled, is selected as the Simulink solver.

The control system for the modeled system was simulated using Matlab Simulink software. The physical modeling in the plant requires that for integration with simulink control design, there is inserted a signal interface to adapt simulink signals to physical model signals and vice-versa. Moreover, the mechanical systems and electrical systems do not also connect directly, hence an interface to connect the two domains, with regard to the actuator and the linkage structure, is used to make sure the system interacts seamlessly and properly. This is done using the Matlab Multiphysics Library package. For the case here, the *rotational simscape interface* is used as the actuator is a rotational mechanical device.

5.0.1 configuration analysis and selection

First, in the open loop structure of the physical model, reverse dynamics was used to observe the behaviour of the system with respect to different configuration of flexible link inclusion. Each link was made flexible while the others remained rigid and the torque and vibration was observed. Figure 5.1 shows the result for the configurations. (Naming syntax: link a, link b, link c, -link r, link r1-. eg. configuration RFR-RR-; where R is *rigid* and F is *flexible*). The FBL structure used in this study incorporates a spring in link $r1$. While this was implemented, it is observed that appropriate values for stiffness and damping coefficient was important to avoid increase in vibration during the motion of the system, and an exclusion of the spring connection, disrupted system from running because it simulates a near-singularity situation. The value of the spring stiffness and damping are set to $10e^{7}$N/m and $10e^{5}$N/m/s for good performance.. From the observation, configuration *F,F,R,-RR-*, also being among the two longest links, was selected and used for the control system implementation. The vibrations/torques in open loop or with same 'test' control system shows that the coupler vibrated relatively highly and same with the mid link crank. Of the two beams at the two sides the one opposite the direction of the motion of the FBL vibrated least. However,



Config. FRR-RR-

Config. RFR-RR-

Config. RRF-RR-

Config. RRR-FR-

Config. RRR-RF-

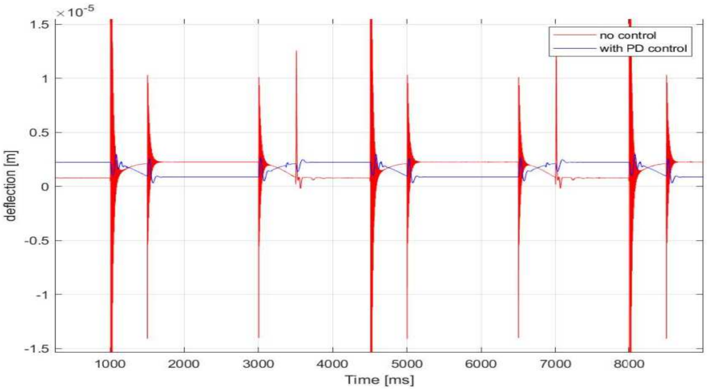
Figure 5.1. Torques for different arrangement of crank, coupler, follower and middle links[Y-axis in 'Nm', X-axis in 'seconds'

the two side beams made flexible, is selected because the trade-off between weight contribution and vibrations effect is best as they have the longest lengths and thus largest mass in the mechanism.

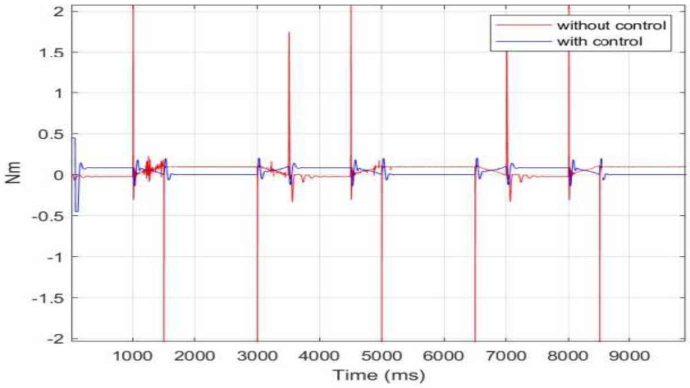
5.0.2 PD with gravity compensation

As a first step, a control system with a closed-loop feedback directly from the actuator joint was implemented, using PID and PD controls. The result showed that the system could not really affect the link vibration, but only to track reference motion just for the crank connected to the actuation point. This is one example where physical modeling helps to quickly understand the problem with designs and implementations. It is observed that the FBL design implied a somewhat disconnected system such that at the crank position $0 degrees$ the actuator has no control over the rest part of the structure. The effect of the return torque causes the system of the side beams and the coupler to go into oscillation while the crank is static. Thus a different design like the LQR is to be integrated to handle vibrations.

However, a changed PD control approach can affect the vibrations if the reference feedback point was altered from the direct actuator angle to the coupler link angle. This is with regard to the target behaviour of the FBL used in the study. For other FBL/multibody structure, this idea could be followed. For this therefore, the reference input is designed with respect to the movement of link b (the coupler) $\theta_2(q)$ instead of q. In this setup, while the motion for mechanism was followed, the link vibration was reduced also. Figure 5.2 compares the deflection measurement of the two control systems. Moreover, for the novel FBL mechanism that incorporates a spring in link r1, the design contributes to some vibration in downward motion of the mechanism to return the coupler to the horizontal plane. This is evident in the feedback of q. The change of feedback reference also attenuates this effect. The PD control was tuned to the parameters, $K_p = 20e^{-1}$, K_d $= 6.67e^{-2}$, and N = 100. Increasing the parameters for K_p and and tuning K_d accordingly, the system is more aggressive, torque increases and a continued increase leads to an erratic behaviour for example at $K_p = 100e^{-1}$ while K_d is unchanged from value above; or takingk_d down to $0.1e^{-1}$ while K_p has initial value above. In general, tuning the values downward leads to slowing down of the system and



Deflection in link a: with/without coupler PD control



Actuator torque: with/without coupler PD control

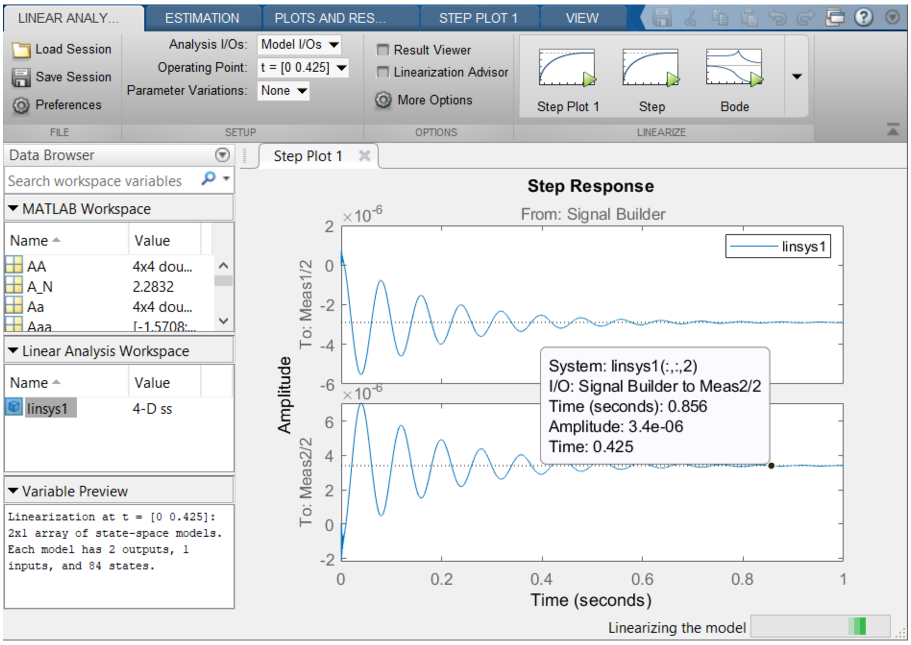
Figure 5.2. Torque and deflection using coupler PD control vs with no control

inability to follow the reference.

5.0.3 PD-LQR Control

Linear analysis and generation of SS equation

The LQR section requires a state space structure. In the modeling approach the FBL plant has to be linearized about a suitable operating point. The **trim** command/tool is used to obtain the operating point from analysis. However, in Simscape Multibody latest release as of the time of this work, this feature fails because the indexing of the states of the physical model uses a *dot* notation while at the time only *slash*"/" notation is supported. The operating point has to selected and set manually from analysing various points of the system and with respect to the input signal. Figure 5.3 shows the linear analysis and the step plot



Analysis I/Os: Model I/Os

Operating Point: t = [0 0.425]

Parameter Variations: None

Name	Value
AA	4x4 dou...
A_N	2.2832
Aa	4x4 dou...
Aaa	[-1.5708:...

Name	Value
linsys1	4-D ss

```
Linearization at t = [0 0.425]:
2x1 array of state-space models.
Each model has 2 outputs, 1
inputs, and 84 states.
```

System: linsys1(:,:,2)
I/O: Signal Builder to Meas2/2
Time (seconds): 0.856
Amplitude: 3.4e-06
Time: 0.425

Figure 5.3. Linear analysis

Model Reduction

A state space model is generated with 84 states. This number of states is high and leads to high computational demand. Moreover, it is possible to reduce the order of the model and retain the response of the system. The balanced reduction approach is used as it produces the best result. First, the hankel single value diagram is plotted to see the contribution of the states to the system, figure 5.4. Then the *balred* command is used to perform the system order reduction based on the hankel diagram in the command line format "$reducedmodel = balred(model, ORDER)$", where the '*order*' is the new order as informed by the hankel diagram. In this case, the system can be reduced to 4th order since the orders above 4 does not make significant contribution to the dynamics. A 4-states state space model is thus obtained which is used subsequently for the LQ design and the Kalman filter.

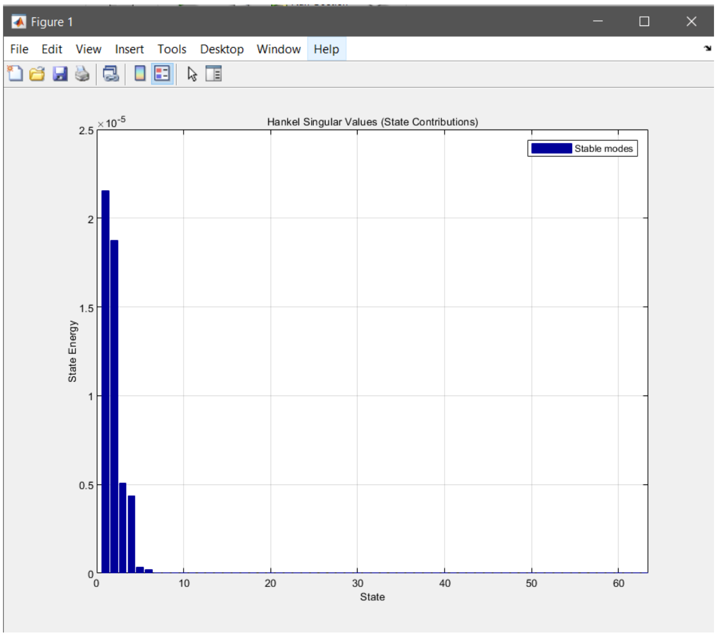


Figure 5.4. Hankel single values plot

PD-LQR Setup

The setup of the PD-LQR design is shown below. The PD loop is the outer loop and it is for the reference tracking while the LQR-loop is the inner loop and for the vibration suppression. The LQ loop takes the deflection ouptut and provides an optimal gain that attenuates vibration based on the value of the respective weighting factors. The vibration of the system is checked for a transition of the mechanism for q_{ref} from 0.0rad to 1.57rad and vice-versa. A two-step reference input is created by the *signal builder* block. The size of the matrix Q and matrix R 4, are respectively $[length(A)X length(A)]$ and $[length(B)X length(B)]$ - square matrices, where $length(A)$,$length(B)$ are lengths of the A and B state-space matrices, equal to the number of states and number of inputs - in this case 4 and 1 respectively. Q is an identity matrix with the principle diagonal as the weights. Increasing the values of the Q, a reduction in the vibrations is seen. Figure 5.6 shows the plots for the value of the diagonal elements of Q set to 100 and the value for R is set to 1. The torque saturation takes into account the limitation of the actuator chosen for the plant. Thus a continued increase of the matrix weights

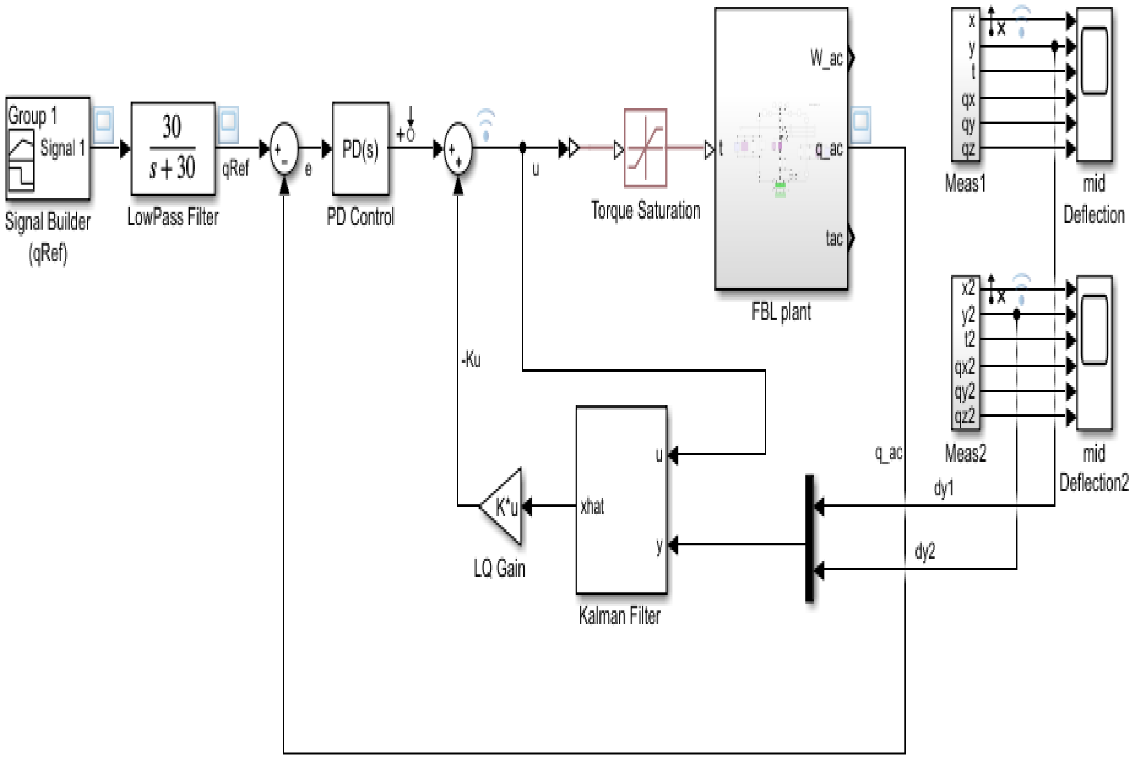


Figure 5.5. LQR controller scheme

does not necessarily imply a continued decrease of vibrations. The lowpass filter is used to avoid large torque spikes. As seen in the figure, the controller considerably suppresses vibration after about 0.06s. For the time response of plant with regard to the reference radian angle, figure 5.7 shows a good response with a rise time of 0.1s and a settling time of 0.2s. There is no overshoot.

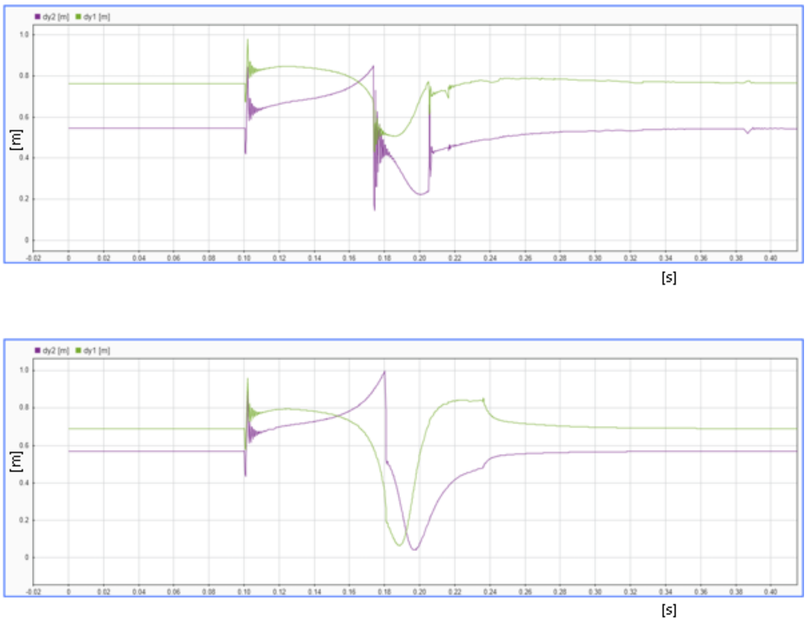


Figure 5.6. Vibration suppression in horizontal deflection

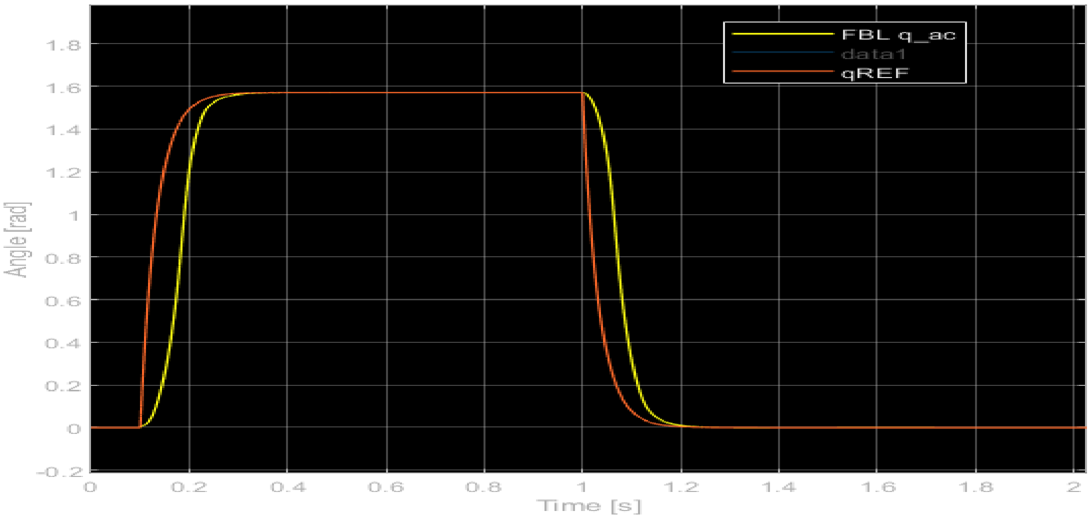


Figure 5.7. Time response plot

Chapter 6

Conclusion

6.0.1 Summary

Multibody systems in the most basic words consists of interconnected bodies. The system can be comprised of the only rigid bodies, only flexible bodies, or a mix of rigid and flexible bodies. Multibody systems applies to many engineering fields and largely seen in such areas as robotics, aeronoutics,etc. To handle these systems it is required to study how the respective bodies link to and affect each other and in turn affect the overall system composition. This is an analysis of the movement and force dynamics of the system. There is need for light and high-speed mechanisms and one that consumes the least amount of energy as possible. In this light, in the application of the multibody systems such as the bar-linkage mechanism , there is need for such mechanisms that adhere to the aforementioned demand; reduced inertia forces and driving torque requirement and fast operation.

This work presents the dynamic modelling and control of multibody systems with flexible links. The integration of flexible parts in multibody systems contributes to a lighter system and in view of high speed operation. The dynamics of the such systems is complex and a it's modelling is important for effective control. The approach of dynamic modelling of a multibody system have been presented using Mathworks SimScape Multibody and using the lumped parameter method. The methodology is presented with a four-bar linkage system with a different architecture - incorporates a centrally-placed crank which symmetrically divides the mechanism. With the presented method, the dynamic behaviour of the flexible-link system have been investigated considering the integration architecture of the flexible link(s). The deflection of the links in open loop configuration is analysed such that a chosen configuration is best with regard to ratio of the relative mass of the link(s) and the vibrations in the absence of control; thus the control effort demand is lowest.

Matlab/Simulink is used to implement a proposed control strategy of the bar-linkage multibody system modelled in SimScape, in this case with two flexible links for the two longest links in the mechanism. The method allows direct control of the physical model of the system and good analysis of the system behaviour.

Light weight links reduce the inertial force that the system actuator has to deal with. The use of light weights link materials come with a change in dynamics as the weights deform unlike the heavier rigid types. To model the dynamics of the system, SimScape physical modeling is used and with the finite segment method which models the flexible links with the mass element discretization of the links and springs and dampers with rotational degree of freedom. Higher number of discretized mass elements better model the flexibility dynamics and a modal analysis could be used to analyze and plot the mode shapes of the links.

In the numerical simulation in Matlab/Simulink, PD control with the feedback of the actuator position is investigated and showed that the feedback of the target link's position is better than directly of the actuator position. The numerical simulations in the modeling approach helps to understand this behaviour and thus choose a a good control scheme. In this light, the PD-LQR control method is presented and with better results. The PD loop handled the reference tracking while the LQR loop handled vibration control. For a step change in reference input, which represents the typical transition of the mechanism, the rise time is in 0.1 second and the settling time, 0.2 second.

The methodology presented can be applied to other kinds of flexible-link multi-body systems to analyse the dynamics and control the plant.

Appendix A

Matlab code

Below is the matlab code for four-bar linkage mechanism used in the study:

```

%% Configurations for Simscape/Simulink
close all, clear variables, clc
%% defaults
lbase = 270;
sp=20;
Bmax=300;
Bmax=Bmax + 20;
b = lbase/2;% mm rod length
d = b/2;
theta2i = pi/2; %rad
theta2f1 = pi; %rad
theta2f2 = 0;% rad
A=80; %A = A max ; mm larger size% B = B max; mm larger size% C = C max; mm larger size
C=120;
e = 95;
Cq = 100;% mm quadrilateral width
theta1i = atan((lbase-b)/(lbase+b)); %rad initial angle between the crank and the vertical when the rod b is horizontal (theta2i = pi / 2)
thetap = theta1i ; %rad value of the angle between rod b and r1 between 0 and theta1. To minimize the moment of the actuator, equality is required
a = lbase/(sin(theta1i)+cos(theta1i)); % mmcrank length
```

```matlab
theta1i = theta1i * (180/pi); %degrees
b1 = 2*(a*cos(theta1i)-(Bmax/2)); % mm
d1 = lbase-(A/2)-(a*sin(theta1i)); % mm
e1 = C/2 + sp; %
theta1f1 = asin((lbase-b)/(2*a));
theta1f11 = theta1f1*(180/pi);%
r1 = b*(a*cosd(theta1i)-(b/2));
r1=r1/((b*(sin(thetap)+cos(thetap))) - (2*a*cosd(theta1i))*(cos(thetap)-1));
r = (a*cosd(theta1i)) - r1;
thetar = atan(((r1*sin(thetap)) + b/2)/((a*cos(theta1i ))-(r1*cos(thetap ))-b/2));
thetarr1 = thetar*(180/pi);
thetap1 = thetap*(180/pi);
centr = a*sin(theta1f1) + d;
%% New configs.
% corrections for curved crank and follower r1, r r=b/2;
% r1=(a*cos(thetap))-r; % adj=(r+r1)*0.96375; r1=adj-r;
% adj=(r+r1)*0.982000; r1=adj-r;
r=b/2; r1=(a*cos(thetap))-r;
%%
y=lbase/10;
w = 2;
d = 0.8;
%% Crank Rigid ** (hole,peg) (O===[])
la=sqrt(((a/10)^2)/2);
A = linspace(pi/2.2, -pi/2.2, 10)';
B = linspace(pi/2, 3*pi/2, 10)';
C = linspace(5*pi/2, pi/2, 20)';
D = linspace(-pi/2, 0, 20)';
E = linspace(0, -pi/2, 20)';
csRight = [w/2*sin(A) la + w/2*cos(A)];
csLeft = [-la + w/2*cos(B) ((w/2*sin(B))+1)];
csLeftHole = [-la + d/2*cos(C) ((d/2*sin(C))+1)];
csConnLine1 = [-(la-w/2)+la*cos(D) la+la*sin(D)];
csConnLine2 = [-(la-w/2)+(la-w)*cos(E) (la-w)*sin(E)+la];
connX = [-(la/0.98-w/2),w; -la,w];
cscb2 = [csLeft; csConnLine1; csRight; csConnLine2; connX; csLeftHole];
```

```matlab
% figure; hold on; axis equal; grid on; plot(cscb2(:,1),
% cscb2(:,2),'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize',
% 9,'MarkerEdgeColor', [1 0 0]);
%% Rocker Rigid ** (hole,peg) (O===[])
A = linspace(pi/2.001, -pi/2.001, 10)';
B = linspace(pi/2, 3*pi/2, 10)';
C = linspace(4*pi/2, 0, 20)';
D = linspace(-pi/2, 0, 20)';
E = linspace(0, -pi/2, 20)';
csRight = [w/2*sin(A) la + w/2*cos(A)];
csLeft = [-la + w/2*cos(B) ((w/2*sin(B))+1)];
csRightHole = [d/2*cos(C) la + (d/2*sin(C))];
csConnLine1 = [-(la-w/2)+la*cos(D) la+la*sin(D)];
csConnLine2 = [-(la-w/2)+(la-w)*cos(E) (la-w)*sin(E)+la];
connX = [-(la/0.98-w/2),w; -la/0.1,w];
cscb3 = [csRightHole; csRight; csConnLine2; csLeft; csConnLine1];
% figure; hold on; axis equal; grid on; plot(cscb3(:,1),
% cscb3(:,2),'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]);
%% r Link, rigid
l5=r/10;
Aaa = linspace(-pi/2, pi/2, 10)';   % right edge (nb:column vector)
Bbb = linspace(pi/2, 3*pi/2, 10)';  % left side edge
Ccc = linspace(3*pi/2, -pi/2, 20)'; % full circle(touching the connection line)
csR22= [l5/2+w/2*cos(Aaa), w/2*sin(Aaa)];
csL22= [-l5/2+w/2*cos(Bbb), w/2*sin(Bbb)];
csLH22= [-l5/2 + d/2*cos(Ccc) d/2*sin(Ccc)];
csCL22= [-l5/2 -w/2; 0 -w/2];
csr2 = [csR22; csL22; csLH22; csCL22];
% figure; hold on; axis equal; grid on; plot(csr2(:,1),
% csr2(:,2),'Color',... [0.6 0.6 0.6], 'Marker', '.','MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]);
%% r Link, Flexible
l5=r/10;
nr=16;
l5f=l5/nr;
Aaa = linspace(-pi/2, pi/2, 10)';   % right edge (nb:column vector)
```

```matlab
Bbb = linspace(pi/2, 3*pi/2, 10)';  % left side edge
Ccc = linspace(3*pi/2, -pi/2, 20)'; % full circle(touching the connection line)
csR22= [w/2*cos(Aaa), w/2*sin(Aaa)];
csL22= [w/2*cos(Bbb), w/2*sin(Bbb)];
csLH22= [d/2*cos(Ccc), d/2*sin(Ccc)];
    csrf{1}= [csR22; -15f w/2; -15f -w/2];
    csrf{6}= [csL22; csLH22; 0 -w/2; 15f -w/2; 15f w/2];
    csrf{2} = [0 -w/2; 15f -w/2; 15f w/2; 0 w/2];
% figure; hold on; axis equal; grid on; plot(csrf{1}(:,1),
% csrf{1}(:,2),'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize',
% 9,'MarkerEdgeColor', [1 0 0]); figure; hold on; axis equal; grid on;
% plot(csrf{6}(:,1), csrf{6}(:,2), 'Color', [0.6 0.6 0.6], 'Marker','.',...
% 'MarkerSize', 9, 'MarkerEdgeColor', [1 0 0]); figure; hold on;axis equal;
% grid on; plot(csrf{2}(:,1), csrf{2}(:,2), 'Color', [0.6 0.6 0.6],
% 'Marker', '.',... 'MarkerSize', 9, 'MarkerEdgeColor', [1 0 0]);
%% r1 Link Rigid
r0=r1/10;
r3=0.5*(r0);
r4=(r0)-2*r3;
rn=r0-r3;
ar = linspace(-pi/2, pi/2, 10)';   % right edge (nb:column vector)
br = linspace(pi/2, 3*pi/2, 10)';  % left side edge
cr = linspace(3*pi/2, -pi/2, 20)'; % full circle(touching the connection line)
csRr= [w/2*cos(ar), w/2*sin(ar)];
csLr= [w/2*cos(br), w/2*sin(br)];
csLHr= [d/2*cos(cr) d/2*sin(cr)];
csCLnh= [0 w/2; -r3, w/2; -r3 -w/2; 0 -w/2];
csCLh= [0 -w/2; r3 -w/2; r3 w/2; 0.1 w/2];
csr1a = [csRr; csCLnh];
csr1b = [csLHr; csCLh; csLr];
% figure; hold on; axis equal; subplot(1,2,2); plot(csr1a(:,1), csr1a(:,2),
% 'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]); subplot(1,2,1); plot(csr1b(:,1), csr1b(:,2),
% 'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]);
%% r1 Link Flexible
r0=r1/10;
r3=0.5*(r0);
% r4=(r0)-2*r3; % delete line and replace r4 in simulink with 0 rn=r0-r3;
% % rn = r3 -- delete and update simulink
ar = linspace(-pi/2, pi/2, 10)';    % right edge (nb:column vector)
br = linspace(pi/2, 3*pi/2, 10)';   % left side edge
cr = linspace(3*pi/2, -pi/2, 20)';  % full circle(touching the connection line)
csRr= [w/2*cos(ar), w/2*sin(ar)];
csLr= [w/2*cos(br), w/2*sin(br)];
csLHr= [d/2*cos(cr) d/2*sin(cr)];
nr1=16*2;
re = r0/nr1;
csr1e1 = [csRr; -re w/2; -re -w/2];
csr1e2 = [0 w/2; 0 -w/2; re -w/2; re w/2];
csr1e4 = [csLr; csLHr; 0 -w/2; re -w/2; re w/2];
% figure; hold on; axis equal; subplot(1,2,2); plot(csr1e1(:,1),
% csr1e1(:,2), 'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]); subplot(1,2,1); plot(csr1e4(:,1),
% csr1e4(:,2), 'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]);
%% Coupler rigid (hole,hole,hole)
l=b/10;
l_t=l;
A = linspace(-pi/2, pi/2, 10)';    % right edge (nb:column vector)
B = linspace(pi/2, 3*pi/2, 10)';   % left side edge
C = linspace(3*pi/2, -pi/2, 20)'; % full circle(touching the connection line)
csLeft = [-l/2 + w/2*cos(B) w/2*sin(B)];
csRight = [l/2 + w/2*cos(A) w/2*sin(A)];
csLeftHole = [-l/2 + d/2*cos(C) d/2*sin(C)];
```

```
csMidHole = [d/2*cos(C) d/2*sin(C)];
csRightHole = [l/2 + d/2*cos(C) d/2*sin(C)];
csConnLine1 = [-l/2 -w/2; 0 -w/2];
csConnLine2 = [0 -w/2; l/2 -w/2];
csb = [csRight; csLeft; csLeftHole; csConnLine1;  csMidHole; ...
        csConnLine2; csRightHole];
% figure; hold on; axis equal; grid on; plot(cscb3(:,1), cscb3(:,2),
% 'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]);
%% Coupler Flexible (hole,hole,hole)
nco=16*2;
lc=l/nco;
A = linspace(-pi/2, pi/2, 10)';  % right edge (nb:column vector)
B = linspace(pi/2, 3*pi/2, 10)'; % left side edge
C = linspace(3*pi/2, pi/2, 20)'; % full circle(touching the
    connection line)
C2 = linspace(pi/2, -pi/2, 20)';
csLeft = [w/2*cos(B) w/2*sin(B)];
csRight = [w/2*cos(A) w/2*sin(A)];
csLeftHole = [d/2*cos(C) d/2*sin(C)];
csLeftHoleM = [lc+d/2*cos(C) d/2*sin(C)];
csLeftHole2 = [d/2*cos(C2) d/2*sin(C2)];
    csb1 = [csLeftHole; csLeft];
    csb2 = [csLeftHole2; 0,-w/2; lc,-w/2; lc,w/2; 0,w/2];
    csb3 = [0,-w/2; lc,-w/2; lc,w/2; 0,w/2];
    csb5 = [0,-w/2; lc,-w/2; csLeftHoleM; lc,w/2; 0,w/2];
    csb6 = csb2;
    csb9 = csb5;
    csb10 = [csLeftHole2; csRight];
% figure; hold on; axis equal; grid on; plot(csb12(:,1), csb12(:,2),
% 'Color',[0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]);
%% Crank Flexible
la2=la*1.05;
A = linspace(pi/2, -pi/2, 10)';  % right edge
B = linspace(pi/2, 3*pi/2, 10)'; % left edge
C = linspace(3*pi/2, pi/2, 10)'; % hole
C2 = linspace(pi/2, -pi/2, 10)'; % hole
csRight = [w/2*sin(A) w/2*cos(A)];
```

```matlab
csLeft = [w/2*cos(B) w/2*sin(B)];
csLeftHole = [d/2*cos(C) d/2*sin(C)];
csLeftHole2 = [d/2*cos(C2) -(la2-w/2)+d/2*sin(C2)];
ncr= 20*2;
deg(1) = -pi/2;
inc = deg(1)/ncr;
for i=2:ncr+1
    deg(i)=deg(i-1) - inc;
    D = linspace(deg(i-1), deg(i), 20)';
    csCL{i-1} = [la2*cos(D) la2*sin(D)];
    E = linspace(deg(i), deg(i-1), 10)';
    csCL2{i-1} = [(la2-w)*cos(E) (la2-w)*sin(E)];
end
l2f=la2/ncr;
%matrices
cscba1 = [csLeft; csLeftHole];
cscba2 = [csLeftHole2; csCL{1}; csCL2{1}];
for i=3:ncr+1
    cscba{i}=[csCL{i-1}; csCL2{i-1}];
end
cscba{42} = csRight;
% angles & positions for reference frames
for i=1:ncr
    ang(i)=(pi/2)*(i/10);
    yax(i)=(la2-w/2)*cos(ang(i));
    xax(i)=(la2-w/2)*sin(ang(i));
end
% figure; hold on; axis equal; grid on; plot(cscba1(:,1),
% cscba1(:,2),'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize',
% 9,'MarkerEdgeColor', [1 0 0]); % figure; hold on; axis equal; grid on;
% plot(cscba2(:,1), cscba2(:,2),'Color', [0.6 0.6 0.6], 'Marker', '.',...
% 'MarkerSize', 9,'MarkerEdgeColor', [1 0 0]); for i=3:ncr+1
% plot(cscba{i}(:,1), cscba{i}(:,2), 'Color', [0.6 0.6 0.6], 'Marker',
% '.',... 'MarkerSize', 9, 'MarkerEdgeColor', [1 0 0]); end
% plot(cscba{42}(:,1)+0.1*la, cscba{42}(:,2)+0.1*la, 'Color', [0.6 0.6
% 0.6], 'Marker', '.',... 'MarkerSize', 9, 'MarkerEdgeColor', [1 0 0]);
%% Rocker Flexible
% same as coupler except for two ends alternated
A = linspace(pi/2, -pi/2, 10)';  % right edge
B = linspace(pi/2, 3*pi/2, 10)'; % left edge
C = linspace(pi, 0, 20)';        % hole
C2 = linspace(0, -pi, 10)';      % hole
csRight = [w/2*sin(A) w/2*cos(A)];
csLeft = [w/2*cos(B) w/2*sin(B)];
csRightHole = [d/2*cos(C) d/2*sin(C)];
csRightHole2 = [(la2-w/2)+d/2*cos(C2) d/2*sin(C2)];
    cscba1r = csLeft;
    cscba2r = [csCL{1}; csCL2{1}];
    cscba21r = [csCL{20}; csRightHole2; csCL2{20}];
    cscba22r = [csRightHole; csRight];
% figure; plot(cscba21r(:,1), cscba21r(:,2), 'Color', [0.6 0.6 0.6],
% 'Marker', '.',... 'MarkerSize', 9, 'MarkerEdgeColor', [1 0 0]);
%% straight beam config.:
%%  Crank or Rocker: Rigid (straight beam)
l5s=a/10;
Aaas = linspace(-pi/2, pi/2, 10)';   % right edge (nb:column vector)
Bbbs = linspace(pi/2, 3*pi/2, 10)';  % left side edge
Cccs = linspace(3*pi/2, -pi/2, 20)'; % full circle(touching the connection line)
csR22s= [l5s/2+w/2*cos(Aaas), w/2*sin(Aaas)];
csL22s= [-l5s/2+w/2*cos(Bbbs), w/2*sin(Bbbs)];
csLH22s= [-l5s/2 + d/2*cos(Cccs) d/2*sin(Cccs)];
csCL22s= [-l5s/2 -w/2; 0 -w/2];
csr2s = [csR22s; csL22s; csLH22s; csCL22s];
% figure; hold on; axis equal; grid on; plot(csr2s(:,1), csr2s(:,2),
% 'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]);
%% Crank or Rocker: Flexible (straight beam)
l5s=a/10;
nsb=10*2;
lf=l5s/(2*nsb);
Aaas = linspace(-pi/2, pi/2, 10)';   % right edge (nb:column vector)
Bbbs = linspace(pi/2, 3*pi/2, 10)';  % left side edge
```

```
Cccs = linspace(3*pi/2, -pi/2, 20)'; % full circle(touching the connection line)
csR22s= [w/2*cos(Aaas), w/2*sin(Aaas)];
csL22s= [w/2*cos(Bbbs), w/2*sin(Bbbs)];
csLH22s= [d/2*cos(Cccs) d/2*sin(Cccs)];
    css{1}= [csR22; -lf w/2; -lf -w/2];
    css{6}= [csL22; csLH22; 0 -w/2; lf -w/2; lf w/2];
    css{2} = [0 -w/2; lf -w/2; lf w/2; 0 w/2];
% figure; hold on; axis equal; grid on; plot(css{2}(:,1), css{2}(:,2),
% 'Color', [0.6 0.6 0.6], 'Marker', '.',... 'MarkerSize', 9,
% 'MarkerEdgeColor', [1 0 0]);
%% Flexible Links Params.
beam_thickness=0.5; % all beams (in cm)
T=beam_thickness;
peg_thickness=0.5;
Pt=peg_thickness;
peg_thickness2=1;
wm=beam_thickness*1e-2;
dm = d*10^-2; % all beams hole diameter (in m)
material.aluminum.rho = 2770; % kg/m^3 ; Density
rho=material.aluminum.rho;
material.aluminum.E   = 70e9; % N/m^2  ; Young's Modulus %(modified)
material.aluminum.G   = 27e9; % N/m^2  ; Shear Modulus%
E = material.aluminum.E;
% material.steel.rho = 7800;
%material.steel.E   = 200e9; material.steel.G = 77.2e9;
rho2=rho;
%%  LPM Joints Stiffness & Damping
% Link/Length/Width/Thickness/Density/Youngs_Mod/SecondMomentofArea...
% Link/Length(per N) ------------------------------------
dc =2.5781e-05; % approx. damping proportionality constant
% assumpt.:damping be linear and bound by a constitutive type law (source:
% By S. Miller, T. Soares, Y. Van Weddingen, and J. Wendlandt, 2017)
% ------------------------------------
%% Crank/Rocker
lam = la*10^-2;   % hinge to hinge length (in m)
```

```
N = ncr;              % number of individual elements
L = 2*pi*lam/4;    % effective length
lx = L/N;            % length of each elements
rx2  =lam+wm/2; rx1 = lam-wm/2; % bar effective xsec
Ia = (pi/16)*(rx2^4-rx1^4);      % SecondMomentofArea
K.crank = (E*Ia)/lx;              % Stiffness [Nm/rad]
Z.crank = dc * K.crank;           % Damping  [Nm/(rad/s)]
%% Coupler
N = nco;
L = l*10^-2;
lx = L/N;
Ia = L*(wm^3)/12;
K.coupler = (E*Ia)/lx;
Z.coupler = dc * K.coupler;
%% r link
N = nr;
L = l5*10^-2;
Ia = L*(wm^3)/12;
lx = L/N;
K.rlink = (E*Ia)/lx;
Z.rlink = dc * K.rlink;
%% r1 link
N = nr1;
L = r0*10^-2;
Ia = L*(wm^3)/12;
lx = L/N;
K.r1link = (E*Ia)/lx;
Z.r1link = dc * K.r1link;
%% Crank/Rocker straight
L = l5s*10^-2;    % hinge to hinge length (in m)
N = nsb;              % number of individual elements
Ia = L*(wm^3)/12;
lx = L/N;
K.sb = (E*Ia)/lx;
Z.sb = dc * K.sb;
%% motor props
Tmax = 0.45; % max continuous torq = 15.4e-3 Nm
lpf = 6.07e-3;%0.01; % motortimeconstant=6.07e-3; % low pass filter coefficients
```

```
qstart = 0; % intial q output
Kt = 18.4e-3; % N*m/A
Kv = 0; % 520rpm/V
WpT = 104; % speed/torque gradient rpm/mNm
La = 1.2e-5; % 0.231e-3 H
Ra = 1; % 3.69 ohm
J = 1e-6; % (5.55gcm^2) rotor inertia
B = 0; % (Nm/(rad/s) rotor damping
Pwr= 24; % Watts
Spd = 18000; %(rpm) max perm.sble speed
Gear_ratio = 21/1;
%% Gravity comp.
%weights of beams , kg %(* gravitational acceleration) Follower=crank
g=9.81;
% Ma=0.0650708; Mc=Ma; % Crank.curved
%%
Ma= 0.061852*g; Mc=Ma; % Crank.straight
Mb=0.0370443*g; % coupler
Mr = 0.0214228*g; % r link
Mr1= 0.0386*g; % r1 link
% joints
M1=0.0015664*g; M3=M1; M4=M1;
Pa=M1;Pc=M1; Pd=M1;
% joint_@pump % longer dimension than other joints
M2=0.0109648*g;
Pb=M2;
y8=Pa+Pb+Pc+Pd+Ma+Mb+Mc+Mr1;
A_N=Pa+Pb+Pc+Ma+Mb+Mc+(3*Mr1/2);
rf=pi/2;
%% LQR
load('linearanalysis.mat')
H=linsys1(:,:,2);
hsvd(H)
Hb1=balred(H,4);
Hb2=balred(H,6);
figure, step(H), figure, step(Hb1),figure, step(Hb2)
MM=Hb1;
Aa = MM.A;
Bb = MM.B;
```

```
Cc  = MM.C;
Dd  = MM.D;
Q=(1e2)*eye(length(Aa));
R=  1;
N =  0;
[Klqr,Slqr,elqr]=lqr(Aa,Bb,Q,R,N);
mSYS=ss(Aa,Bb,Cc,Dd);
%% kalman
% M = [C D;zeros(1,12)  1];   % [y;u]  = M *  [x;u]
% QWV =  blkdiag(B*B',1e-2);
% QXU = M'*diag([1  1e-3])*M;
Qn=1;
Rn=0.01*eye(length(Dd));
Nn=0;
[kest,L,P]  = kalman(mSYS,Qn,Rn,Nn);
n=length(Klqr);
AA=Aa - Bb * Klqr;
for  i=1:n
      BB(i,:)=Bb(i) * Klqr(i);
end
CC=Cc;
DD=Dd;
% for  i=1:n
%          sys(:,i)=ss(AA,BB(:,i),CC,DD);
% end
% subplot(312)
% step(sys(:,1))
% subplot(313)
% step(sys(:,2))
```

Bibliography

[1] Connelly J.D., Huston R.L. The dynamics of flexible multibody systems: a finite segment approach - I.Theorical aspects. Computers and Structures, 50(2), 1994.

[2] Connelly J.D., Huston R.L. The dynamics of flexible multibody systems: a finite segment approach - II.Example Problems. Computers and Structures, 50(2), 1994.

[3] Fung, R.F., Chen K.W., Dynamic analysis and vibration control of a flexible slider-crank mechanism using PM synchronous servo motor drive, Journal of sound and Vibration, 214(4), (1998).

[4] Santosha K.D., Eberhard P., Dynamic analysis of flexible manipulators, a literature review. Mechanisms and Machine Theory, 41 (2006).

[5] Piras G., Cleghorn W. L., Mills J.K., Dynamic finite-element analysis of a planar high-speed, high-precision parallel manipulator with flexible links, Mechanism and Machine Theory, 40 (2005).

[6] Giovagnoni M., Rossi A., Transient Analysis of a flexible crank, Mechanism and machine theory, 24(4), (1989).

[7] Giovagnoni, M. (1994) A numerical and experimental analysis of a chain of flexible bodies. ASMEJ. Dyn. Sys., Meas., Control 116(1), 73-80 (1994).

[8] Lowen, G. G., and Jandrasits, W. G., Survey of investigations into the dynamic behavior of mechanisms containing links with distributed mass and elasticity, Mechanism and Machine Theory, 7, (1972).

[9] Shabana, A.A., Flexible multibody dynamics: review of past and recent developments, Multibody System Dynamics, 1 (1997).

[10] Cekus D., Posiadala B., Warys P., Integration of Modelling in Solidworks and Matlab/Simulink environments, Archive of mechanical engineering, 60(1),

(2014).

[11] Muscolo, G. G., Design and Development of the Four-Bar Linkage Module, proc. FRU-2 project, IIT Genova (2018)

[12] Muscolo, G. G., Cardito, F., Design and Development of the complete Test Bench, proc. FRU-2 project, IIT Genova (2018)

[13] Zhang, X. et al., Robust H_∞ vibration control for flexible linkage mechanism systems with piezoelectric sensors and actuators, Journal of Sound and Vibration 243(1), 145-155(2001)

[14] Kar, I. N. et al., Multimode vibration control of a flexible structure using H_∞-based robust control, IEEE/ASME transactions on mechatronics, 5(1), (2010)

[15] Sun, Q., Stable inversion control for flexible-link cooperating manipulators: A case study of a four-bar linkage mechanism, proc. IEEE/ASME international conference on advanced intelligent mechatronics (1999)

[16] Boscariol, P. et al., Evolution of a dynamic model for flexible multibody systems, Available at https://www.researchgate.net/publication/311353894

[17] Gasparetto, A., Accurate modelling of a flexible-link planar mechanism by means of a linearized model in the state-space form for desing of a vibration controller. Journal of Sound and Vibration, 240(2), 2001

[18] Caracciolo, R., Trevisani, A., Simultaneous rigid-body motion and vibration control of a flexible four-bar linkage, Journal of mechanism and machine theory, 36, 221243 (2001)

[19] Wang., J. et al., Adaptive simultaneous motion and vibration control for a multi-link mechanism with uncertain general harmonic disturbance, Journal of sound and vibration, 408., (2017)

[20] Trevisani, A., Feedback control of flexible four-bar linkages: a numerical and experimental investigation, Journal of Sound and Vibration, 268, 947970, (2003)

[21] Boscariol, P., et al. Vibration Reduction in a Flexible Link Mechanism through the Synthesis of an MPC Controller, Proceedings of the 2009 IEEE International Conference on Mechatronics. April, (2009).

[22] Trevisani, A., Valcher, M. E., Adaptive control of a flexible-link mechanism using an energy-based approach, Proceeding of the 2004 American Control Conference Boston, Massachusetts July, (2004)

[23] Caracciolo, R. et al., Deformation Control in Rest-to-Rest Motion of Mechanisms with Flexible Links, Journal of shock and vibration, (2018).

[24] Caracciolo, R. et al., Robust mixed-norm position and vibration control of flexible link mechanisms, Journal of mechatronics, 15, 767-791, (2005).

[25] Rong, B. et al., Dynamics and genetic fuzzy neural network vibration control design of a smart flexible four-bar linkage mechanism, Journal of Multibody system dynamics, (2012).

[26] Shawky, A., et al., Modeling and nonlinear control of a flexible-link manipulator, Journal of applied mathematical modelling, 37, 9591-9602(2013)

[27] Ulbrich, H., Von Stein, H., A Combined FeedforwardFeedback Control Strategy for improving the Dynamics of a Flexible Mechanism, Journal of multibody system dynamics, 7, 229-248, (2002).

[28] Hill, D. Dynamics and control of a rigid and flexible four bar coupler, Journal of vibration and control, 20(1) 131145, (2012).

[29] Miller, S., Soares T, Van Weddingen Y, Wendlandt J. Modeling Flexible Bodies with Simscape Multibody Software. Technical Paper, Mathworks, (2017).

[30] Miller, S. Flexible Body Models in Simscape Multibody, Mathworks, Technical paper, https://www.mathworks.com/matlabcentral/fileexchange/47051 , (2018)

[31] Physical Modeling, Mathworks, https://ch.mathworks.com/solutions/physical-modeling.html, (2018)

[32] Simscape Multibody Documentation, Mathworks, http://www.mathworks.com/help/physmod/sm/index.html, (2018)

[33] Chudnovsky,M., Kennedy, D., Mukherjee, A. Wendlandt, Modeling Flexible Bodies in SimMechanics and Simulink, Mathworks, White paper, https://ch.mathworks.com/company/newsletters/articles/modeling-flexible-bodies-in-simmechanics-and-simulink.html, (2006)

[34] Shabana, A. A., Dynamics of Multibody Systems. 4th Ed., Cambridge University Press (2013)

[35] Ogata, K., Modern Control Engineering, fifth edition. Prentice Hall, Boston, (2010).

[36] Academic dictionaries and encyclopedia., PID. Online: https://enacademic.com/dic.nsf/enwiki/41570 (2019).

[37] Hari O. B., Tuning of PID controllers, International journal of mathematical modeling, simulation and applications, 2(3), 337-344, (2009).

[38] Chang, K. E-design: computer-aided engineering design. Academic Press, Massachusetts, USA (2015).

[39] Margolis, D., The importance of Physical System Modelling to Industry: System Models that could have prevented some costly mistakes, International federation of Automatic Control, 48-21, (2015).

[40] Vavro, J. et al. Kinematic and dynamic analysis of planar mechanisms by means of SolidWorks software. International Polish-Slovak conference on Machine Modeling and simulations, (2016).