Modelling and control of a rotating flexible link

Modelling and control of a rotating flexible link

R.J.M. v.d. Ven

WFW 2001.10

Research project report

Supervisor: dr.ir. B.d.Jager
Coach(es): prof.dr.ing. O.Egeland

EINDHOVEN UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING
SYSTEMS AND CONTROL GROUP

Eindhoven, March 2001

Preface

This report is the result from a traineeship in the period from November 2000 until March 2001. The traineeship is held at the *Norges Teknisk-Naturvitenskapelige Universitet* (NTNU) in Trondheim. I would like to thank Professor O. Egeland who made this traineeship possible and who gave me usefull hints during the traineeship. The control of flexible structures is a challenging subject and nowadays still a lot of research is done in this field. In this report a single rotating flexible link is considered. I also would like to thank dr.ir. B. de Jager for his critical and accurate view on the report.

Eindhoven, March 2001

Richard van de Ven

Contents

Chapter 1

Introduction

In robot control the actuators are located in the joints while the controlled point will in general be at some other location, which is typically the tip point of the manipulator. A classical controller will use feedback from the actuator(s) and this type of control system is denoted as a collocated control system. In a robotic system elasticity will always be present. The elasticity can be concentrated in the links or/and in the joints. In this report the attention is focused on a rotating flexible link. For a manipulator with stiff elements the collocated type of control causes only small errors in the tip position of the manipulator. For a manipulator with flexible element(s) on the other hand, the collocated type of control can lead to large errors if high speed of motions is required. One may try to use a feedback loop from the tip of the manipulator but the system will then have nonminimum phase dynamics. For a linear model of the manipulator the nonminimum phase dynamics is due to right half plane zeros and these right half plane zeros limit the bandwidth of the control system.

In this project a single rotating flexible link is considered. The problem definition of the project is dual:

Is it possible to establish criteria for the appearance of right half plane zeros in the model of a flexible manipulator and to design a controller which circumvents the bandwidth limitation imposed by the right half plane zeros?

The goal of the project is then:

Develop criteria for the appearance of right half plane zeros and design a controller which circumvents the bandwidth limitation of the system

This report is organized as follows:

In chapter 2 the dynamic modelling of a single flexible link is considered. There are various ways to obtain a model of the system. In this report the attention is focused on the assumed modes approach and two dynamic models are obtained which only differ by the reference frame in which the position of the link is reported. The reference

frames are denoted as pseudo-pinned and pseudo-clamped. The attention in the rest of the report is mainly focused on the pseudo-pinned model. In chapter 3 the dynamic model is further analysed. A short analysis of the system is carried out in the time and frequency domain. Then the zerodynamics is analysed. The controller design is regarded in chapter 4. First the limitations of classical controller design are investigated. Then two relative new controllers are developed: a trajectory tracking controller and a state-to-state controller. The conclusions are given in chapter 5.

Chapter 2

Dynamic modelling of a single flexible link

In this chapter the dynamic modelling of a rotating flexible link is considered. The link is a distributed parameter system. The equations of motion are derived using an energy approach. The expressions for the kinetic- and potential energy and the work done by nonconservative forces are derived in the most general form. Then the equations of motion are reported in two different reference frames: the pseudo-clamped and the pseudo-pinned reference frame. Using the assumed-modes approach, a finite dimensional model is derived which can be used for control purposes.

2.1 Introduction to dynamic modelling of distributed parameter systems

Discrete dynamic models are built of discrete elements like discrete masses and massless springs and dashpots. The number of degrees-of-freedom (DOF) is generally equal to the number of discrete masses in the model. For a n-degree of freedom system the motion can be characterized by n simultaneously ordinary differential equations (ODE). The solution to this problem is then obtained by solving an algebraic eigenvalue problem and consists of n eigenvalues and eigenvectors.

For a continuous system on the otherhand the mass and stiffness are distributed throughout the system. The position of the system lies in a domain D. The exterior of the domain D is defined to be the boundary S. Because there are infinitely many points in this domain, a continuous system can be regarded as having infinitely many DOF's. The equations of motion are in this case governed by a boundary value problem which constists of (a set of) partial differential equations (PDE) and boundary conditions. The motion is dependent on time and position. The solution to the eigenvalue problem

consists of infinitely many eigenvalues and eigenfunctions. The eigenfunctions are orthogonal and can be decoupled. The boundary value problem can be found by means of Newtons second law or by means of Hamilton's extended principle.

2.1.1 Dynamic modelling of the single flexible link

A single rotating flexible link can be divided into three parts: a joint and an actuator at the base of the link, the link itself and, possibly, a payload at the tip of the link. The motor can be modelled as a rotating mass with constraints on velocity, acceleration and jerk. The joint introduces damping and friction. The simplest model of a flexible link in bending vibration is the Euler-Bernoulli beam.

In this report an Euler-Bernoulli beam with an actuator on one side and a payload at the tip of the link is considered. A model which includes both rotating inertia and shear deformation is called a Timoshenko beam. For control purposes it is necessary that the dynamic model contains all features which are essential to the dynamic behaviour of the flexible link. As reported in several articles the nonlinear terms, rotating inertia and shear deformation, are only of significant influence to the dynamic model if the beam length is considerably large compared to its bending stiffnes and if the speed of motion is very high. For control purposes these conditions rarely occur and therefore these terms are neglected. Often the interest lies in time intervals which are too short for damping effects to become measurable. So it is valid to ignore damping in the dynamic modelling. But it can be usefull to add a little damping to the link in order to obtain a strictly stable system.

For practical reasons the model of a distributed parameter system is often approximated by a finite dimensional model. The approximation can be done by several techniques. The two most frequently used techniques in literature are the assumed modes approach and the finite element approach. In this report the assumed modes approach is used.

2.2 Dynamic modelling

2.2.1 Assumption of the physical system

The system is built from a joint and actuator at the base of the link and a single flexible link. The link may carry a payload. The motor has a hub inertia J_H. Friction and damping in the joint are neglected but can be added later on. The link is considered to have length L, mass per unit length ρ and flexural rigidity EI. The payload has a mass M_P and inertia J_P. Several assumptions are made to the system:

1. The motion of the link is constrained to a horizontal plane: gravitational and torsional effects are ignored

2. Deflections of the link are small
3. Shear deformation and rotating inertia are ignored
4. The link is rigidly clamped to a hub
5. The mass is uniformly distributed and the link has uniform cross-section and constant material properties. Furthermore it is assumed that the bending of the link stays in the linear range.
6. Friction in the joint is neglected
7. Hub radius is ignored
8. No excentricity of the payload

2.2.2 Derivation of the equations of motion: general approach

Two common reference frames in which the equations of motion can be derived are the pseudo-clamped and the pseudo-pinned reference frame. In literature the classic clamped reference frame is used when the beam is rigidly clamped to an object with infinite inertia and thus there cannot be a nonzero slope at the base of the link. In the classic pinned reference frame on the other hand, there is no inertia at the base of the link. The reference frames used here are therefore denoted as "pseudo" because the link has an interia at the base of the link wich is neither infinite nor zero. In Figure 2.1 a schematic representation of the pseudo-clamped and pseudo-pinned reference frames is given. In the pseudo-clamped case the frame is tangent to the link at the base. The reference frame in the pseudo-pinned case is the one in which the x-axis intersects the instantaneous center of mass of the whole structure.

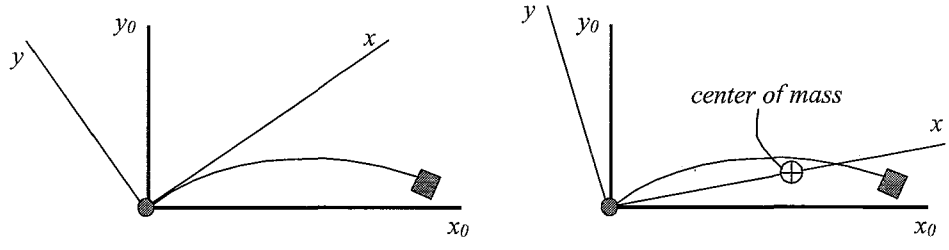


Figure 2.1: Pseudo-clamped reference frame (left) and pseudo-pinned reference frame (right)

In Figure 2.2 the flexible link with motor and a payload at the tip of the link is displayed. The deflection of the link is given in a general non-inertial reference frame (x_t, y_t). The inertial reference frame is denoted as (x_0, y_0).

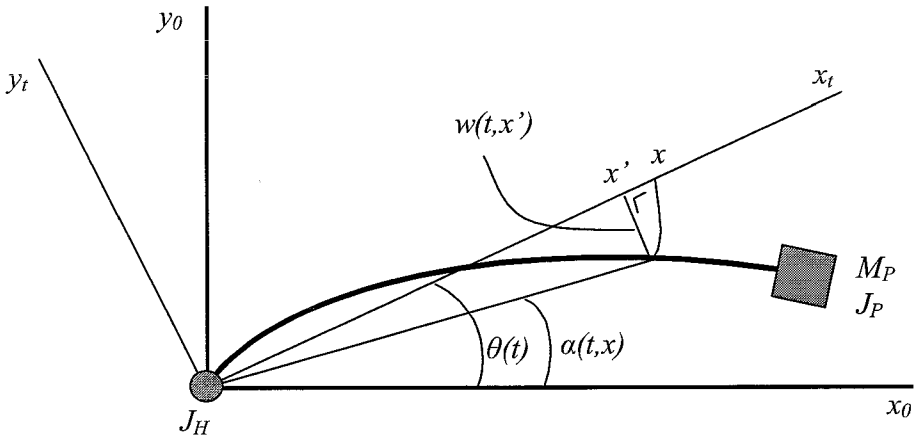


Figure 2.2: Top view of the flexible link with payload

The angle $\alpha(t,x)$ is approximated by:

$$\alpha(t,x) = \theta(t) + arc\tan\left(\frac{w(t,x')}{x'}\right) \approx \theta(t) + \frac{w(t,x)}{x} \tag{2.1}$$

A point on the flexible link is given by:

$$r(t,x) = \begin{bmatrix} x_{r_0} \\ y_{r_0} \end{bmatrix} = \begin{bmatrix} x\cos(\alpha(t,x)) \\ y\sin(\alpha(t,x)) \end{bmatrix} \tag{2.2}$$

The kinetic energy of the link is the sum of the kinetic energy of the hub, link and payload:

$$T = T_H + T_L + T_P \tag{2.3}$$

in which:

$$T_H = \frac{1}{2}J_H\dot{\alpha}^2(t,0) \tag{2.4}$$

$$T_L = \frac{1}{2}\int_0^L \dot{\mathbf{r}}^T(t,x)\dot{\mathbf{r}}(t,x)dm \tag{2.5}$$

$$T_P = \frac{1}{2}M_P\dot{\mathbf{r}}^T(t,L)\dot{\mathbf{r}}(t,L) + \frac{1}{2}J_P\dot{\alpha}^2(t,L) \tag{2.6}$$

The potential energy, which is only of elastic type of the beam, is given by:

$$V = \frac{1}{2}EI\int_0^L (\frac{\partial^2 w(t,x)}{\partial x^2})^2 dx \tag{2.7}$$

Finally the work done by the nonconservative forces (the applied torque $\tau(t)$):

$$W_{nc} = \tau(t)\alpha(t,0) \tag{2.8}$$

2.3 Pseudo-pinned reference frame

First the equations of motion are derived using the pseudo-pinned reference frame. The equations of motion for the flexible link are found by using Hamilton's extended principle:

$$\int_{t_1}^{t_2} (\delta(T-V) + \delta W_{nc})dt = 0 \tag{2.9}$$

This leads to the following equations of motion (see for example [7]):

$$EIw_p^{[4]}(t,x) + \rho(\ddot{w}_p(t,x) + x\ddot{\theta}_p(t)) = 0 \tag{2.10}$$

$$\tau(t) - J\ddot{\theta}_p(t) - \nu_p(t) = 0 \tag{2.11}$$

where a dot represents a time derivate. A derivate with respect to x will be denoted with a prime. If the number of derivates is larger then 2 then the number of derivates is denoted with $\overset{[number]}{w}$ or $w^{[number]}$ for derivates with respect to t and x respectively. The index p is used to refer to the pseudo-pinned case. The first equation describes the flexible behaviour of the link and the second equation describes the hub's motion.

$\nu_p(t)$ en J are given by the following equations:

$$\nu_p(t) = \rho\int_0^L x\ddot{w}_p(t,x)dx + J_H\ddot{w}_p'(t,0) + M_P L\ddot{w}_p(t,L) + J_P\ddot{w}_p'(t,L) \tag{2.12}$$

$$J = J_H + J_L + J_P + M_P L^2, \quad J_L = \rho\int_0^L x^2 dx \tag{2.13}$$

where $\nu_p(t)$ is the torque at the base due to the flexible behaviour of the link and J is the total inertia of the system seen at the motor.

The boundary conditions for the pseudo-pinned case are given by:

$$
\begin{aligned}
w_p(t,0) &= 0 && (2.14)\\
EIw_p''(t,0) &= J_H(\ddot{\theta}_p(t) + \ddot{w}_p'(t,0)) - \tau(t) && (2.15)\\
EIw_p''(t,L) &= -J_P(\ddot{\theta}_p(t) + \ddot{w}_p'(t,L)) && (2.16)\\
EIw_p^{[3]}(t,L) &= M_P(L\ddot{\theta}_p(t) + \ddot{w}_p(t,L)) && (2.17)
\end{aligned}
$$

The first boundary condition is a pure geometric one and the other ones are dynamic boundary conditions resulting from force balances. The second equation of motion (2.11) can be solved for $\theta_p(t)$ and be substituted in the first equation (2.10). The only independent variable is seen to be $w_p(t,x)$. A solution to these equations is obtained by seperation of variables, i.e., it is assumed that the solution can be seperated into a time dependend and position dependent part:

$$w_p(t,x) = q_p(t)\phi_p(x) \tag{2.18}$$

where $\phi_p(x)$ represent the eigenfunctions of the link and $q_p(t)$ represent the corresponding time-dependent generalised coordinates. The equations of motion are seperated accordingly:

$$
\begin{aligned}
EI\phi_p^{[4]}(x) + \rho\omega^2(\phi_p(x) + \frac{x}{J}\nu_0) &= 0 && (2.19)\\
\ddot{q}_p(t) + \omega^2 q_p(t) &= 0 && (2.20)
\end{aligned}
$$

where ω are the eigenfrequencies of the system and we made use of:

$$
\begin{aligned}
\nu(t) &= \ddot{q}_p(t)\nu_0 && (2.21)\\
\nu_0 &= \rho\int_0^L x\phi_p(x)dx + J_H\phi_p'(0) + M_P L\phi_p(L) + J_P\phi_p'(L) && (2.22)
\end{aligned}
$$

and the separated boundary conditions (τ is set to zero):

$$
\begin{aligned}
\phi_p(0) &= 0 && (2.23)\\
EI\phi_p''(0) &= -\omega^2 J_H\phi_p'(0) && (2.24)\\
EI\phi_p''(L) &= \omega^2 J_P\phi_p'(L) && (2.25)\\
EI\phi_p'''(L) &= -\omega^2 M_P\phi_p(L) && (2.26)
\end{aligned}
$$

The general integral to (2.19) and (2.20) is of the form:

$$
\begin{aligned}
\phi_p(x) &= A\sin(\beta x) + B\cos(\beta x) + C\sinh(\beta x) + D\cosh(\beta x) + Fx && (2.27)\\
&= \tilde{\phi}_p(x) + Fx\\
q_p(t) &= q(0)\cos(\omega t) + \dot{q}(0)\sin(\omega t) && (2.28)
\end{aligned}
$$

where $\tilde{\phi}_p(x)$ is the solution to the classic pinned case. In the pseudo-pinned case F turns out to be zero and the solution reduces to the classic pinned case. The values for β are obtained from the characteristic equation. This characteristic equation is found by imposing a nontrivial solution to (2.27). Using the boundary conditions, the coefficents B, C and D can be expressed in terms of A. From the first boundary condition it follows that:

$$D = -B$$

From the second, third and fourth boundary condition the following equations are obtained:

$$\begin{aligned} J_H\beta^3 A - 2\rho B + J_H\beta^3 C &= 0 \\ (-\rho C + Mp\beta S)A + (\rho(S - Sh) + M_P\beta(C - Ch))B + (\rho Ch + M_P\beta Sh)C &= 0 \\ (-\rho S - J_P\beta^3 C)A + (-\rho(C + Ch) + J_P\beta^3(S + Sh))B + (\rho Sh - J_P\beta^3 Ch)C &= 0 \end{aligned}$$

where $C = \cos(\beta L)$, $Ch = \cosh(\beta L)$, $S = \sin(\beta L)$ and $Sh = \sinh(\beta L)$. The characteristic equation is found by setting the determinant of the coefficent matrix to zero, where the coefficents d_{ij} are taken from the equations above:

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.29}$$

The characteristic equation is then found to be:

$$\begin{aligned} CSh - SCh - \frac{2M_P}{\rho} SSh\beta - (\frac{2J_P}{\rho} CCh + \frac{J_H}{\rho}(1 + CCh))\beta^3 - \\ \frac{M_P}{\rho^2}(J_H + J_P)(CSh - SCh)\beta^4 + \frac{J_H J_P}{\rho^2}(CSh + SCh)\beta^6 - \\ \frac{J_H J_P M_P}{\rho^3}(1 - CCh)\beta^7 = 0 \end{aligned}$$

The values of β are found by solving the characteristic equation. The solutions to the characteristic equation can be found numerically. Note that there are infinitely many solutions to the characteristic equation because the terms involving $\sin, \cos, \sinh$ and $\cosh$ are also dependent on β. This corresponds to an infinite dimensional model of the flexible link. The relation between β and the eigenfrequencies of the system, ω, is given by:

$$\omega_i = \beta_i^2 \sqrt{\frac{EI}{\rho}}, \quad i = 1, 2, \ldots \tag{2.30}$$

The coefficents A, B, C and D of (2.27) are not independent and can be expressed into a single coefficent by appropriate substitutions. This leads to the following solution for the eigenfunctions:

$$\phi_p(x) = A(\sin(\beta x) + \gamma(\cos(\beta x) - \cosh(\beta x)) + \zeta \sinh(\beta x)) \tag{2.31}$$

where γ and ζ are constants. All eigenfunctions can be shown to be orthogonal to each other. The constants A are determined by an orthonormalization procedure.

2.3.1 Finite dimensional model

As already mentioned in the previous section, the dynamic behaviour is described by an infinite dimensional model. From a practical point of view this is not desirable and the model is truncated. A finite dimensional model of the flexible link is obtained by truncating the deflection as a finite number of modes. The number nf determines the number of flexible modes still included in the model and thus the accuracy of the model.

$$w_p(t,x) = \sum_{i=1}^{nf} q_{p_i}(t)\phi_{p_i}(x) \tag{2.32}$$

The coefficent A of the solution to the eigenfunctions $\phi_p(x)$ is chosen such that it satisfies an orhonormality condition. In the pseudo-pinned case the orthornormality condition is given by (see [7]):

$$\begin{aligned} J_H\phi'_{p_i}(L)\phi'_{p_j}(L) + \rho\int_0^L \phi_{p_i}(x)\phi_{pj}(x)dx + M_P\phi_{p_i}(L)\phi_{p_j}(L) + J_P\phi'_{p_i}(L)\phi'_{p_j}(L) &= \triangle_{ij}, \quad (2.33)\\ i,j &= 0,1,\ldots,nf \end{aligned}$$

where $\triangle_{ij}$ represents the Kronecker delta. This condition can also be expressed as:

$$EI\int_0^L \phi''_{p_i}(x)\phi''_{p_j}(x)dx = \triangle_{ij}\,\omega_i^2, \quad i,j = 0,1,\ldots,nf \tag{2.34}$$

In Figure 2.3 the first three normalised mode shapes of a pseudo-pinned link are displayed. For large hub inertia the eigenfunctions approach the ones of a classic clamped end because the slope at the base of the link approaches zero.

Then the first eigenfunction and generalised coordinate is set equal to the rigid-body mode: $\phi_{p_0}(x) = x$ and $q_{p_0}(t) = \theta(t)$. A linear model is derived by using the following expressions for the kinetic energy, potential energy and the work done by nonconservative

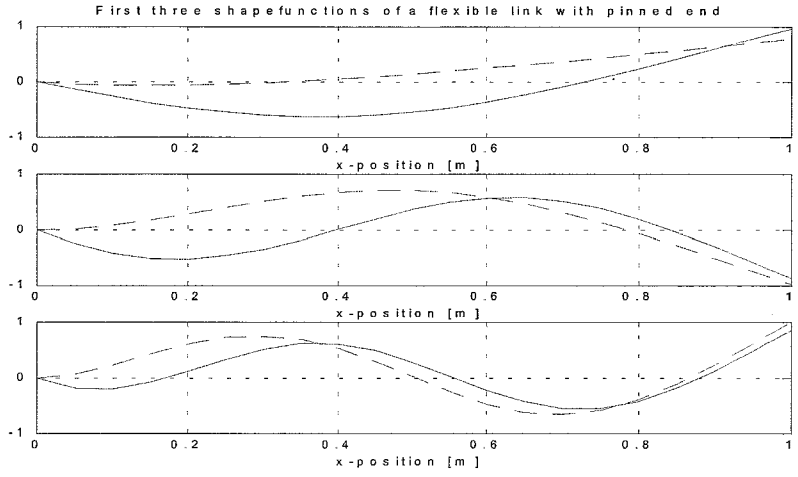


Figure 2.3: Mode shapes of a pinned flexible beam, $J_H = 0.01\ kg/m^2$ $(-)$ and $J_H = 1\ kg/m^2$ $(--)$

forces:

$$T = \frac{1}{2}\sum_{i=0}^{nf}\sum_{j=0}^{nf}\dot{q}_{p_i}(t)\dot{q}_{p_j}(t)m_{p_{ij}} \tag{2.35}$$

$$V = \frac{1}{2}\sum_{i=0}^{nf}\sum_{j=0}^{nf}q_{p_i}(t)q_{p_j}(t)\omega_{ij}^2\triangle_{ij} \tag{2.36}$$

$$W = \tau(t)\sum_{i=0}^{nf}\phi'_{p_i}(0)q_{p_i}(t) \tag{2.37}$$

where $m_{p_{ij}}$ are determined from the orthonormality condition (2.33):

$$\begin{aligned} m_{p_{00}} &= J \\ m_{p_{ij}} &= 1 \quad i = j \\ m_{p_{ij}} &= 0 \quad i \neq j \end{aligned}$$

This results in the following linear model:

$$\mathbf{M}_p\ddot{\mathbf{q}}_p + \mathbf{K}_p\mathbf{q}_p = \mathbf{F}_p\tau \tag{2.38}$$

where

$$\mathbf{M}_p = \begin{bmatrix} J & \mathbf{0}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix},\ \mathbf{K}_p = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \omega^2 \end{bmatrix},\ \mathbf{F}_p = \begin{bmatrix} 1 \\ \phi'(\mathbf{0}) \end{bmatrix} \tag{2.39}$$

and

$$\omega = diag\{\lceil\ \omega_1 \quad \cdots \quad \omega_{nf}\ \rceil\}$$

The system dynamics can also be written in state space form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.40}$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_p & \dot{\mathbf{q}}_p \end{bmatrix}^T \tag{2.41}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}_p^{-1}\mathbf{K}_p & \mathbf{0} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}_p^{-1}\mathbf{F}_p \end{bmatrix} \tag{2.42}$$

$$\mathbf{u} = \tau \tag{2.43}$$

Often also a little damping is added to the rigid body mode and the flexible modes. The damping coefficents are stored in a diagonal matrix $\mathbf{D}_p$. Typically the damping ratio for the flexible modes is $\zeta_e << 1$ and often also $\zeta_e << 0.1$. The first element in the matrix $\mathbf{D}_p$ represents the damping introduced by the motor and joint. The state space formulation of the system then becomes:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}_p^{-1}\mathbf{K}_p & -\mathbf{M}_p^{-1}\mathbf{D}_p \end{bmatrix} \tag{2.44}$$

$$\mathbf{D}_p = \begin{bmatrix} b_H & \mathbf{0}^T \\ \mathbf{0} & 2\zeta_e\omega \end{bmatrix} \tag{2.45}$$

In a real system there will always be some damping present. By adding a little damping to the model this will force the poles of the system to lie stricly in the left half plane (LHP) of the complex plane.

2.4 Pseudo-clamped reference frame

In the pseudo-clamped case the reference frame is chosen to be tangent to the link at the base. The relationship between the pseudo-clamped and the pseudo-pinned eigenfunctions follows from the pure geometric relation:

$$\phi_{c_i}(x) = \phi_{p_i}(x) - x\phi'_{p_i}(0) \tag{2.46}$$

where the index c is used to refer to the pseudo-clamped case. Because the characteristic equation is the same as in the pseudo-pinned case the values for β are the same. The solution to the eigenfunctions and generalised coordinates is of the same form. In the

pseudo-clamped case however F is nonzero. The orthonormalization conditions are given by:

$$\rho \int_0^L \phi_{c_i}(x)\phi_{c_j}(x)dx + M_P\phi_{c_i}(L)\phi_{c_j}(L) + J_P\phi'_{c_i}(L)\phi'_{c_j}(L) - \frac{\mu_i\mu_j}{J} = \triangle_{ij} \qquad (2.47)$$

$$EI \int_0^L \phi''_{c_i}(x)\phi''_{c_j}(x)dx = \triangle_{ij}\ \omega_i^2$$

where $i, j = 0, 1, \ldots, nf$ and:

$$\mu_i = \rho \int_0^L x\phi_{c_i}(x)dx + M_P L\phi_{c_i}(L) + J_P\phi'_{c_i}(L), \quad i = 1, 2, \ldots, nf \qquad (2.48)$$

The same expressions for the kinetic- and potential energy, (2.35) and (2.36), and the work done by nonconservative forces (2.37) are used as in the pseudo-pinned case. However, because the first orthornormality condition differs from the one of a pseudo-pinned beam, (2.33), the coefficents $m_{c_{ij}}$ are different:

$$\mathbf{m}_c = \begin{bmatrix} m_{c_{11}} & \frac{\mu_1\mu_2}{J} & \cdots & \frac{\mu_1\mu_{nf}}{J} \\ \frac{\mu_2\mu_1}{J} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{\mu_{nf-1}\mu_{nf}}{J} \\ \frac{\mu_{nf}\mu_1}{J} & \cdots & \frac{\mu_{nf}\mu_{nf-1}}{J} & m_{c_{nfnf}} \end{bmatrix} \qquad (2.49)$$

where

$$m_{c_{ii}} = 1 + \frac{(\mu_i)^2}{J} \quad i = 1, 2, \ldots, nf \qquad (2.50)$$

Also the terms $\phi'_{c_i}(0)$ are zero for $i = 1, 2, \ldots, nf$ because for a pseudo-clamped beam the deflection is reported in a frame tangent to the link at the base. This leads to the following linear model in the pseudo-clamped case:

$$\mathbf{M}_c\ddot{\mathbf{q}}_c + \mathbf{K}_c\mathbf{q}_c = \mathbf{F}_c\tau \qquad (2.51)$$

where

$$\mathbf{M}_c = \begin{bmatrix} J & \mu^T \\ \mu & \mathbf{m}_c \end{bmatrix},\ \mathbf{K}_c = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \omega^2 \end{bmatrix},\ \mathbf{F}_c = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \qquad (2.52)$$

and μ and ω are given by:

$$\mu = \begin{bmatrix} \mu_1 & \cdots & \mu_{nf} \end{bmatrix}^T \qquad (2.53)$$

$$\omega = diag\{\begin{bmatrix} \omega_1 & \cdots & \omega_{nf} \end{bmatrix}\} \qquad (2.54)$$

2.5 Comparison of the models

The two models are very much alike. They both have the same eigenfrequencies because the characteristic equation for both systems is the same. An interesting difference between both models is the way the rigid body mode and the flexible ones are coupled. In the pseudo-pinned model the coupling between the flexible modes and the rigid body mode is only through the input since all elements in $\mathbf{F}_p$ are different from zero and the matrices $\mathbf{M}_p$ and $\mathbf{K}_p$ are diagonal. In the pseudo-clamped model on the contrary the coupling between the flexible modes and the rigid body mode is via the mass matrix $\mathbf{M}_c$. Another difference between the models is the reference frame in which the position of the link is reported.

Chapter 3

Model analysis

In the rest of this report the pseudo-pinned model will be used. The deflection of a point on the link is thus reported in a frame which intersects the instanteneous center of mass of the whole structure. Simulations show that the dynamic behaviour of the two models is almost the same and the choice for the pseudo-pinned model is therefore rather arbitrary. In this chapter the model is further analysed. First the system is analysed in the time domain. Then the system is analysed in the frequency domain using Bode diagrams. Furthermore the zerodynamics is treated and its consequences for the control of this flexible link.

3.1 Parameter values and model output

In this and the following chapters it is assumed that the link is made of steel with a length $L = 1000\ mm$ and that the link has a rectangular cross section of $h = 50\ mm$ and $b = 3\ mm$. This results in the following physical quantities:

- $\rho = 4\ kg/m$
- $I = \frac{1}{12}hb^3 = 1.13 \cdot 10^{-10}\ m^4$
- $E = 2.1 \cdot 10^{11}\ N/m^2$

The other physical quantities of interest are assumed as:

- $J_H = 0.01\ kg/m^2$
- $J_P = 0\ kg/m^2$
- $M_P = 0\ kg$

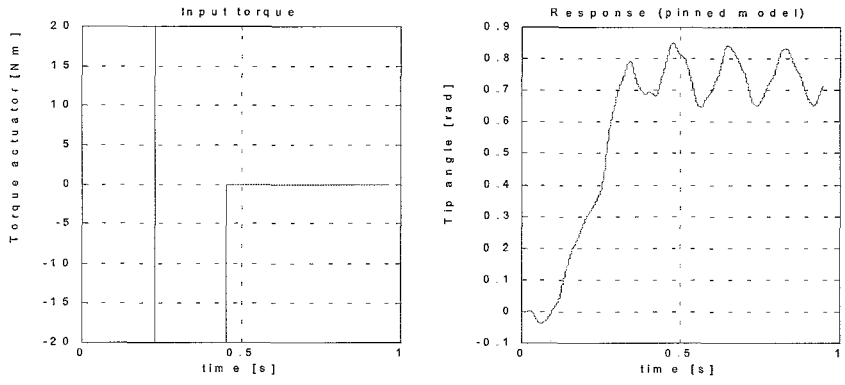


Figure 3.1: Input torque and response of the pinned model

- $\zeta_e = 0.01$
- $b_H = 0.05\ Ns$

As output of the model the angle between a point on the flexible link and the inertial frame is chosen. The output is then given by:

$$y = Cx \tag{3.1}$$

$$C = \begin{bmatrix} 1 & \frac{\phi_1(\bar{x})}{\bar{x}} & \cdots & \frac{\phi_{nf}(\bar{x})}{\bar{x}} & 0 & \cdots & 0 \end{bmatrix} \tag{3.2}$$

where the first component of C represents the rigid body mode and the second to the $(nf+1)^{th}$ element represent the contribution of the mode shapes to the output. When the tip angle is chosen as output L is substituted for $\bar{x}$.

3.2 A first identification of the system in the time domain

To gain some insight into the characteristics of the model some responses of the model are analysed in the time domain. An input torque is applied to the model and the response of the tip angle is viewed. First a rigorous non-smooth input torque is applied. In practice an actuator will not be able to generate such an input torque but the response shows some very important characteristics of the model. In Figure 3.1 the tip angle is displayed for a pseudo-pinned model with three flexible modes.

First it is seen that an inverse response occurs. The angle crosses the x-axis 3 times, see Figure 3.2, which indicates that the system has three right half plane (RHP) zeros (see [5]). This is equal to the number of flexible modes included in the model. The inverse response has an interesting interpretation. A torque applied at the base of the link

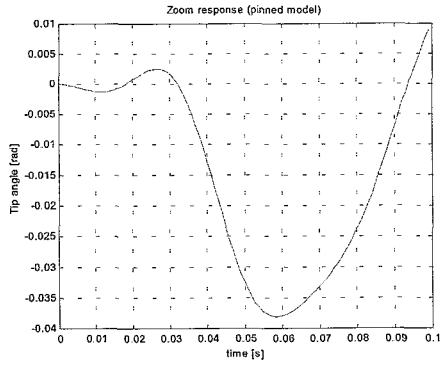


Figure 3.2: Zoom of the response to square torque

cannot immediately affect the tip position. The base of the link moves forward and the endpoint of the link in a reaction "kicks back". Furhtermore when no longer a torque is applied the tip angle shows resudial vibrations that persist long after the torque is applied because the damping of the flexible modes is very small. As can be seen from Figure 3.1, the higher frequencies are superponed on the lower ones. The response of the flexible modes to this torque is visualised in Figure 3.3.

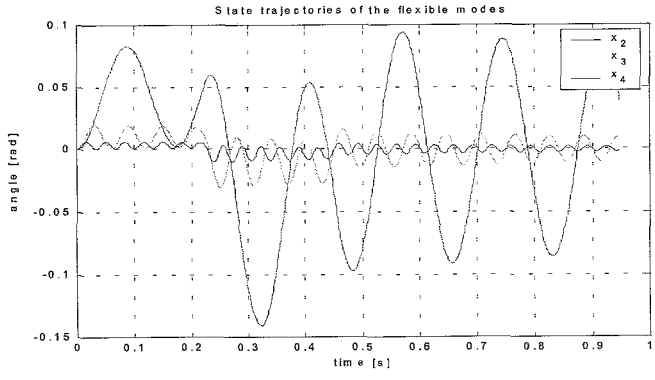


Figure 3.3: State trajectories of the flexible modes

The first flexible mode has the lowest frequency and the largest amplitude. A first estimation of the frequencies of the flexible modes can already be made from this figure. Further simulations also show that when the input torque is chosen to be a smooth function, as in practice, the amplitude of the trajectories from the flexible modes becomes very small, especially of the higher modes.

3.3 Analysis of the system in the frequency domain

The time domain identification already showed some main characteristics of the model. The model identification is further carried out in the frequency domain. The eigenfrequencies of the model can be clearly distinguished from the Bode diagram (see Figure 3.4) at the locations of the resonance peaks. Here a modal model with five flexible modes is taken. The phase lag at the frequencies (around $\omega = 10^{-2}\ rad/s$) is due to the little damping which is added to the rigid body mode.

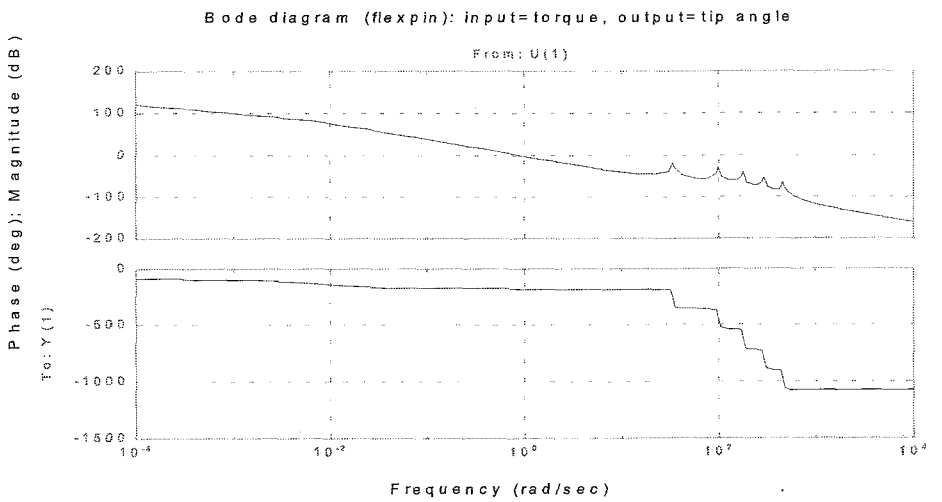


Figure 3.4: Bode diagram of the pinned flexible link with 5 modes included in the model

The eigenfrequencies can also be calculated by (2.30). This formula relates the number of "waves" occuring in the link for a modeshape and the frequency of vibration. Looking at the phase plot, with each flexible mode the phase drops with 180 degrees. When damping is small each flexible mode introduces, besides two poles, also two zeros, one in the LHP and one in the RHP. For a LHP zero the slope in the Magnitude plot of the Bode diagram will increase by one at the frequency of the zero and the phase will increase with 90 degrees. A RHP zero will also increase the slope in the Magnitude plot by one but the phase will drop with 90 degrees. A pair of zeros will therefore increase the slope of the Magnitude plot by two and the phase will be left unchanged when the absolute value of the zeros is equal, which is valid for the lightly damped flexible link. Together with the poles, the slope of the asymptote in the Magnitude plot is -2 for high frequencies and the phase decreases with 180 degrees at each eigenfrequency.

The pole-zero pattern for a system with five flexible modes, when the tip is chosen as output, is shown in Figure 3.5. The circles represent the zeros and the crosses represent the poles.

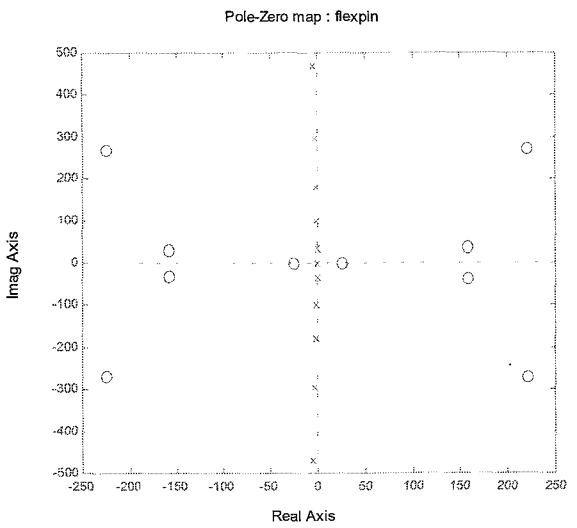


Figure 3.5: Pole-zero locations for a pinned model with five flexible modes included

Two poles lie close to the origin, representing the rigid body mode and all other poles are complex conjungate and lie close to the imaginary axis in the LHP. This indicates the lightly damped eigenmodes. The zeros appear in "pairs". With each eigenmode a pair of zeros is introduced. If no damping was included in the model the poles would lie on the imaginary axis and the pole-zero pattern would be point symmetric about the origin.

3.4 Model truncation

In order to be able to handle the dynamic model the number of flexible modes included in the model is often truncated. In doing that it is important to know how many flexible modes one still has to include for the model to stay valid in the frequency range of interest. For truncating the model two criteria are posed. If one of these criteria is met, it is justified to truncate the model up to that mode:

1. In the (controlled) system there occur no frequencies higher than the frequency of the highest included flexible mode in the model
2. The displacements introduced by the flexible modes higher than the highest included mode are smaller than the required accuracy.

For the fist criterium it is necessary to investigate the frequecy content of the (controlled) system. To this extend the power spectal density for an open-loop controlled system is

plotted (Figure 3.6).

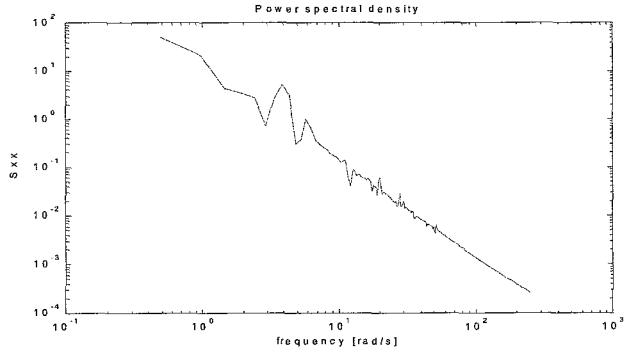


Figure 3.6: Power spectal density of an open loop controlled system

This power spectral density plot is obtained by using a square harmonic input. From this figure it is seen that the power spectral density for high frequencies is very low, even if the control input is not smooth. Regarding the second criterium, it is clear from the Bode diagrams that higher frequencies are governed by lower amplitudes. Also when looking at the time response of the flexible modes in Figure 3.3, it becomes clear that the higher modes are only significant if high accuracies are required and one may question if a model can describe a real system that accurate. The model will be truncated to two flexible modes.

3.5 Zerodynamics

In this section the special structure of the zerodynamics is further analysed. For simplicity damping is neglected.

3.5.1 System dynamics in the normal form

To be able to analyse the internal dynamics the system is transformed to the normal form. First we define the relative degree for the system with a finite number of flexible modes included in the model. The relative degree, r, for an affine system at a point $x = x_0$ is defined as (L represents a Lie-derivative):

$$L_g L_f^k h(x) = 0 \quad \forall x \; near \; x_0, \; \forall k < r-1 \tag{3.3}$$

$$L_g L_f^{r-1} h(x_0) \neq 0 \tag{3.4}$$

where the system is defined as:

$$\dot{x} = f(x) + g(x)u \tag{3.5}$$
$$y = h(x) \tag{3.6}$$

For a linear time invariant system the relative degree is therefore:

$$CA^k x = 0 \quad \forall k < r-1 \tag{3.7}$$
$$CA^{r-1}B \neq 0 \tag{3.8}$$

and the relative degree for a linear system turns out to be the difference in the number of poles and zeros of the model. When the tip position is chosen as output and a finite number of flexible modes is included, the relative degree is two. In order to analyse the zerodynamics, the system is input-output decoupled using a coordinate transformation to the normal form. The new coordinates (ϕ) are related to the x via:

$$\phi = Tx \tag{3.9}$$

The first two new coordinates are given by:

$$\phi_1(x) = Cx \tag{3.10}$$
$$\phi_2(x) = CAx \tag{3.11}$$

The additional coordinates are chosen such that they complete the transformation and they can be chosen such that the input doesn't appear explicitely in these new coordinates:

$$\frac{\partial \phi_i(x)}{\partial x} B = 0 \quad \forall\, r+1 \leq i \leq 2+2nf \tag{3.12}$$

A possible transformationmatrix satisfying property (3.12):

$$T = \begin{bmatrix} \mathbf{C} & & & \\ \mathbf{CA} & & & \\ \mathbf{0} & \mathbf{I}_{nf} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tau_T & \mathbf{I}_{nf} \end{bmatrix}, \; \tau_T = \begin{bmatrix} -\frac{B_{3+nf}}{B_{2+nf}} \\ \vdots \\ -\frac{B_{2+2nf}}{B_{2+nf}} \end{bmatrix} \tag{3.13}$$

Using these new coordinates the new state is written as:

$$\zeta_i = \phi_i \quad 1 \leq i \leq r \tag{3.14}$$
$$\eta_{i-r} = \phi_i \quad r+1 \leq i \leq 2+2nf \tag{3.15}$$

The system dynamics in the new states is then:

$$\begin{aligned}\begin{bmatrix}\dot{\zeta}\\ \dot{\eta}\end{bmatrix} &= TAT^{-1}\begin{bmatrix}\zeta\\ \eta\end{bmatrix} + TBu \\ &= \hat{A}\begin{bmatrix}\zeta\\ \eta\end{bmatrix} + \hat{B}u\end{aligned} \tag{3.16}$$

where $\hat{A}$ and $\hat{B}$ are seperated accordingly to the new states:

$$\hat{A} = TAT^{-1} = \begin{bmatrix}\hat{A}_1 & \hat{A}_2\\ \hat{A}_3 & \hat{A}_4\end{bmatrix},\ \hat{B} = TB = \begin{bmatrix}\hat{B}_1\\ \hat{B}_2\end{bmatrix} \tag{3.17}$$

where all elements of TB are zero except the r^{th} element. The internal dynamics is then equal to the equations for η. The zerodynamics is obtained by setting the input and output to zero. This implies that the ζ's are zero and the zerodynamics for the flexible link is given by:

$$\dot{\eta} = \hat{A}_4\eta \tag{3.18}$$

The zero locations of the original system are then obtained by determination of the eigenvalues of the matrix $\hat{A}_4$.

3.5.2 Pole-zero patterns

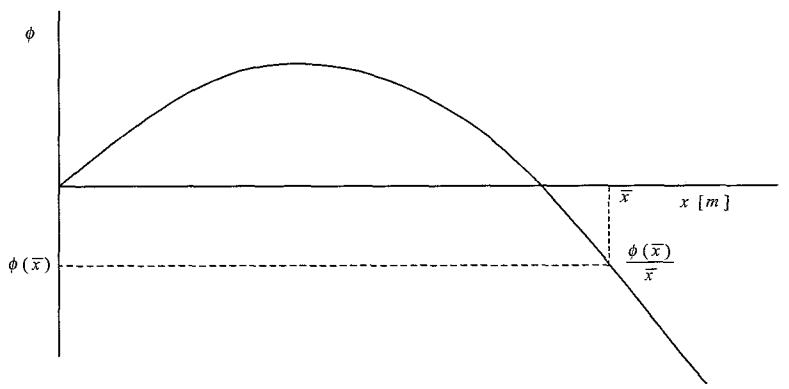
The analysis is restricted to a model with only one or two flexible modes included in the model. The analysis can be extended to more flexible modes but analytical solutions are hard, if not impossible, to obtain. As already mentioned in the previous section, when the tip angle is chosen as output each flexible mode introduces one LHP zero and one RHP zero. Therefore the system with two flexible modes is nonminimum phase. We will now have a look at the location of the zeros when the output is altered. The location of the zeros is studied when the output is moved along the link. The analysis is carried out with the aid of the symbolic computation programme MAPLE (see Appendix D). The equation describing the location of the zeros for a one mode model is (where Z represents a zero):

$$\left(1 + J\phi'(0)\frac{\phi(\overline{x})}{\overline{x}}\right)Z^2 + \omega^2 = 0 \tag{3.19}$$

The solutions to this equation are:

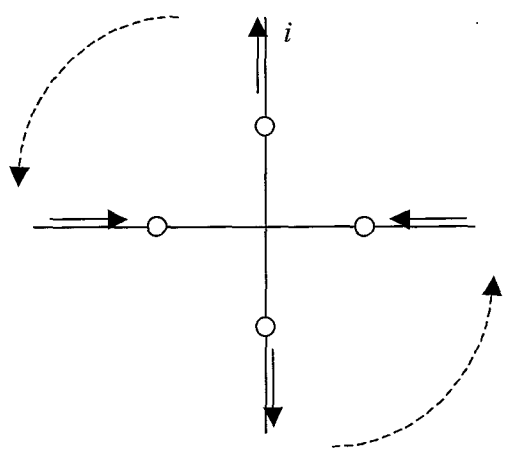
$$Z_1 = \sqrt{-\frac{\omega^2}{1 + J\phi'(0)\frac{\phi(\overline{x})}{\overline{x}}}},\quad Z_2 = -Z_1 \tag{3.20}$$

Beginning at the base of the link the zeros are located on the imaginary axis, symmetrically about the origin. So in this case the system is minimum phase and the zerodynamics is nonhyperbolic. As can be seen from these solutions, the zeros will be pure imaginary if $1 + J\phi'(0)\frac{\phi(\overline{x})}{\overline{x}} > 0$ and the zerodynamics is critically stable. If $1 + J\phi'(0)\frac{\phi(\overline{x})}{\overline{x}} \leq 0$ the zeros will be pure real and the zerodynamics is partly unstable. The zeros appear in a pair and are symmetric about the origin. Assuming fixed J and $\phi'(0)$, the chosen output point on the beam $(\overline{x})$ will determine the nature of the zeros. The collapsing of the zero locations from pure imaginary to pure real will occur at a certain distance after the intersection of the modeshape with the noninertial reference frame.



First modeshape of the flexible link

Furthermore, if $1 + J\phi'(0)\frac{\phi(\overline{x})}{\overline{x}}$ approaches zero from above then $Z_{1,2}$ go towards $\pm i\infty$. If $1 + J\phi'(0)\frac{\phi(\overline{x})}{\overline{x}}$ approaches zero from below then $Z_{1,2}$ approach $\pm\infty$. This completes the pole-zero pattern if one flexible mode is included in the model.



Motion of the zeros for a model with one flexible mode if the output is moved along the link

If two flexible modes are included in the model, the zeros also shift from pure imaginary to pure real. However each zero-pair collapses at a different point from pure imaginary to pure real. The outer zero pair on the imaginary axis goes to infinity and becomes the inner zero pair on the real axis. Then the remaining zero pair goes to infinity on the imaginary axis and becomes the outer pair on the real axis. A similar analysis can be done when more flexible modes are included in the model. The equations describing the zero locations become far more complex. Zeros can shift from imaginary to real and back several times and they aren't always pure imaginary or pure real. This observation can be made when the angles of the zeros are plotted against the measuring point $\overline{x}$ of the link. The general equation describing the zerolocations when nf flexible modes are included in the model is:

$$\left(1+\sum_{i=1}^{nf}\phi_i'(0)\frac{\phi_i(\overline{x})}{\overline{x}}J\right)Z^{2nf}+\left(\sum_{i=1}^{nf}\omega_i^2+\sum_{i=1}^{nf}\sum_{j=1,j\neq i}^{nf}\omega_j^2\phi_i'(0)\frac{\phi_i(\overline{x})}{\overline{x}}J\right)Z^{2nf-2}+\ldots+\prod_{i=1}^{nf}\omega_i^2 \tag{3.21}$$

In appendix A a similar analysis is held when a clamped reference frame is chosen.

3.5.3 Influence of physical parameters on the zerodynamics

The tip of the link is now considered as output. Damping is again neglected. The absolute value of the zeros is determined by the physical parameters of the system. The influence of each parameter is analysed as follows. The zero locations, if one flexible mode is included in the model, is given by formula (3.20). The relation between ω and the physical parameters of the system is given by the relationship (2.30). The nominator of the zeros, $1+J\phi'(0)\frac{\phi(\overline{x})}{\overline{x}}$, can be expressed in terms of β only and β is a function of J_H, ρ and L (using the expression for the characteristic equation and keeping in mind that M_P and J_P are taken zero). Appropriate substitutions then lead to the following relation between the physical parameters of the system and the absolute value of the zeros:

$$|Z|_{1,2}=f(J_H,\rho,L)\sqrt{\frac{EI}{\rho}} \tag{3.22}$$

where $f(J_H,\rho,L)$ represents a function which describes the influence of J_H, ρ and L on the zeros. Already for this one mode model this function couldn't be analytically solved by MAPLE because the expressions become too large. Therefore the influence of these parameters is analytically unknown. However, plotting the zero locations for a fixed L against β , reveals that the function $f(J_H,\rho)$, where the parameters J_H and ρ are allowed to vary in a physical appropriate range, is quite limited (several percents). The influence of L on the other side is quite large and numerical results suggest that $|Z|$ is proportional to $\frac{1}{L^2}$. The absolute value of the zeros, when the tip angle is chosen

as output, for a one mode flexible model can thus be roughly estimated by:

$$|Z| \approx C_0 \frac{1}{L^2} \sqrt{\frac{EI}{\rho}}, \quad C_0 = \text{constant} \tag{3.23}$$

where the constant C_0 can be determined by a single experiment and $C_0 \approx 8.6$.

Chapter 4

Controller design

First the limitations of classical controllers are regarded. Their performance is evaluated in terms of bandwidth and positioning accuracy. Then two relative new types of controllers are developed: a trajectory tracking controller and a state-to-state controller. It is not the purpose of the project to make a trade off between performance and actuator effort. Only topics from controller design are treated which are typical for nonminimum phase systems. But as in most of the controller design techniques, a fast and accurate control also poses a great demand on the actuator. It is emphasized that it is also not pretended to end up with the "optimal" or "best" control law possible for a flexible link. Only the value of command shaping methods on simulation level is regarded and evaluated.

4.1 Classic control techniques

4.1.1 Feedback from the motor

A standard controller will use feedback from the motor position and/or velocity. Because the actuator and the measurement are at the same place this is often denoted as a collocated control system. The control scheme is given in Figure 4.1. A standard PD-

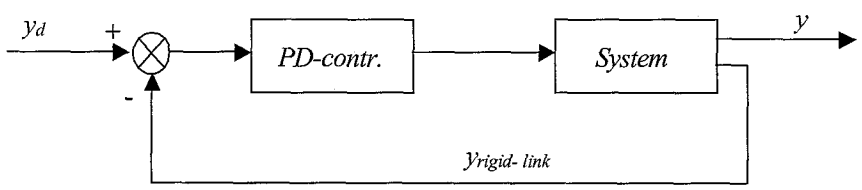


Figure 4.1: Control scheme for a standard collocating PD-controller

controller is capable of controlling the link, assumed to be rigid, with theoretically no bandwidth limitation. However vibrations introduced by the flexibility of the link are not satisfactory surpressed. In Figure 4.2 a typical response for a standard PD-controller which uses feedback from the motor is displayed. A smooth reference trajectory is generated. The rigid body mode will always be able to follow the reference trajectory at the cost of higher demands on the actuator. Vibrations are only damped by the natural damping of the link.

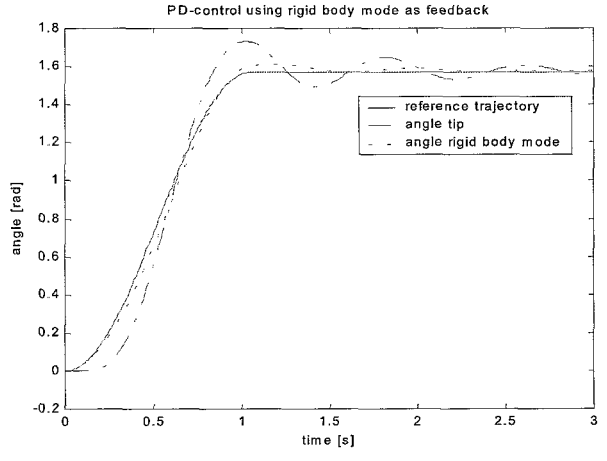


Figure 4.2: Angles under PD-control which uses the angle of the rigid body mode as feedback

This example clearly illustrates the need for a control method which is capable of reducing the vibrations introduced by the flexibility of the link.

4.1.2 Feedback from the tip angle

In the case when feedback is used from the tip angle, the actuator and the sensor are noncollocated. As can be seen in Figure 3.4 the system shows severe phase lag at the first eigenmode. The crossover frequency, i.e. the frequency where $|L(j\omega)|$ (see (4.2)) for the definition of $L(j\omega)$) crosses the 0 dB line for the first time from above, gives us a first indication that the system has a serious bandwidth limitation. In the next subsection a more precise defenition of the bandwidth is given. Because the bandwidth of the closed loop system is smaller than the frequency of the first eigenmode the model is truncated to only one flexible mode. When a PD-controller is applied to the system only small P- and D-values can be chosen in order not to endanger stability. Therefore the reference trajectory is very slow and only low frequencies are apparent in the system. This validates the truncation to one flexible mode because the contribution of higher

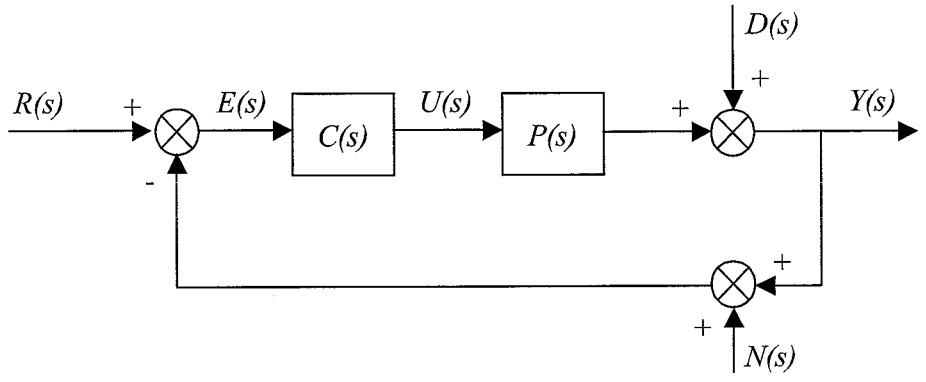


Figure 4.3: General control system

order modes to the output is neglible. When looking at the zerodynamics it is interesting to notice that when the output is chosen sufficiently close to the base of the link, the zerodynamics will be minimum phase. A feedback loop from this point will also have no bandwidth limitation.

Sensitivity: Waterbed formulas

Another drawback of using feedback from the tip angle, besides the bandwidth limitation, is the extreme sensitivity to disturbances when the bandwidth is pushed to its limit. There is a relation between the bandwidth of the system and the sensitivity. In order to analyse the sensitivity, consider the general control system given in Figure 4.3.

The relation between output and inputs is given by the following relation:

$$Y(s) = \frac{C(s)P(s)}{1 + C(s)P(s)}(R(s) - N(s)) + \frac{1}{1 + C(s)P(s)}D(s) \tag{4.1}$$

The sensitivity function ($S(s)$) and the complementary sensitivity function ($T(s)$) are defined as:

$$S(s) = \frac{1}{1 + L(s)}, \quad L(s) = C(s)P(s) \tag{4.2}$$

$$T(s) = \frac{L(s)}{1 + L(s)} \tag{4.3}$$

The sensitivity function is a measure for the sensitivity of the control system to disturbances $D(s)$. For obvious reasons we want $S(s)$ to be small over the whole frequency

range. The bandwidth of the system (ω_B) is defined to be the frequency where $|S(s)|$ crosses $\frac{1}{\sqrt{2}}$ ($\approx -3\ dB$) for the first time[1]. In Figure 4.4 the Bode plots for $S(s)$, $T(s)$ and $L(s)$ are given for $C(s) = 0.1s + 0.1$.It is seen that the bandwidth is approxio-

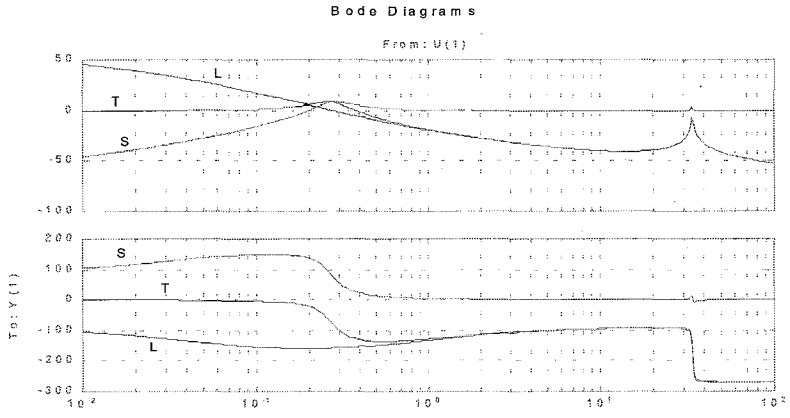


Figure 4.4: Bode diagrams for $L(s)$, $T(s)$ and $S(s)$

mately 0.2 rad/s. The gain and phase margin are $GM = 8\ dB$ at $\omega = 0.3\ rad/s$ and $PM = 23^o$ at $\omega = 40\ rad/s$. In order to increase the bandwidth of the system one may try to increase the values for P and/or D. When doing this the sensitivity function changes which is known as the waterbed effect. The waterbed effect holds if the loop transfer function ($L(s)$) has a least two more poles than zeros or if $L(s)$ has one RHP zero. Because a PD-controller is used $L(s)$ satisfies the second condition and the second waterbed formula applies. For the sensitivity function the following integral holds (2^{nd} waterbed formula):

$$\int_0^\infty \ln|S(j\omega)| f_w(z,\omega) d\omega = 0$$

where the weighting function $f_w(z,\omega)$ is:

$$f_w(z,\omega) = \frac{2z}{z^2+\omega^2} = \frac{2}{z}\frac{1}{1+(\frac{\omega}{z})^2}$$

where z represents the RHP zero. Because the weighting function falls sharply for frequencies beyond the frequency of the RHP zero, the sensitivity integral can be ap-

[1] In another often used definition, the bandwidth is defined to be the frequency where $|T(s)|$ crosses $\frac{1}{\sqrt{2}}$ ($\approx -3\ dB$) for the first time from above. However for systems with zero(s) in the RHP it can be a better choice to use the definition which involves $|S(s)|$.

proximated by

$$\int_0^z \ln|S(j\omega)|d\omega \approx 0$$

The integral tells us that the total area of $|S(j\omega)|$, from $\omega = 0$ to $\omega = z$, under and above the 0 dB line is approximately zero. The waterbed effect is then explained by looking at the shape of the sensitivity function. For low frequencies "negative area" builds up because $|S(j\omega)| < 0$ dB. This is compensated by a peak in $|S(j\omega)|$ somewhere between ω_B and the frequency of the RHP zero. Pushing the magnitude of the sensitivity function down at low frequencies, in order to increase the bandwidth of the system, will therefore lead to a higher peak at the intermediate frequencies. In order to obtain good performance and robustness it is usually required that the peak value of $|S(j\omega)|$, M_s, is smaller than 2 dB. The waterbed effect is illustrated by using $D = 0.2$ and for P three values are taken for which the nominal system is still stable: $P = [0.05\ 0.5\ 5]$. In Figure 4.5 the effect of varying proportional gain is visualised for the magnitude of the sensitivity function.

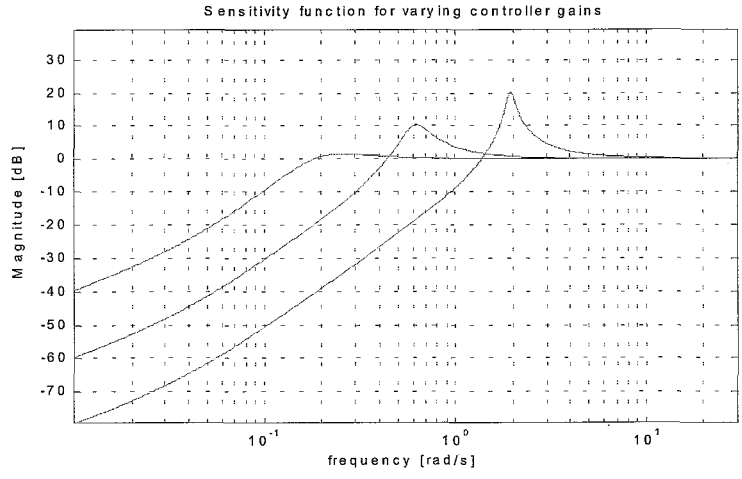


Figure 4.5: Magnitude of the sensitivity function for $P_1 = 0.05$ (left), $P_2 = 0.5$, $P_3 = 5$ (right)

There is a relationship between $|S(j\omega)|$ and the gain and phase margin. A higher peak of $|S(j\omega)|$ will in general also lead to smaller gain and phase margins.

4.2 Trajectory tracking

4.2.1 Problem definition

As seen in the previous sections, the classical PD-controller has several limitations. A PD-controller which uses feedback from the motor will have a, theoretically, unlimited bandwidth but it is uncapable of damping the vibrations introduced by the flexibility of the link. On the other side, a PD-controller which uses feedback from the tip has a serious bandwidth limitation due to the RHP zeros. For output tracking it is possible to use inversion techniques. However for a nonminimum phase system the classical inverse is unstable because the zeros of the original system become the poles of the inverse system. It is still possible to design a control law based on inversion techniques, but this control law will be noncausal. The interpretation of this is that a forward system (the physical system) maps a given input to an output and is necessarily causal. The inverse system on the other hand, maps the output to the input and this need not to be a physical system. So the ouput-input mapping can be noncausal.

The problem for the stable inversion approach is stated as follows: an input function ($u_{ff}(t)$) and state trajectory ($x_{ref}(t)$) are to be found, which satisfy the system equations for all t and the reference trajectory has to map into the desired reference trajectory:

$$\dot{x}_{ref}(t) = Ax_{ref}(t) + Bu_{ff}(t) \quad \forall t \in \{-\infty, \infty\} \tag{4.4}$$

$$y_d(t) = Cx_{ref}(t) \tag{4.5}$$

where $y_d(t)$ is the desired output which has to be tracked. It is also required that $u_{ff}(t)$ and $x_{ref}(t)$ are bounded:

$$u_{ff}(t) \to 0,\ x_{ref}(t) \to 0\ as\ t \to \pm\infty \tag{4.6}$$

If this problem has a solution then the system is stable invertible. The first step in the controller design is the inversion step and can be done offline. The second step is the design of a feedback law which will stabilize the system around the desired state trajectories. The second step can be seen as a robustifying step. The control scheme is displayed in Figure 4.6.

4.2.2 Stable inversion scheme

The dynamic model of the flexible link is linear and for our system the relative degree, r, is two. Taking the r^{th} derivate of the output yields:

$$\frac{d^r y(t)}{dt^r} = CA^r x(t) + CA^{r-1}Bu(t) \tag{4.7}$$

$$= A_x x(t) + B_y u(t) \tag{4.8}$$

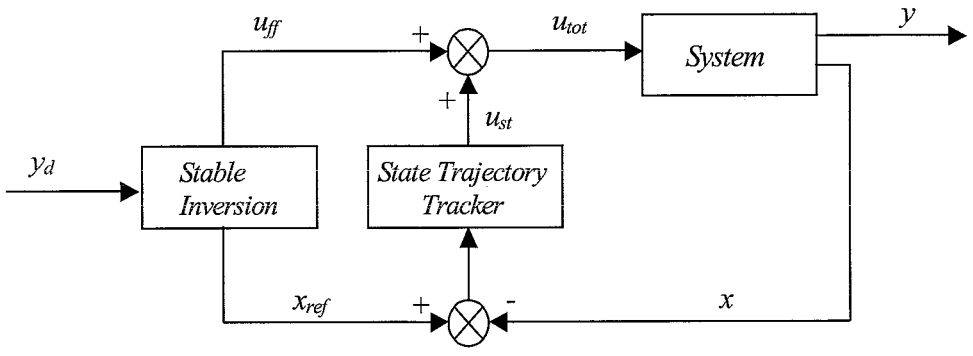


Figure 4.6: Control scheme of the stable inversion controller

This motivates the following control law:

$$u_{ff}(t) = B_y^{-1}(\overset{[r]}{y}_d(t) - A_x x(t)) \quad \forall t \in \{-\infty, \infty\} \tag{4.9}$$

By substituting this control law into equation (4.8) it can be seen that exact tracking, i.e. $\overset{[r]}{y} = \overset{[r]}{y}_d(t)$, is maintained. Using the coordinate transformation of (3.9), where the transformation matrix is defined by (3.13), the effect of this control law can be examined. For the derivation of the feedforward control law, property (3.12) does not necesarally have to hold. Any complete coordinate transformation will do. However for a single input-single output system it is always possible to define a transformationmatrix which satisfies that property and the derivation is simplified. The new states are then defined by (3.14) and (3.15). The desired $\zeta(t)$ is defined as $\zeta_d(t)$. Because the input was designed such that exact tracking was maintained this also implies $\dot{\zeta}(t) = \dot{\zeta}_d(t)$. The system dynamics in the new coordinates (the inverse system) becomes:

$$\dot{\zeta}(t) = \dot{\zeta}_d(t) \tag{4.10}$$
$$\dot{\eta}(t) = \hat{A}_3\zeta_d(t) + \hat{A}_4\eta(t) \tag{4.11}$$

and if a bounded solution to the internal dynamics can be found then there exists a solution to the stable inversion scheme. For the system in new coordinates, the ζ_d can be viewed as the new inputs to the system. The feedforward input which yields exact tracking of the output can be written as:

$$u_{ff}(t) = B_y^{-1}(\overset{[r]}{y}_d(t) - A_\zeta\zeta_d(t) - A_\eta\eta_d(t)) \tag{4.12}$$

where:

$$\begin{bmatrix} A_\zeta \\ A_\eta \end{bmatrix} = A_x T^{-1} \tag{4.13}$$

and the reference trajectories:

$$x_{ref}(t) = T^{-1} \begin{bmatrix} \zeta_d(t) \\ \eta_d(t) \end{bmatrix} \tag{4.14}$$

For hyperbolic internal dynamics, as in the case of the flexible link when the ouput is defined to be the tip angle, the internal dynamics can be decoupled into a stable and unstable subsystem according to the eigenvalues of $\hat{A}_4$:

$$\sigma(t) = \begin{bmatrix} \sigma_s(t) \\ \sigma_u(t) \end{bmatrix} = U\eta(t) \tag{4.15}$$

Applying this to the internal dynamics given by equation (4.11) yields:

$$\dot{\sigma}_s(t) = A_s\sigma_s(t) + B_s\zeta_d(t) \tag{4.16}$$
$$\dot{\sigma}_u(t) = A_u\sigma_u(t) + B_u\zeta_d(t) \tag{4.17}$$

Bounded solutions to the internal dynamics can then be found as:

$$\sigma_s(t) = \int_{-\infty}^{t} e^{A_s(t-\tau)}B_s\zeta_d(\tau)d\tau \tag{4.18}$$
$$\sigma_u(t) = -\int_{t}^{\infty} e^{-A_u(\tau-t)}B_u\zeta_d(\tau)d\tau \tag{4.19}$$

These bounded solutions show that for a bounded solution to the unstable subsystem all future information is needed. For the stable subsystem however only past information is needed. Substituting the bounded solutions into the equation for the feedforward input yields the following feedforward input:

$$u_{ff}(t) = B_y^{-1}(\overset{[r]}{y}_d(t) - A_\zeta\zeta_d(t) - A_\eta U_s\sigma_s(t) - A_\eta U_u\sigma_u(t)) \tag{4.20}$$

where

$$U^{-1} = \begin{bmatrix} U_s & U_u \end{bmatrix} \tag{4.21}$$

is partitioned according to the partitioning in (4.15). The transformations for the stable inversion scheme are displayed in Figure 4.7.

In practice a preview time (T_p) will be used to approximate the solution to the unstable internal dynamics and the bounded solution to the unstable part of the internal dynamics (4.19) becomes:

$$\sigma_u(t) = -\int_{t}^{t+Tp} e^{-A_u(\tau-t)}B_u\zeta_d(\tau)d\tau \tag{4.22}$$

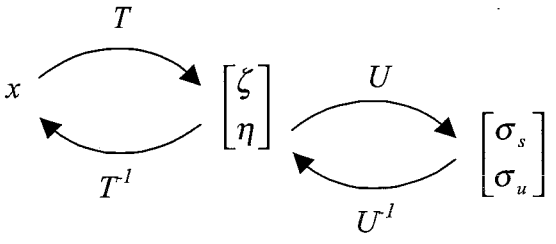


Figure 4.7: Representation of the transformations for the stable inversion scheme

For using a certain preview time, stability still has to be garanteed. In appendix B it is proven that stability is garanteed for all preview time but a short preview time will cause tracking errors. The preview time needed to keep the maximum tracking error bounded depends on the zero locations. Zeros with a large real part will show fast convergence and little preview time is needed. Nonminimum phase systems with zeros located close to the imaginary axis will need large preview time to obtain accurate output tracking.

4.2.3 Feedback scheme

In practice a system is rarely open-loop controlled. Namely for a purely open-loop controlled system Figure 4.3 is redrawn as shown in Figure 4.8.

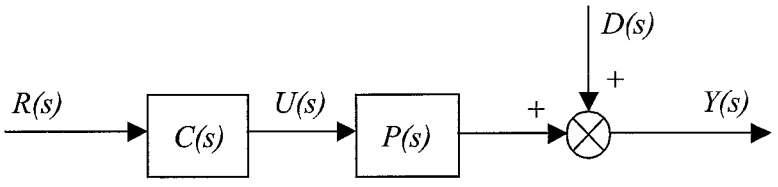


Figure 4.8: Purely open-loop controlled system

and it is clear that disturbances $D(s)$ are fully present in the output signal. Also plant variations ($\triangle P(s)$) deteriorate performance of the system. The stable inversion approach is therefore completed by using a state trajectory tracker. A drawback of this method is that the full state has to be available. Because the controllability matrix (Γ_c) and observability matrix (Γ_o) have full rank, the system is completely controllable and

observable:

$$rank(\Gamma_c) = n_{tot}, \quad \Gamma_c = \begin{bmatrix} B & AB & \cdots & A^{n_{tot}-1}B \end{bmatrix} \tag{4.23}$$

$$rank(\Gamma_o) = n_{tot}, \quad \Gamma_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n_{tot}-1} \end{bmatrix} \tag{4.24}$$

The state can be thus be recovered from the output using an observer. The trajectory tracker can be of simple state feedback type:

$$u_{st}(t) = K\ [x(t) - x_{ref}(t)] \tag{4.25}$$

4.2.4 Application to the flexible link

The stable inversion approach is used to let the tip of the link follow a specified trajectory. The feedforward input is computed off-line. For the nominal model only the feedforward input should give satisfactory results when preview time is chosen large enough. In Figure 4.9 the tip motion is displayed for a preview time of $T_p = 1$ $[s]$. In this case no state trajectory tracker is included.

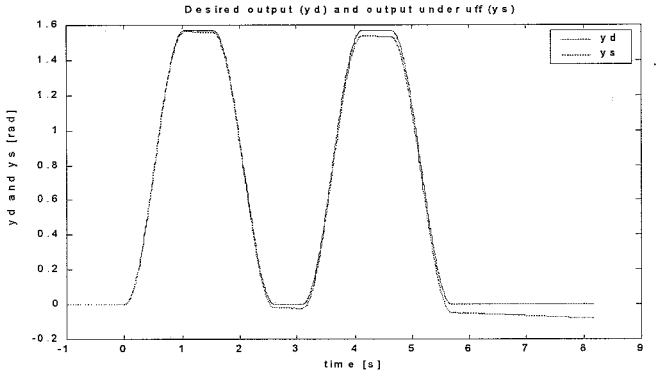


Figure 4.9: Response of a feedforward input with $T_p = 1$ $[s]$

Because no state trajectory tracker is included, the input is approximated by interpolation between discrete time values and the preview time is chosen as $T_p = 1$ $[s]$, the response "drifts" away from the reference trajectory. This behaviour is typical for a purely open-loop controlled system and this illustrates the need for a state trajectory tracker. The feedforward input is displayed in Figure 4.10. The noncausality is seen in this figure because the input is non zero for t 0.

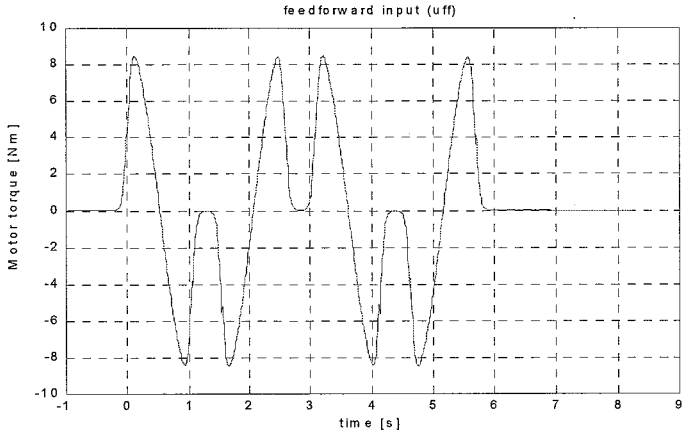


Figure 4.10: The noncausal feedforward input

For the nominal model the maximum error is given by (see Appendix B):

$$||e_y(t)||_2 \leq \frac{||C||_2||B||_2\ M_2\lambda}{\alpha\beta}e^{-\alpha T_p}$$

where all the parameters are fixed for a given model except the preview time T_p. It is interesting to see the relation between the RHP zeros of the system, the preview time and the maximum error[2]. The value of α is determined by the location of the RHP zeros in the complex plane. RHP zeros located close to the imaginary axis are accompanied by small values for α and for RHP zeros far from the imaginary axis a large value for α can be chosen. So systems with "slow" zerodynamics will need more preview time in order to keep the maximum error below a certain limit than systems with "fast" zerodynamics.

Now a state trajectory tracker of type (4.25) with $K = \begin{bmatrix} 10 & 0 & 0 & 10 & 0 & 0 \end{bmatrix}$ is included. This feedback law only prevents "drifting" of the output since only the rigid body mode is fed back. The error for the system with a preview time of one second is then given in Figure 4.11.

4.3 State-to-state transfer

State-to-state transfer is a motion from one state to another. A rest-to-rest transfer is a special case in which the system is reconfigured from one equilibrium point to another. First the formal method for the general state-to-state transfer is outlined. Then a rest-to-rest transfer is applied to the flexible link.

[2]For convenience $y_d(t) - y_s(t)$ is denoted as the error. Strictly speaking, one cannot use the term "the error" because the difference $y_d(t) - y_s(t)$ is a time dependent signal .

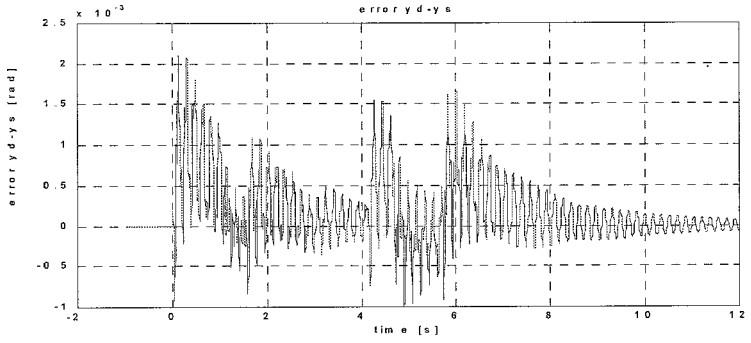


Figure 4.11: The error ($e_y(t) = y_d(t) - y(t)$) using feedback

4.3.1 Formal method

Solving the state-to-state transfer involves several steps. First a new output function is defined which gives the system a relative degree equal to the number of states. This output function, together with its $2nf + 1$ derivates, is used for a new state representation of the system. Thus the system is turned into its normal form:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ &\vdots \\ \dot{z}_{n_{tot}-1} &= z_{n_{tot}} \\ \dot{z}_{n_{tot}} &= a(z) + b(z)u \end{aligned} \tag{4.26}$$

Then the control law is chosen:

$$u = \frac{v - a(z)}{b\,(z)} \tag{4.27}$$

Substituting this control law into (4.26) yields $\dot{z}_n = v$. The new input, v, is chosen such that is drives the system from the first desired state to the other. For that purpose an interpolating function, $y_d(t)$, is defined which drives the system from the initial state to the desired state in the desired time span T_f. The new input is then chosen as $v = {}^{[ntot]}y_d$.

The condition which the output function has to satisfy in order to gain relative degree $r = n_{tot}$ is:

$$CA^kB = 0 \quad \forall k < r - 1 \tag{4.28}$$

$$CA^kB \neq 0 \quad k = r - 1 \tag{4.29}$$

So for a system with two flexible modes included, $n_{tot} = 6$ and the output function has to satisfy 4 conditions, as is apparent from the first equation. Choosing as output

function:

$$y = \theta + \sum_{i=1}^{nf} c_i q_i = \theta + c^T q + c^T \dot{q} \tag{4.30}$$

and the coefficents c are still to be determined by solving the set of linear equations. When damping is neglected, powers of the matrix A have a special blockdiagonal structure:

$$A^k = \begin{bmatrix} \mathbf{0} & (-1)^{\frac{k-1}{2}}\omega^{k-1} \\ (-1)^{\frac{k+1}{2}}\omega^{k+1} & \mathbf{0} \end{bmatrix} \quad \forall k \in \{3,5,\dots\} \tag{4.31}$$

$$A^k = \begin{bmatrix} (-1)^{\frac{k}{2}}\omega^k & \mathbf{0} \\ \mathbf{0} & (-1)^{\frac{k}{2}}\omega^k \end{bmatrix} \quad \forall k \in \{2,4,\dots\} \tag{4.32}$$

$$\omega = diag\{\ 0 \quad \omega_1 \quad \dots \quad \omega_{nf}\ \} \tag{4.33}$$

and the first condition reduces to only nf equations because for $k \in \{2,4,\dots\}$ the condition is always satisfied. This enables us to use the following output matrix which is similar to the one already defined.:

$$\begin{aligned} y &= Cx \\ &= \begin{bmatrix} 1 & c_1 & \cdots & c_{nf} & 0 & \cdots & 0 \end{bmatrix} x \end{aligned} \tag{4.34}$$

The coefficents c_i are then determined by solving the following linear system:

$$V \cdot diag\{\begin{bmatrix} \phi\prime_1(0) & \cdots & \phi\prime_{nf}(0) \end{bmatrix}\} \cdot c = b \tag{4.35}$$

where

$$V = \begin{bmatrix} 1 & \cdots & 1 \\ \omega_1^2 & \cdots & \omega_{nf}^2 \\ \vdots & & \vdots \\ \omega_1^{2nf-2} & \cdots & \omega_{nf}^{2nf-2} \end{bmatrix}, c = \begin{bmatrix} c_1 \\ \vdots \\ c_{nf} \end{bmatrix}, b = \begin{bmatrix} \frac{1}{J} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

This new output function and its derivates are used as new state variables and the system is turned into its normal form. These coordinates are related to the old ones by:

$$z = Tx, \quad T = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n_{tot}-1} \end{bmatrix} \tag{4.36}$$

The nominal input is then given by:

$$\tau_d = \frac{y_d^{[n_{tot}]} - CA^{n_{tot}}x}{CA^{n_{tot}-1}B} \tag{4.37a}$$

4.3.2 Application to the flexible link

A rest-to-rest maneuver is considered. The end point of the link is reconfigured from one equilibrium point to another in a specific time span T_f. The initial and final states are given in the original coordinates as:

$$\begin{bmatrix} \theta(0) \\ q_1(0) \\ \vdots \\ q_{nf}(0) \\ \dot{\theta}(0) \\ \dot{q}_1(0) \\ \vdots \\ \dot{q}_{nf}(0) \end{bmatrix} = \mathbf{0}, \quad \begin{bmatrix} \theta(T_f) \\ q_1(T_f) \\ \vdots \\ q_{nf}(T_f) \\ \dot{\theta}(T_f) \\ \dot{q}_1(T_f) \\ \vdots \\ \dot{q}_{nf}(T_f) \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ \mathbf{0} \end{bmatrix} \tag{4.38}$$

Using the coordinate transformation, these initial and final states are mapped into desired initial and final states of the new states. The initial and final states of the states z are used to design an output trajectory $y_d(t)$ which satisfies these conditions. For this a polynomial of degree $4nf + 5$ is used:

$$y_d = a_1 t^{4nf+5} + a_2 t^{4nf+4} + \cdots + a_{4nf+6} \tag{4.39}$$

where the coefficents, a_i, are uniquely determined by imposing the boundary conditions of the interpolating polynomial and its derivates. The polynomial brings the system from the intial state to the final state and is such that the conditions at the boundary of the time interval are met. It also accounts for zero torque at the boundary of the time interval. Using the dynamic model with two flexible modes included, the polynomial for the rest-to-rest maneuver in $T_f = 1$ $[s]$ is seen in Figure 4.12. As is seen in the figure, the real tip angle (y_m), obtained after transformation of the new defined output function which gains maximum relative degree, oscillates around the design trajectory (y_d) but this effect vanishes exactly at the final time T_f. A smaller T_f leads to larger differences between the design output and the model output at intermediate times but the control law garantees that the design output and the model output exactly coincide at the final time.

4.4 Comparison of controllers for the flexible link

In section 4.1 the disadvantages of classical control techniques are shown. In the case of a PD-controller with a feedback loop from the rigid body mode, high speed of motions is allowed without endangering stability but the endpoint positioning will be inaccurate due to residual vibrations. A feedback loop from the endpoint of the link allows only

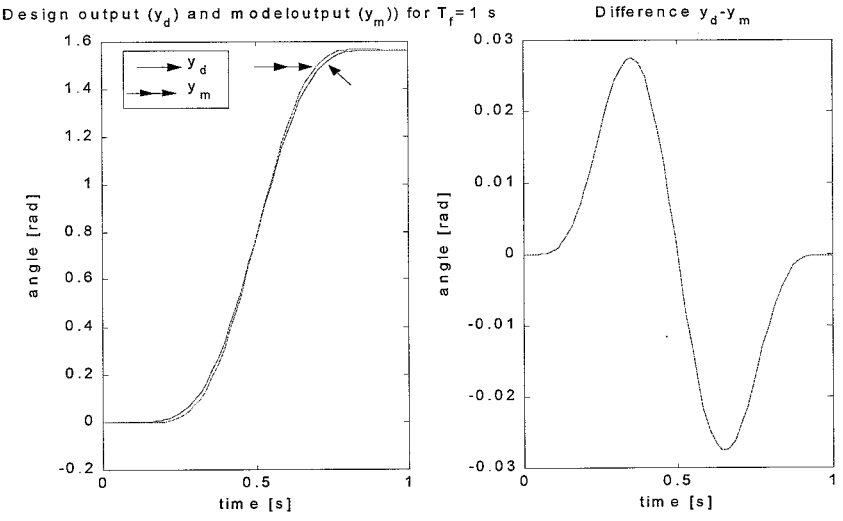


Figure 4.12: (left) Interpolating and tip position (right) Difference between interpolating trajectory and tipposition

very low speed of motions in order not to endanger stability and consequently endpoint accuracy will be better.

The preview-based stable inversion control law relies heavily on the dynamic model of the flexible link. This model depence is both its strength and weakness. Regarding the speed of response, this control method has for the nominal model theoretically no bandwidth limitation. A faster response poses only greater demands on the actuator. If the model is correct and enough preview time is taken then the tracking error can be kept small. Looking at the complexity of the complete control system it is seen that when an accurate dynamic model is available then only motor position and velocity measurements are sufficient to prevent "drifting". If the complete state of the system is fed back then the full state has to be measured or recovered from the output using an observer. Although no extensive analysis concerning robustness of the controlled system is done, some general notions are in place. In the dynamic model all kinds of effects are ignored, e.g. torsion of the link and hub dynamics like Coulomb friction. These effects will be present in a practical robotic system and can therefore degrade the performance of the controlled system. Also the damping factor of the flexible modes is assumed to be $\zeta_e = 0.01$ but in practice the damping factors of the flexible modes will be hard to measure and thus inaccurate. For a pertubed plant or when disturbances act on the output, good tracking can only be achieved using a state trajectory tracker. But in this case 6 trajectories have to be tracked with only one actuator. This is a heavily underactuated situation and several control aims, like good tracking, disturbance reduction and sensor noise attenuation, are in conflict and cannot all be accomplished in the frequency band of interest.

The state-to-state control law relies on exact cancelation of vibrations at the end of the motion task. Using the right interpolating trajectory and leaving the actuator out of consideration, the state-to-state maneuver can be accomplished in any predefined time.

Both proposed control laws work well for the nominal model and have no bandwidth limitation as in the classical PD-controller design which uses feedback from the tip. The accuracy is dependent on the number of flexible modes included in the model. When dealing with a non nominal model, the poles and zeros of the system are different from the nominal model. Using the new proposed control laws which are based on the nominal model, exact cancellation of vibrations isn't garanteed anymore. So for a practical system the bandwidth for which accurate positioning is possible, is likely to be also limited to around the frequency of the first flexible mode as in the case of the classical control techniques. Experimental data is necessary to be able to address robustness issues more rigorously.

Chapter 5

Conclusions

- Two types of models for a single rotating flexible link are derived using the assumed modes approach: the psuedo-pinned and the pseudo-clamped model.
- The number of flexible modes included in the model, in order to describe the dynamic behaviour of the system in the frequency range of interest with the desired accuracy, is argued to be two.
- For a one mode and two mode flexible link, the zerodynamics is investigated and analytic solutions are found. These solutions show the nonminimum phase behaviour of the system if the output is taken to be the tip angle. The solutions also provide insight in the behaviour of the zero locations in the complex plane when the output is chosen as the angle pointing to a general point on the link.
- Analytic solutions for the relation between the absolute value of the zeros and the physical paramters of the system in the model are, already for a one mode flexible model, hard to obtain.
- Simulations show the performance limitations of classical control techniques (in particular PD-controlled flexible links):
  - A PD-controller which uses feedback from the motor has theoretically no bandwidth limitation but performance is poor because of residual vibrations.
  - The bandwidth of a PD-controlled system which uses feedback from the tip angle is severely limited. The bandwidth cannot be pushed beyond the frequency of the right half plane zero without stability problems. The practical bandwidth of the system is shown, using a sensitivity analysis, to be even smaller due to disturbance amplification.
- Preview-based stable inversion control is capable of exact trajectory tracking for the nominal model.

- State-to-state transfer in any predefined time can be achieved using a feedback linearizing input.
- Both the stable inversion approach and the state-to-state transfer have theoretically for the nominal model no bandwidth limitation
- Because the preview-based stable-inverse control law and the state-to-state conroller rely on exact cancellation of vibrations, their bandwidth for a practical system, where disturbances and model errors are always present, is likely to be limited to the frequency of the first flexible mode as in the case of the classical PD-controllers.

Bibliography

[1] A. Isidori, *Nonlinear control systems*, Springer-Verlag, Berlin, 3^{rd} edition, 1995

[2] C. Canudas de Wit, B. Siliciano and G. Bastin, *Theory of Robot Control*, Springer-Verlag, London, 1996

[3] J. van de Vegte, *Feedback control systems*, Prentice-Hall, 2^{nd} edition, 1990

[4] L. Meirovitch, *Principles and techniques of vibrations*, Prentice-Hall, 1997.

[5] S. Skogestad and I. Postlewaithe, *Multivariable Feedback Control: Analysis and design*, John Wiley & Sons Ltd, Chichester, England, 1996.

[6] R. H. Cannon Jr. and Eric Schmitz, *Precise Control of Flexible Manipulators*, Robotics Research, MIT press, Massachussets, 1984.

[7] F. Bellezza, L. Lanari and G. Ulivi, *Exact modeling of the flexible slewing link*, Proc. of the 1990 IEEE International Conference on Robotics and Automation, p. 734-739.

[8] D. K. Lindner, K. M. Reichard and L. M. Tarkenton, *Zeros of Models of Flexible Structures*, IEEE Transactions on automatic control, vol. 38, no. 9, September 1993, p. 1384-1387.

[9] Q. Zou and S. Devasia, *Preview-Based Stable-Inversion for Output Tracking*, Proc. of the American Control Conference, San Diego, California, June 1999, p. 3544-3548.

[10] A. De Luca, *Feedforward/Feedback Laws for the control of Flexible Robots*, Proc. of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, April 2000, p.233-240.

[11] D. Chen and B. Paden, Stable inversion of nonlinear nonminimum phase systems, Int. J. of Control 1996 vol 64, no 1, p. 81-97.

[12] F. Khorrami, Dynamical properties of manipulators exhibiting flexibilities, Proc. of the 1990 IEEE International Conference on Systems Engineering, 1990, p. 1-4.

[13] D. Wang and M. Vidyasagar, *Transfer Functions for a Single Flexible Link*, Proc. of the 1989 IEEE International Conference on Robotics and Automation, August 1989, p. 1042-1047.

[14] H. Yang, H. Krishnan and M.H. Ang Jr., *A simple rest-to-rest control command for a flexible link robot*, Proc. of the 1997 IEEE International Conference on Robotics and Automation, April 1997, p. 3312-3317.

[15] S. S. Ge, T. H. Lee and G. Zhu, *Improving Joint PD Control of Single-Link Flexible Robots by Strain/Tip Feedback*, Proc. of the 1996 IEEE International Conference on Control Applications, Dearborn, September 1996, p.965-969.

[16] S. Devasia, *Stable Inversion for Nonlinear Systems with Nonhyperbolic Internal Dynamics,* Conference on Decision and Control, San Diego, December 1997, p. 2882-2888.

Appendix A

Zerodynamics for the pseudo-clamped beam

In this appendix the zerodynamics for a pseudo-clamped beam is analysed. If only one flexible mode is included in the model and damping is neglected, the equation which describes the zerolocations is given by:

$$Z_1 = \sqrt{-\frac{\omega_1^2 J}{J + \mu_1^2 - \mu_1 \frac{\phi(\overline{x})}{\overline{x}} J}}, \quad Z_2 = -Z_1 \tag{A.1}$$

As in the pseudo-pinned case, the zeros change from pure imaginary to a pure positive and negative real zero if the output is moved from the base of the link to the tip of the link. The zeros are pure imaginary if $J + \mu_1^2 - \mu_1 \frac{\phi(\overline{x})}{\overline{x}} J > 0$ and the zeros are pure real if $J + \mu_1^2 - \mu_1 \frac{\phi(\overline{x})}{\overline{x}} J \quad 0$. For the general dynamic model the zerolocations are described by means of a complex equation. For completeness also the equation for a model with 2 flexible modes is given:

$$\begin{aligned}(-J^2 - J\mu_2^2 - \mu_1^2 J - \mu_1^2\mu_2^2 + \mu_1 J^2 \frac{\phi_1(\overline{x})}{\overline{x}} + \mu_1 J \frac{\phi_1(\overline{x})}{\overline{x}} \mu_2^2 + \mu_2 J^2 \frac{\phi_2(\overline{x})}{\overline{x}} \\ + \mu_2 J \frac{\phi_2(\overline{x})}{\overline{x}} \mu_1^2) Z^4 + (J^2\omega_1^2\mu_2 \frac{\phi_2(\overline{x})}{\overline{x}} - J\omega_2^2\mu_1^2 - J^2\omega_1^2 - J^2\omega_2^2 + J^2\omega_2^2\mu_1 \frac{\phi_1(\overline{x})}{\overline{x}} \\ - J\omega_1^2\mu_2^2) Z^2 - J^2\omega_1^2\omega_2^2 = 0\end{aligned}$$

These equations are obtained by using a Maple programme (see Appendix D).

Appendix B

Stability proof of the stable inverse control law

The proof of stability for the stable inverse control law consists of two parts and is valid for the nominal system only. First a proof is given that an approximation of the computation of the unstable part of the zerodynamics, by applying a certain preview time T_p, leads to a bounded error in the computation of the feedforward input. Then this proof is used to show that a bounded error in the computation of the feedforward input leads to a bounded tracking error.

B.1 Part 1

Using a certain preview time, T_p, the solution to the unstable part of the zerodynamics is approximated by (see (4.19) and (4.22)):

$$\overline{\sigma}_u(t) = -\int_t^{t+T_p} e^{-A_u(\tau-t)} B_u \zeta_d(\tau) d\tau \tag{B.1}$$

where the desired trajectory is specified in ζ_d. The error in the computation of the solution to the unstable zerodynamics is then defined as:

$$e_{\sigma_u}(t) = \sigma_u(t) - \overline{\sigma}_u(t) \tag{B.2}$$

Using the vector 2-norm (Euclidian vector norm), $||z||_2$, the error is expressed as:

$$||e_{\sigma_u}(t)||_2 = || \int_{t+T_p}^{\infty} e^{-A_u(\tau-t)} B_u \zeta_d(\tau) d\tau ||_2 \tag{B.3}$$

The matrix $-A_u$ is Hurwitz (i.e. all eigenvalues of $-A_u$ are < 0) and therefore the following property holds:

$$||e^{-A_u t}||_2 \leq \kappa e^{-\alpha t}, \quad \kappa, \alpha > 0 \tag{B.4}$$

Assuming that ζ_d and all its derivates are bounded, $||\zeta_d|| \leq M, \ \forall t \geq 0$, and using the above property, (B.3) reduces to:

$$||e_{\sigma_u}(t)||_2 \leq \frac{M||B_u||_2\kappa}{\alpha} e^{-\alpha T_p} \tag{B.5}$$

The error in the computation of the unstable part of the zerodynamics, e_{σ_u}, causes an error in the computation of the feedforward input ($e_u(t)$):

$$e_u(t) = \tilde{u}_{ff}(t) - u_{ff}(t) \tag{B.6}$$

where $\tilde{u}_{ff}(t)$ represents the computated feedforward input which uses the approximated solution to the unstable part of the zerodynamics (B.1). The following expressing can be found by using (B.6), (B.2) and (4.20):

$$\sup_t ||e_u(t)||_2 \leq \frac{\lambda}{\alpha} e^{-\alpha T_p}, \quad \lambda > 0 \tag{B.7}$$

where λ is a positive scalar.

B.2 Part 2

For the system the following errors are defined:

$$e_x(t) = x(t) - x_{ref}(t) \tag{B.8}$$
$$e_y(t) = y(t) - y_d(t) \tag{B.9}$$

The dynamics of the state error and the tracking error are:

$$\dot{e}_x(t) = Ae_x(t) + Be_u(t) \tag{B.10}$$
$$e_y(t) = Ce_x(t) \tag{B.11}$$

Substituting the solution into (B.10) and into (B.11) yields:

$$||e_y(t)||_2 \leq ||C||_2||B||_2 \sup_\tau ||e_u(\tau)||_2 \int_{-\infty}^{t} ||e^{A(t-\tau)}||_2 d\tau \tag{B.12}$$

For a stable system, i.e. all eigenvalues of A are < 0, the following property holds:

$$||e^{A(t-\tau)}||_2 \leq M_2 e^{-\beta(t-\tau)} \quad M_2, \beta > 0 \tag{B.13}$$

The tracking error is then written as:

$$||e_y(t)||_2 \leq \frac{||C||_2||B||_2 M_2 \lambda}{\alpha\beta} e^{-\alpha T_p} \tag{B.14}$$

Appendix C

Matlab files

C.1 Dynamic models for the flexible link

```
% DYNMODELS  Dynamic models for a single flexible link
%
% A dynamic model for a single flexible link with a payload is
% implemented. On the actuator side a moter hub with inertia Jh is
% added.
%
% 1. Pinned model - flexpin
% 2. Clamped model -  flexclam

% Reference: Exact modelling of the flexible slewing link
%            F.Bellezza, L.Lanari, G,Ulivi

close all; clear all;
nf=2;

% setting of parameters: nf = # flexible modes
ksi=0.01; rig_b=0.05;
% size link: h=50 mm, b=3 mm, L=1000 mm material: steel
m=4; L=1; E=2.1e11; Iz=1.13e-10; Mp=0; Jp=0; Jh=0.01;
J=Jh+Jp+Mp*L^2+1/3*m*L^3;

syms x ;
beta=0:0.01:5+nf*pi;

% Characteristic equation to obtain values for beta: char=0
% char1: Pinned model and Clamped model - unconstrained
```

```
char1=[]; beta1=[];
for i=1:length(beta),

   char1(i) = abs((Jh*Mp*cos(beta(i)*L)*Jp*cosh(beta(i)*L)/(m^3)-Mp*Jh*Jp...
      /(m^3))*beta(i)^7+      (cos(beta(i)*L)*Jh*Jp*sinh(beta(i)*L)/(m^2)+...
      Jh*sin(beta(i)*L)*Jp*cosh(beta(i)*L)/(m^2))*beta(i)^6+(-1.*Jp*...
      cos(beta(i)*L)*Mp*sinh(beta(i)*L)/(m^2)+Mp*sin(beta(i)*L)*Jh*...
      cosh(beta(i)*L)/(m^2)+Mp*sin(beta(i)*L)*Jp*cosh(beta(i)*L)/(m^2)-...
      Jh*Mp*cos(beta(i)*L)*sinh(beta(i)*L)/(m^2))*beta(i)^4+(-2.*...
      cos(beta(i)*L)*Jp*cosh(beta(i)*L)/m-1.*Jh*cosh(beta(i)*L)*cos(beta(i)*L)...
      /m-Jh/m)*beta(i)^3-2.*Mp*beta(i)*sin(beta(i)*L)*sinh(beta(i)*L)/m-...
      1.*sin(beta(i)*L)*cosh(beta(i)*L)+cos(beta(i)*L)*sinh(beta(i)*L));

   if i>2 & char1(i-2)>char1(i-1) & char1(i)>char1(i-1),
      beta1=[beta1 beta(i-1)];
   end
   if length(beta1)==nf,
      break
   end
end

% Computation of the eigenfunctions: determined upon a scaling factor
% fie_f1: pinned eigenfunctions
for i=1:nf,
   gamma1=-1.*(-2.*Mp*beta1(i)*sin(beta1(i)*L)*m*sinh(beta1(i)*L)+Mp*beta1(i)...
      ^4*sin(beta1(i)*L)*Jp*cosh(beta1(i)*L)+m^2*cos(beta1(i)*L)*...
      sinh(beta1(i)*L)-2.*m*cos(beta1(i)*L)*Jp*beta1(i)^3*cosh(beta1(i)*L)-...
      1.*m^2*cosh(beta1(i)*L)*sin(beta1(i)*L)-1.*Mp*beta1(i)^4*sinh(beta1(i)*L)...
      *Jp*cos(beta1(i)*L))/(-1.*m^2*sin(beta1(i)*L)*sinh(beta1(i)*L)+2.*m*...
      sin(beta1(i)*L)*Jp*beta1(i)^3*cosh(beta1(i)*L)-1.*m^2-2.*Mp*beta1(i)*...
      cos(beta1(i)*L)*m*sinh(beta1(i)*L)+Mp*beta1(i)^4*cos(beta1(i)*L)*Jp*...
      cosh(beta1(i)*L)-1.*m^2*cosh(beta1(i)*L)*cos(beta1(i)*L)+Mp*beta1(i)^4*...
      sinh(beta1(i)*L)*Jp*sin(beta1(i)*L)-1.*Mp*beta1(i)^4*Jp);
   zeta1=(Mp*beta1(i)^4*cos(beta1(i)*L)*Jp*cosh(beta1(i)*L)-1.*Mp*beta1(i)^4*...
      sinh(beta1(i)*L)*Jp*sin(beta1(i)*L)-1.*Mp*beta1(i)^4*Jp+2.*...
      sinh(beta1(i)*L)*m*cos(beta1(i)*L)*Jp*beta1(i)^3+2.*m*sin(beta1(i)*L)*...
      Mp*cosh(beta1(i)*L)*beta1(i)-1.*m^2*cosh(beta1(i)*L)*cos(beta1(i)*L)-1.*...
      m^2+m^2*sin(beta1(i)*L)*sinh(beta1(i)*L))/(-1.*m^2*sin(beta1(i)*L)*...
      sinh(beta1(i)*L)+2.*m*sin(beta1(i)*L)*Jp*beta1(i)^3*cosh(beta1(i)*L)-...
      1.*m^2-2.*Mp*beta1(i)*cos(beta1(i)*L)*m*sinh(beta1(i)*L)+Mp*beta1(i)^4*...
      cos(beta1(i)*L)*Jp*cosh(beta1(i)*L)-1.*m^2*cosh(beta1(i)*L)*...
      cos(beta1(i)*L)+Mp*beta1(i)^4*sinh(beta1(i)*L)*Jp*sin(beta1(i)*L)-...
      1.*Mp*beta1(i)^4*Jp);
```

```
   fie_f1(i)=sin(beta1(i)*x)+gamma1*(cos(beta1(i)*x)-cosh(beta1(i)*x))+...
      zeta1*sinh(beta1(i)*x);
end

% Calculation of the scaling factor from orthogonality relationship
A1=1./sqrt(Jh.*subs(diff(fie_f1,x),x,0).^2+int(m*fie_f1.^2,x,0,L)+Mp.*...
   subs(fie_f1,x,L).^2+...
   Jp.*subs(diff(fie_f1,x),x,L).^2);
fie1=A1.*fie_f1;

% clamped eigenfunctions are computed from the pinned eigenfunctions
fie1c=fie1-x.*subs(diff(fie1,x),x,0);
mhu1=double(int(m*x.*fie1c,x,0,L)+Mp*L*subs(fie1c,x,L)+Jp*...
   subs(diff(fie1c,x),x,L));

% Checking if the eigenfunctions are orthonormal:
% The matrices(ortpin, ortclam) have to be identity matrices
%ortpin=[]; ortclam=[];
%for i=1:nf,
%   for j=1:nf,
%      ortpin(i,j)=double(Jh*subs(diff(fie1(i),x),x,0)*...
% subs(diff(fie1(j),x),x,0)+int(m*fie1(i)...
% *fie1(j),x,0,L)+Mp*subs(fie1(i),x,L)*...
% subs(fie1(j),x,L)+Jp*subs(diff(fie1(i),x),x,L)...
% *subs(diff(fie1(j),x),x,L));
% ortclam(i,j)=double(int(m*fie1c(i)*fie1c(j),x,0,L)+Mp*...
% subs(fie1c(i),x,L)*subs(fie1c(j),x,L)+Jp*...
% subs(diff(fie1c(i),x),x,L)*subs(diff(fie1c(j),x)...
% ,x,L)-mhu1(i)*mhu1(j)/J);
%   end
%end

% Derivation of the dynamic models from the eigenfunctions
% PINNED MODEL, CLAMPED MODEL

% PINNED MODEL
omega=double(sqrt(int(E*Iz*(diff(diff(fie1,x),x)).^2,x,0,L)));

fie_ing=double(subs(fie1,x,L));
fiecc=[];
for i=1:nf,
   fiecc=[fiecc fie_ing(i)/L];
end
```

```
fieF=[];
for i=1:nf,
   fieF=[fieF subs(diff(fie1(i),x),x,0)];
end

Mpin=[J zeros(1,nf); zeros(nf,1) eye(nf)];
Kpin=[0 zeros(1,nf);zeros(nf,1) diag(omega.^2)];
Zpin=[0 zeros(1,nf);zeros(nf,1) diag(2.*ksi.*omega)];
Fpin=[1; fieF.'];

Zpin(1,1)=rig_b;

Apin=[zeros(nf+1) eye(nf+1); -inv(Mpin)*Kpin -inv(Mpin)*Zpin];
Bpin=[zeros(nf+1,1); inv(Mpin)*Fpin];
Cpin=[1 fiecc zeros(1,nf+1)]; % tip angle as output
Dpin=0;

% CLAMPED MODEL - UNCONSTRAINED
kclam=double(E*Iz.*int(diff(diff(fie1c,x),x).^2,x,0,L));

fieM=[];
for i=1:nf,
   for j=1:nf,
      if i==j,
         fieM(i,j)=1+mhu1(i)*mhu1(j)/J;
      else
         fieM(i,j)=mhu1(i)*mhu1(j)/J;
      end
   end
end

fie_ingclam=double(subs(fie1c,x,L));
fieCclam=[];
for i=1:nf,
   fieCclam=[fieCclam fie_ingclam(i)/L]; % the angle as output
end

Mclam=[J mhu1; mhu1.' fieM];
Kclam=[0 zeros(1,nf);zeros(nf,1) diag(kclam)];
Zclam=[0 zeros(1,nf);zeros(nf,1) diag(2.*ksi.*sqrt(kclam))];
Fclam=[1; zeros(nf,1)];

Zclam(1,1)=rig_b;
```

```
Aclam=[zeros(nf+1) eye(nf+1); -inv(Mclam)*Kclam -inv(Mclam)*Zclam ];
Bclam=[zeros(nf+1,1); inv(Mclam)*Fclam];
Cclam=[1 fieCclam zeros(1,nf+1)];
Dclam=0;

% Three State space models of the flexible link
flexpin=ss(Apin,Bpin,Cpin,Dpin);
flexclam=ss(Aclam,Bclam,Cclam,Dclam);

% Plotting of essential system information
w=logspace(-1,3,5000);
figure(1); grid on;
bode(flexpin); hold on;
bode(flexclam,'g'); hold on;
title('Bode diagram (flexpin): input=torque, output=tip angle')
legend('flexpin','flexclam');
figure(2)
subplot(211);
pzmap(flexpin); title('Pole-Zero map : flexpin');
subplot(212);
pzmap(flexclam); title('Pole-Zero map : flexclam'); hold on;
```

C.2 Stable inverse control law

```
% FFORWARD calculates a feedforward control law for a nonminimum
%  phase system. The dynamic model (ss) is loaded via the matrices
%  A,B,C,D. A reference trajectory is specified using reftraject.m.
%  A transformation matrix to the normal form is used to devide the
%  internal dynamics into a stable and unstable part. Bounded
%  solutions to the zerodynamics are obtained by evaluating the
%  integrals defined in sig_s.m and sig_u.m
%
% Reference : Preview-Based Stable-Inversion for Output Tracking
% authors   : Q.Zou and S.Devasia  [1999 AACC]

global t2 dtau Tp a b yref Asig_s Asig_u Bsig1_s Bsig2_s Bsig3_s
Bsig1_u Bsig2_u Bsig3_u;

close all;

% parameters : user defined
```

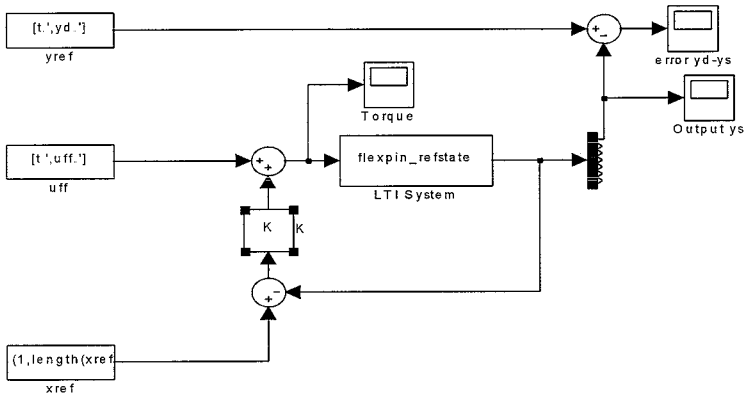
```
dtau=0.01; Tp=1; a=5; b=0.5; yref=pi/2;
% parameters : model dependent (nd = # states, ndz = dim(zerodynamica))
r=2; nd=2+2*nf; ndz=nd-r;
%A=Apin; B=Bpin; C=Cpin; D=Dpin;
A=Aclam; B=Bclam; C=Cclam; D=Dclam;
B1=B(1:nf,1); CT=C(1:1+nf);
flexpin=ss(Apin,Bpin,Cpin,Dpin);

Tend=4*b+4*pi/a+2*Tp;
tau=-Tp:dtau:Tend;
t=tau;

% Computation of the reference trajectories
[yd,ydd,yddd]=reftraject(Tp,dtau,tau,a,b,yref);
Yd=[yddd.' yd.' ydd.'];
ksid=[yd.' ydd.'];

Ax=C*A^r;
By=C*A^(r-1)*B; invBy=inv(By);

% Coordinate transformation
tadd=[];
for i=1:nf,
   tadd(i)=-B(2+nf+i)/B(2+nf);
end

Tadd=[tadd.' eye(nf)];
Tin=[eye(nf)  zeros(nf,nf+1); zeros(nf) Tadd];
T=[C ; C*A;
   zeros(2*nf,1) Tin ];
invT=inv(T);

Adak=T*A*inv(T);
A1dak=Adak(1:r,1:r);
A2dak=Adak(1:r,r+1:nd);
A3dak=Adak(r+1:nd,1:r);
A4dak=Adak(r+1:nd,r+1:nd);
Bdak=T*B;
B1dak=Bdak(1:r);
B2dak=Bdak(r+1:nd);

Aksieta=Ax*inv(T);
Aksi=Aksieta(1:r);
Aeta=Aksieta(r+1:nd);
```

```
Adaketa= A4dak-B2dak*invBy*Aeta;
Bdaketa=[B2dak*invBy A3dak-B2dak*invBy*Aksi];

% Decoupling of the zero-dynamics into a stable and unstable part

[Uus,labda]=eig(Adaketa);
[labda_s,ind]=sort(real(diag(labda)));
count=0;
for i=1:length(labda),
   if sign(labda_s(i))==-1,
      count=count+1;
   end
end

Up=[];
for i=1:length(labda),
   Up(:,i)=Uus(:,ind(i));
end
U=inv(Up);

Adaksu=diag(U*Adaketa*inv(U));
Adaks=Adaksu(1:count);
Adaku=Adaksu(count+1:ndz);
Bdaksu=U*Bdaketa;
Bdaks=Bdaksu(1:count,:);
Bdaku=Bdaksu(count+1:ndz,:);

invU=inv(U);
Udaks=invU(:,1:count);
Udaku=invU(:,count+1:ndz);

% Calculation of bounded solutions to the zerodynamics:
% Stable part of the zerodynamics: sigma_s
% Untable part of the zerodynamics: sigma_u

ns=length(Adaks); nu=length(Adaku);

sigma_s=[]; sigma_u=[];
if ns==0,
   sigma_s=zeros(length(tau),1);
   Udaks=zeros(ndz,1);
end
if nu==0,
```

```
   sigma_u=zeros(length(tau),1);
   Udaku=zeros(ndz,1);
end

for j=1:ns
   Asig_s=Adaks(j); Bsig1_s=Bdaks(j,1); Bsig2_s=Bdaks(j,2);
   Bsig3_s=Bdaks(j,3);
   for i=1:length(tau),
    t2=tau(i);
    [length(tau),i,ns,j]
    sigma_s(i,j)=quad('sig_s',-Tp,t2);
end
end
for j=1:nu
   Asig_u=Adaku(j); Bsig1_u=Bdaku(j,1); Bsig2_u=Bdaku(j,2);
   Bsig3_u=Bdaku(j,3);
   for i=1:length(tau),
    t2=tau(i);
    [length(tau),i,nu,j]
      sigma_u(i,j)=-quad('sig_u',t2,t2+Tp);
   end
end

% Calculation of the feedforward input (uff) and the reference
% state trajectories (xref)
uff=[];
for i=1:length(tau),
   uff(i)=invBy*[yddd(i)-Aksi*ksid(i,:).'-Aeta*...
         Udaks*sigma_s(i,:).'- Aeta*Udaku*sigma_u(i,:).'];
   xref(:,i)=invT*[yd(i) ydd(i) [invU*[sigma_s(i,:) ...
               sigma_u(i,:)].'].'].';
end

[ys,ts,xs]=lsim(flexpin,uff,tau);
ts=ts.'; ys=ys.';

% Plotting : output of the control part
figure(7)
subplot(121)
plot(tau,uff); grid on;
title('feedforward input (uff)');
xlabel('time [s]'); ylabel('Motor torque [Nm]');
%figure(6)
subplot(122)
```

```
plot(tau,yd); hold on;%grid on;
plot(ts,ys,'r');
legend('yd','ys')
title('Desired output (yd) and output under uff (ys)');
xlabel('time [s]'); ylabel('yd and ys [rad]');
subplot(212)
plot(tau,yd-ys); grid on;
title('error(yd-ys)')
xlabel('time [s]'); ylabel('error(yd-ys) [rad]');

function y=sig_s(tau)

global t2 dtau Tp a b yref Asig_s Bsig1_s Bsig2_s Bsig3_s;

[yd,ydd,yddd]=reftraject(Tp,dtau,tau,a,b,yref);

y=exp(Asig_s.*(t2-tau)).*(Bsig3_s.*ydd);


function y=sig_u(tau)

global t2 dtau Tp a b yref Asig_u Bsig1_u Bsig2_u Bsig3_u;

[yd,ydd,yddd]=reftraject(Tp,dtau,tau,a,b,yref);

y=exp(-Asig_u.*(tau-t2)).*Bsig3_u.*ydd);
```

C.2.1 Simulink model



Simulink model for the stable inverse control law with state feedback

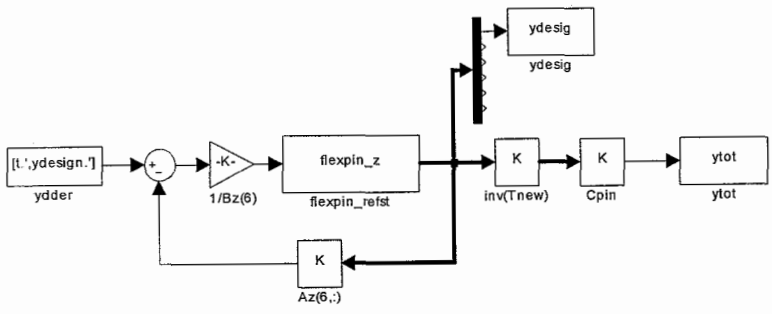
C.3 State-to-state controller

```
% State to state transfer

close all;

% 1. Determination of an appropriate output function which gives
%    the system a relative degree r=ntot
%    Proposed output function: y=[1 c1 c2 0 0 0]*x

rel=[]; rel2=[];
for i=1:nf,
   rel(:,i)=Apin^(2*i-1)*Bpin;
   rel2(:,i)=rel(2:nf+1,i);
end

c=rel2.'\(-rel(1,:).');

Cnew=[1 c.' zeros(1,1+nf)];
Cstat=[Cnew; eye(2+2*nf)];

Tnew=[];
for i=1:2+2*nf,
   Tnew(i,:)=Cnew*Apin^(i-1);
end
```

```
% 2. Design output
%    This design output is used as the new input
% computation of ydesign and interpolating trajectory in MAPLE
dt=0.001; Tf=1;
t=0:dt:Tf;

ydesign=[]; interpol=[];
for i=1:length(t),
   if Tf==0.5, %Tf=[0.5 1 1.5]->ydesign^[3+4*nf]+interp. pol.
      ydesign(i)=.1469033067e14*t(i)^7-.2570807869e14*t(i)^6+...
         .1752823547e14*t(i)^5-.5842745155e13*t(i)^4+.9737908600e12*...
         t(i)^3-.7303431444e11*t(i)^2+1738912249.*t(i);
      interpol(i)=11889998.28*t(i)^13-38642494.42*t(i)^12+52694310.57...
         *t(i)^11-38642494.42*t(i)^10+16101039.34*t(i)^9-3622733.852*...
         t(i)^8+345022.2716*t(i)^7;
   elseif Tf==1,
      ydesign(i)=570810240*pi*t(i)^7-1997835840*pi*t(i)^6+2724321600*...
         pi*t(i)^5-1816214400*pi*t(i)^4+605404800*pi*t(i)^3-90810720*...
         pi*t(i)^2+4324320*pi*t(i);
      interpol(i)=462*pi*t(i)^13-3003*pi*t(i)^12+8190*pi*t(i)^11-12012*...
         pi*t(i)^10+10010*pi*t(i)^9-9009/2*pi*t(i)^8+858*pi*t(i)^7;
   elseif Tf==1.5,
      ydesign(i)=9212963.672*t(i)^7-48368158.00*t(i)^6+98935102.92*...
         t(i)^5-98935377.85*t(i)^4+49467852.32*t(i)^3-11130311.20*...
         t(i)^2+795026.1300*t(i);
      interpol(i)=7.456749929*t(i)^13-72.70346020*t(i)^12+297.4239506*...
         t(i)^11-654.3345095*t(i)^10+817.9208392*t(i)^9-552.0987699*...
         t(i)^8+157.7432797*t(i)^7;
   end
end

% 3. torque
part1=Cnew*Apin^(2+2*nf);
part2=1/(Cnew*Apin^(1+2*nf)*Bpin);

% 4. System in old and new coordinates
flexpin_refst=ss(Apin,Bpin,Cstat,Dpin);
Az=Tnew*Apin*inv(Tnew);
Bz=Tnew*Bpin;
Cz=[eye(2+2*nf)];
flexpin_z=ss(Az,Bz,Cz,Dpin);

%%% Running of Simulink file %%%
```

```
% Output
figure(1)
subplot(121)
plot(tout,ydesig,tout,ytot)
title('Design output (y_d) and modeloutput (y_m)) for T_f=1 s');
xlabel('time [s]'); ylabel('angle [rad]')
legend('y_d','y_m')
subplot(122)
plot(tout,ydesig-ytot)
title('Difference y_d-y_m' );
xlabel('time [s]'); ylabel('angle [rad]')
```

C.3.1 Simulink model



Simulink model for the state-to-state control law

Appendix D

Maple files

D.1 Zerodynamics

D.1.1 Pseudo-pinned model

```
zerospinned:=proc(n)
global i,j,J,phi,omega,invMB,CA,CA2,zerdyn,invMK,
Anf,w,L,A,C,M,K,F,Jh,m,
nulp,T,B;

with(linalg);

# SETTING OF CONSTANTS (OPTIONAL)
#Jh:=1; m:=1; L:=1;
#J:=Jh+1/3*m*L^3;

# START CALCULATION
M:=array(1..n,1..n,sparse);
for i from 1 to n do
  for j from 1 to n do
     if i=j and i=1 then M[i,j]:=J;
     elif i=j then M[i,j]:=1;
     else M[i,j]:=0;
     fi;
  od;
od;

K:=array(1..n,1..n,sparse);
for i from 1 to n do
```

```
  for j from 1 to n do;
     if i=j and i>1 then K[i,j]:=omega[i-1]^2;
     else K[i,j]:=0;
     fi;
  od;
od;

# zeta=subs(x=0,diff(phi,x));
F:=array(1..n,sparse);
for i from 1 to n do
     if i=1 then F[i]:=1;
     else F[i]:=zeta[i-1];
     fi;
od;

# pi=subs(x=x,phi)/x
C:=array(1..2*n);
for i from 1 to 2*n do
    if i=1 then C[i]:=1;
    elif i<n+1 then
    C[i]:=pi[i-1];
    else C[i]:=0;
    fi;
od;

invMK:=multiply(inverse(M),K);

A:=array(1..2*n,1..2*n);
for i from 1 to 2*n do
 for j from 1 to 2*n do
    if (i<n+1 and j<n+1) or (i>n and j>n) then
    A[i,j]:=0;
    elif j>n and i+n=j then
    A[i,j]:=1;
    elif j>n and i<n+1 then A[i,j]:=0;
    elif j<n+1 and i>n then
    A[i,j]:=-invMK[i-n,j];
    fi;
 od;
od;

invMB:=multiply(inverse(M),F);

B:=array(1..2*n);
```

```
for i from 1 to 2*n do
   if i<n+1 then
   B[i]:=0;
   else B[i]:=invMB[i-n];
   fi;
od;

CA:=multiply(C,A);

T:=array(1..2*n,1..2*n);
for i from 1 to 2*n do
   for j from 1 to 2*n do
    if i=1 then
    T[i,j]:=C[j];
    elif i=2 then
    T[i,j]:=CA[j];
    elif (i=j+1 and i<n+2) or (i=j and i>n+1) then
    T[i,j]:=1;
    elif j=n+1 and i<>n+1 then
    T[i,j]:=-B[i]/B[n+1];
    else
    T[i,j]:=0
    fi;
  od;
od;


Anf:=multiply(multiply(T,A),inverse(T));

zerdyn:=array(1..2*n-2,1..2*n-2);
for i from 1 to 2*n-2 do
  for j from 1 to 2*n-2 do
  zerdyn[i,j]:=Anf[i+2,j+2];
  od;
od;
nulp:=eigenvals(zerdyn,implicit);


lprint('The number of flexible modes included:'); print(n-1);
lprint('zeros of the flexible link:'); print(nulp);

end:
```

D.1.2 Pseudo-clamped model

```
zerosclamped:=proc(n)
local i,j,J,phi,omega,invMB,CA,CA2,zerdyn,invMK,
Anf,w,L,nulp,A,B,C,T,M,K,F;

with(linalg);

# SETTING OF CONSTANTS (OPTIONAL)
#Jh:=1; m:=1; L:=1;
#w:=1; J:=Jh+1/3*m*L^3;

# COMPUTATION
M:=array(1..n,1..n,sparse);
for i from 1 to n do
  for j from 1 to n do
     if i=j and i=1 then M[i,j]:=J;
     elif i=j then M[i,j]:=1+mu[i-1]^2/J;
     elif i=1 then
     M[i,j]:=mu[j-1];
     elif j=1 then
     M[i,j]:=mu[i-1];
     else M[i,j]:=0;
     fi;
  od;
od;

K:=array(1..n,1..n,sparse);
for i from 1 to n do
  for j from 1 to n do;
     if i=j and i>1 then K[i,j]:=omega[i-1]^2;
     else K[i,j]:=0;
     fi;
  od;
od;

F:=array(1..n,sparse);
for i from 1 to n do
    if i=1 then F[i]:=1;
    else F[i]:=0;
    fi;
```

```
od;

# pi=subs(x=L,phi)
C:=array(1..2*n);
for i from 1 to 2*n do
   if i=1 then C[i]:=1;
   elif i<n+1 then
   C[i]:=pi[i-1];
   else C[i]:=0;
   fi;
od;

invMK:=multiply(inverse(M),K);

A:=array(1..2*n,1..2*n);
for i from 1 to 2*n do
 for j from 1 to 2*n do
   if (i<n+1 and j<n+1) or (i>n and j>n) then
   A[i,j]:=0;
   elif j>n and i+n=j then
   A[i,j]:=1;
   elif j>n and i<n+1 then A[i,j]:=0;
   elif j<n+1 and i>n then
   A[i,j]:=-invMK[i-n,j];
   fi;
 od;
od;

invMB:=multiply(inverse(M),F);

B:=array(1..2*n);
for i from 1 to 2*n do
   if i<n+1 then
   B[i]:=0;
   else B[i]:=invMB[i-n];
   fi;
od;

CA:=multiply(C,A);

T:=array(1..2*n,1..2*n);
for i from 1 to 2*n do
   for j from 1 to 2*n do
    if i=1 then
```

```
    T[i,j]:=C[j];
    elif i=2 then
    T[i,j]:=CA[j];
    elif (i=j+1 and i<n+2) or (i=j and i>n+1) then
    T[i,j]:=1;
    elif j=n+1 and i<>n+1 then
    T[i,j]:=-B[i]/B[n+1];
    else
    T[i,j]:=0
    fi;
  od;
od;

Anf:=multiply(multiply(T,A),inverse(T));

zerdyn:=array(1..2*n-2,1..2*n-2);
for i from 1 to 2*n-2 do
  for j from 1 to 2*n-2 do
  zerdyn[i,j]:=Anf[i+2,j+2];
  od;
od;
zeros:=eigenvals(zerdyn,implicit);
lprint('The number of flexible modes included:'); print(n-1);
lprint('zeros of the flexible link:'); print(zeros);


end:
```

D.2 Interpolating trajectory

```
interpoltraj:=proc(nf)

local poly,t,a,Tf,i,polyset,diffpoly,opl,Tb,ydder;

# Begin and final time
Tb:=0;
Tf:=1.5;

poly:=0;
```

```
for i from 1 to 4*nf+6 do
poly:=poly+a[i]*t^(4*nf+6-i);
end;
print(poly);

diffpoly:=array(1..4*nf+6,sparse);
diffpoly[1]:=subs(t=Tb,poly)=0;
diffpoly[2]:=subs(t=Tf,poly)=Pi/2;

for i from 1 to 2*nf+2 do
diffpoly[2+i]:=subs(t=Tb,diff(poly,t$i))=0;
end;

for i from 1 to 2*nf+2 do
diffpoly[2+2*nf+2+i]:=subs(t=Tf,diff(poly,t$i))=0;
end;

polyset:=convert(diffpoly,set);
opl:=solve(polyset);

for i from 1 to 4*nf+6 do
poly:=subs(opl[i],poly);
end;
print(plot(poly,t=Tb..Tf));

lprint('The interpolating polynomial: '); print('poly');
ydder:=diff(poly,t$(2*nf+2));
print(plot(ydder,t=Tb..Tf));
lprint('the (2*nf+2)th derivative of the interpolating polynom: ');
print(ydder);

end:
```