Eindhoven University of Technology

MASTER

Robust control based on mu-synthesis applications for dynamic systems

Heikoop, R.

Award date:
1995

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Robust control based on μ-synthesis
Applications for dynamic systems

Ruud Heikoop

WFW report 92.128

Coach: Ir. A.G. de Jager
Professor: Dr. ir. J.J. Kok

November, 1992

Eindhoven University of Technology
Department of Mechanical Engineering
Division of Mechanical Engineering Fundamentals

Summary

Dynamic systems often have to follow a desired trajectory with a certain accuracy. There are many controls methods available to do this. Some of these methods are adaptive control and robust control. A disadvantage of the adaptive controllers is that stability cannot be always proved if the model, which describes the system, contains model errors. However, as a consequence of un-modelled dynamics, measurement noise and parameter variations the model will always contain model errors. The important property of a robust controller is that it is less sensitive to model errors.

In this report a robust controller based on the singular value (μ), which was introduced in 1982 by Doyle [6], has been presented. The controller design is based on a linear model. The structure of the model errors can be taken into account during this design and the controller should also perform in the presence of model errors. In this study attention has been paid to the determination of the model errors as a consequence of unmodelled dynamics and parameter variations. The research has been carried out by simulations and experiments. A rigid system, an RT-robot, and a flexible system, an xy-table, have been simulated and the controller has been implemented in the xy-table for experiments. The results of the μ-controller have been compared to the results of an H_∞-controller, also designed with robustness specifications, and a PD-controller.

Simulations of the RT-robot have shown that the μ-controller is more robust with respect to parameter variations than the PD- and H_∞-controllers. In the case of the unmodelled dynamics the μ-controller is not more robust with respect to these dynamics than the H_∞- and PD-controller. Simulations of the xy-table only show a better tracking accuracy for the μ-controller, but no increase of the robustness. During the experiments with the xy-table the μ-controller does not perform better than the PD- and H_∞-controllers.

Summarizing, it can be concluded that for the systems used in this research the μ-controllers doesn't perform satisfactory, which implies that further research into μ-control will be necessary. Especially, attention has to be paid to the determination of the structure of the model errors.

Contents

Notation

μ	structured singular value
$\|\cdot\|$	absolute value
$\|\|\cdot\|\|_\infty$	infinity norm
$\mathbf{R}^{nxm}$	the set of nxm real matrices
$\mathbf{C}^{nxm}$	the set of nxm complex matrices
A, A_{ij}, A_i	matrices, A_i sometimes a scalar
A^T	transpose of A
A^{-1}	inverse of A
A^*	complex conjugate transpose of A
$\tilde{A}$	perturbed matrix
a	column
a_i	i'th term of a or scalar
$\hat{a}$	estimate
a_d	desired, tracking error $a-a_d$
$\dot{a}$	first order time derivative
$\ddot{a}$	second order time derivative
$\underline{\sigma}(A)$	minimum singular value of A
$\bar{\sigma}(A)$	maximum singular value of A
$\mu_{\Delta..}(A)$	structured singular value of A with respect to structure $\Delta..$
$\Delta..$	set of .. bounded perturbations
$B\Delta..$	set of .. normalized perturbations
Δ_b	set of mixed complex, real, block-diagonal perturbations
$B\Delta_b$	set of normalized mixed complex, real, block-diagonal perturbations
Δ_p	performance block

Some important matrices and functions

G	general system of the interconnection structure
K	controller
Δ	perturbation matrix
$W_1(s)$	weighting function which reflects the requirements
$W_2(s)$	weighting function which reflects the uncertainties (model errors)
S(s)	sensitivity function
T(s)	complementary sensitivity function

Chapter 1. Introduction

In this report some controllers for dynamic systems will be considered and compared. Often the system's end-effector has to track a desired trajectory with a tracking error which is smaller than the maximum permitted tracking error. Generally, a controller design is based on a model of the dynamic system or the model is a part of the controller. If the model corresponds to the system a controller can be designed such that the tracking error will be zero. However, in practice, the model will never exactly correspond to the system. There will always be model errors like:

1. Uncertainties in the model parameters, e.g. not exactly known masses, spring constants, damper constants.
2. Model structure uncertainties or higher order dynamics. For example a system is often considered stiff while there could be flexibilities which have influence on the system behaviour. Another example is unmodelled motor dynamics.
3. Measurement noise, which is often not modelled, but which could also cause tracking errors.

There are many methods to control dynamic systems. In some cases adaptive controllers perform well (small tracking errors), but stability cannot always be proved if the model contains model errors. In this study robust controllers will be used to control the system. Robust means that the controller is less sensitive to model errors caused by for instance parameter variations and unmodelled dynamics. The controlled system is not only required to be robust to model errors, but it also has to perform well. In other words robust-performance for the controller is required.
For linear models several robust controller design methods are available. One of them is the H_∞-design method. This method takes the model errors into account during the controller design. The only restriction for these model errors is that they are norm bounded. A disadvantage of this design method is that in multivariable design the obtained condition for the robust performance can be arbitrary conservative, so that the resulting design is far from satisfactory.
Usually, the model errors have some structure. The H_∞-control design method does not take into account this structure. For this reason the notion of Structured Singular Values (μ) has been introduced [6]. The μ-synthesis is a control design method based on this notion and takes the structure of the model errors into account. For this reason the controller design can be less conservative than an H_∞-controller design. The performance, robust stability and robust performance can be written in μ-terms (chapter 2). The μ-synthesis design method has some problems:

1. How to compute the μ-values? These values cannot always be exactly computed. In appendix A a summary of the μ-computation is given.
2. The μ-synthesis design method and also the H_∞ design method require a linear model description of the system. This means that a nonlinear model of the system has to be linearized. In this research the nonlinear model will be linearized by a state feedback. As a consequence of model errors this feedback linearization, if it exists, is usually not exact, so that the obtained system still has to be linearized along a nominal trajectory. Because the nonlinear model contains model errors, the linear model will also contain model errors. More about the feedback linearization in chapter 3 and appendix B. A method how to quantify these model errors will also be considered in chapter 3. If a linear model exists it has to be put into a general framework, which is used for the analysis and synthesis. This framework is described in chapter 2.
3. It is a problem to determine the model errors. Comparing the model used for the controller design to a more advanced model can be a way to determine these errors. Another way is identification of the model errors. For example information of (former) experiments can be used to estimate the model errors. An analysis in the frequency domain can also be very useful. However, the designer has to decide based on his knowledge of the system how to determine the model errors. The determination of the model errors is obviously an important

part of the controller design. When the model errors have been obtained they also have to be put in the general framework.

In chapter 3 a description of an RT-robot is given. For this robot a μ-controller, an H_∞-controller and a PD-controller will be designed and compared with each other. Simulations will be carried out for two types of perturbations (model errors). First, the motor dynamics (unmodelled higher order dynamics) and secondly, variations in the end effector mass are considered as uncertainty.
In chapter 4 a PD-, an H_∞- and a μ-controller will be designed for an xy-table, which is a flexible manipulator. For the controller design a simple linear model, without flexibilities will be used. The flexibility is considered as a model error. The designed controllers are compared with each other by means of simulations.
In chapter 5 the controllers designed in the previous chapter are implemented in the actual xy-table. The flexibility of the xy-table can be changed by varying the stiffness of a component of the system. Experiments will be carried out for several values of the stiffness. The controllers are compared with each other. The experimental results will also be compared to the simulation results.
In chapter 6 the conclusions based on the experimental and the simulation results are presented. Recommendations for further research into the μ-control are also given.

Chapter 2. Theory of μ-synthesis

2.1 Introduction

In this chapter, first a general interconnection structure usually used for the μ-synthesis and the Linear Fractional Transformation will be introduced. Further, the μ-analysis and synthesis are considered. The conditions for the performance, robust stability and robust performance presented in the analysis part will be compared to the expressions obtained for the H_∞-analysis of a SISO system. This is done to better comprehend the complex expressions. For the synthesis part, the controller design, a currently used method, will be considered. This method is the D-K iteration. In section 2.3 two types of perturbations will be considered. These perturbations can represent the model errors caused by unmodelled dynamics or parameter variations.

2.2 General interconnection structure and Linear Fractional Transformation

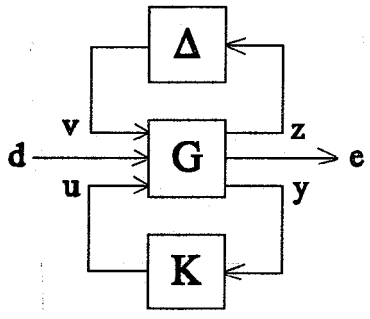


Figure 2.1 The general interconnection structure.

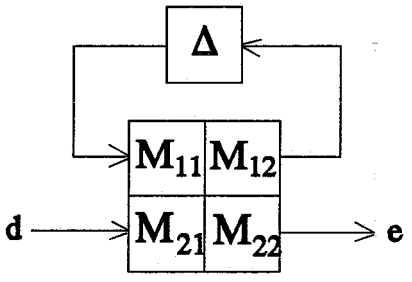


Figure 2.2 Linear Fractional Transformation description

As already stated in the introduction a general framework is used for μ-analysis and μ-synthesis. This framework is illustrated in figure 2.1. Any linear interconnection of inputs, outputs, commands, perturbations and a controller can be rearranged to match this diagram. The general system G represents the model of the system to be controlled and also contains the weighting functions which reflect how the uncertainty (model errors) affects the system. K represents the controller. Δ represents the perturbations as a consequence of the model errors caused by unmodelled dynamics and parameter variations. The inputs are v, feeding back the perturbations into the system, d, consisting of system inputs and disturbances and u, the controller outputs. The three outputs are z, the input to the perturbation matrix Δ, e, the output signals of interest and y, the controller inputs.
The μ-analysis is a method for analyzing the performance and robustness properties of feedback systems. For the robustness analysis the controller K can be incorporated in G, because in that case the controller is known. The diagram of figure 2.1 now reduces to that in figure 2.2, with $M = F_l(G,K)$. $F_l(G,K)$ is called a Linear Fractional Transformation on G by K [3, 8]. The subscript l on F_l pertains to the "lower" loop of G which is closed by K. The transfer function from the input d to the output e can be also expressed as an LFT:

$$e = F_u(M,\Delta)d = [M_{22}+M_{21}\Delta(I-M_{11}\Delta)^{-1}M_{12}]d \qquad (2.1)$$

The subscript u on F_u now pertains to the "upper" loop of M which is closed by Δ. For the robust performance analysis the block diagram of figure 2.2 has to be extended with the performance specifications. This is described in section 2.4. In the next section first, a short review of the perturbations is given.

2.3 Types of perturbations

In this report two types of perturbations are used to describe the model errors

a. Real valued perturbations. The scalar real valued norm bounded perturbation Δ_r is a set of real numbers. The only constraint on the elements of the set is that their absolute value is smaller than or equal to a real number δ. These perturbations can be used for parametric uncertainty. The set of the perturbations is defined as

$$\Delta_r = \{\Delta | \Delta \in [-\delta, +\delta], \delta \in \mathbb{R}^+\}$$
$$\text{normalized } \mathbf{B}\Delta_r = \{\Delta | \Delta \in [-1, +1]\}$$

b. Complex valued perturbations. The pxq complex valued norm bounded perturbation Δ_c is a set of pxq frequency dependent complex matrices. The only constraint on the matrices is that they have a frequency dependent bound δ_c on their maximum singular value ($\bar{\sigma}(\cdot)$). These perturbations can be used for unmodelled dynamics, e.g. motor dynamics, higher order dynamics. The set of complex perturbations is defined as

$$\Delta_c = \{\Delta | \bar{\sigma}(\Delta(\omega)) \le \delta_c(\omega), \Delta(\omega) \in \mathbb{C}^{pxq}, \delta_c(\omega) \in \mathbb{R}^+, \forall\ \omega\}$$
$$\text{normalized } \mathbf{B}\Delta_c = \{\Delta | \Delta \in \Delta_c, \bar{\sigma}(\Delta) \le 1\}$$

The perturbations can occur in a variety of structures. Usually the structure is block diagonal, because of the following reasons (refer to [11]):

1. Connections of LFT's always lead to a new LFT with a block-diagonal structure.
2. An LFT which describes the uncertainty in one subsystem may be (block-)diagonal itself.

The elements of the block diagonal perturbations can appear in special forms. In this report the elements are always member of $\mathbf{B}\Delta_r$ or $\mathbf{B}\Delta_c$. Some interesting block forms, which are used are represented.

Real repeated scalar blocks

$$\Delta_{rr} = \{\Delta | \Delta = \delta I_n, \delta \in \mathbb{R}\}$$
$$\mathbf{B}\Delta_{rr} = \{\Delta | \Delta \in \Delta_{rr}, \delta \in [-1, +1]\}$$

Real non-repeated scalar blocks

$$\Delta_{rn} = \{\Delta | \Delta = \text{diag}\{\delta_1,\delta_2,...,\delta_n\}, \delta_1,\delta_2,...,\delta_n \in \mathbb{R}\}$$
$$\mathbf{B}\Delta_{rn} = \{\Delta | \Delta \in \Delta_{rn}, \delta_1,\delta_2,...,\delta_n \in [-1, +1]\}$$

Complex non-repeated scalar blocks

$$\Delta_{cn} = \{\Delta | \Delta = \text{diag}\{\delta_1,...,\delta_{n1}, \Delta_1,...,\Delta_{n2}\}, \delta_1,...,\delta_{n1} \in \mathbb{C}, \Delta_1,...,\Delta_{n2} \in \mathbb{C}^{kxk}\}$$
$$\mathbf{B}\Delta_{cn} = \{\Delta | \Delta \in \Delta_{cn}, |\delta_1|,...,|\delta_{n1}| \leq 1, \bar{\sigma}(\Delta_1),...,\bar{\sigma}(\Delta_{n2}) \leq 1\}$$

Full complex blocks

$$\Delta_f = \{\Delta | \bar{\sigma}(\Delta(\omega)) \leq \delta_c(\omega), \Delta(\omega) \in \mathbb{C}^{nxn}, \delta_c(\omega) \in \mathbb{R}^+, \forall\ \omega\}$$
$$\mathbf{B}\Delta_f = \{\Delta | \Delta \in \Delta_f, \bar{\sigma}(\Delta) \leq 1\}$$

General block-diagonal perturbation

The combination of the block structures above leads to the definition of a general block-diagonal perturbation Δ_b [3, 6].
Two integers S and F represent the number of repeated scalar blocks and the number of full blocks, respectively. To bookkeep their dimensions the following positive integers are introduced $r_1,...,r_S$; $m_1,...,m_F$. The i'th repeated scalar block is $r_i x r_i$, while the j'th full block is $m_j x m_j$.
The associated block-diagonal perturbation $\Delta_b \subset \mathbb{C}^{nxn}$ is given by

$$\Delta_b = \{\Delta | \Delta = \text{diag}\{\delta_1 I_{r_1},...,\delta_S I_{r_S}, \Delta_1,...,\Delta_F\} : \delta_i \in \mathbb{C}, \Delta_j \in \mathbb{C}^{m_j x m_j}\} \quad (2.2)$$
$$\mathbf{B}\Delta_b = \{\Delta | \Delta \in \Delta : \bar{\sigma}(\Delta) \leq 1\}$$

This block-diagonal perturbation is also used in appendix A for the computation of the structured singular value.

2.4 μ-Analysis

Robust Stability

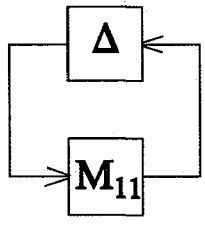


Figure 2.5 An interconnection structure for robust stability.

For robust stability analysis only the blockdiagram of figure 2.5 has to be considered.
Define RH_∞ as the set of all real stable proper scalar transfer functions and define RH_∞^{nxn} as the set of all nxn matrices with entries in RH_∞. There are two restrictions on the framework within which μ-analysis can be applied:

1. $M_{11}(s) \in RH_\infty^{nxn}$

2. The $k_q x k_q$ block-diagonal elements of Δ_b are subsets of $RH_\infty^{k_q x k_q}$

The first demand requires that the controlled system under consideration is nominally stable. The second demand is satisfied by requiring that Δ only has norm bounded perturbations as its elements ($\Delta \in \mathbf{B}\Delta_b$). The definition of μ is based on the small gain theorem.

Small gain theorem [11]:

Let $M_{11}(s) \in RH_\infty^{nxn}$ and $\Delta \in \mathbf{B}\Delta_f$
The feedback structure in figure 2.5 is internally stable iff
$\det(I-\Delta M_{11}(j\omega)) \neq 0 \quad \forall\, \Delta \in \mathbf{B}\Delta_f, \omega \in (-\infty, \infty)$ iff
$\sup \{\bar{\sigma}(M_{11}(j\omega))\} < 1, \omega \in (-\infty, \infty)$ (2.3)

In another notation this means $||M_{11}(j\omega)||_\infty < 1$.
This condition is necessary and sufficient for robust stability for unstructured perturbations ($\Delta \in \mathbf{B}\Delta_f$). However, if the perturbation is structured ($\Delta \in \mathbf{B}\Delta_b$), then the condition is only sufficient, since $\mathbf{B}\Delta_f \subset \mathbf{B}\Delta_b$. For these types of perturbations the μ robust stability theorem based on the structured singular value μ can be used. The definition of the structured singular value is given in appendix A. The theorem is as follows [11]:

μ Robust stability theorem

Let $M_{11}(s) \in RH_\infty^{nxn}$ and $\Delta \in \mathbf{B}\Delta_b$
the feedback structure in figure 2.5 is internally stable iff
$\det(I-\Delta M_{11}(j\omega)) \neq 0\ \forall\, \Delta \in \mathbf{B}\Delta_b, \omega \in (-\infty, \infty)$ iff
$\sup \{\mu_{\Delta b}(M_{11}(j\omega))\} < 1, \omega \in (-\infty, \infty)$ (2.4)

Or in a simple notation $||M_{11}(j\omega)||_\mu < 1$ for $\Delta \in \mathbf{B}\Delta_b$.

The singular value in the small gain theorem can be computed easily and exact. The structured singular value (μ) cannot be always computed exact. This value has often to be approximated, see appendix A.
In the case of a SISO system a similar condition for robust stability exists. A SISO system is robustly stable for $\Delta(j\omega)$, a multiplicative uncertainty with $||\Delta(j\omega)||_\infty < 1$, if and only if $||M_{11}||_\infty = ||W_2(j\omega)T(j\omega)||_\infty < 1$. Where $T(j\omega)$ is the complementary sensitivity function, the response of an output signal to a reference signal. In figure 2.4 the complementary sensitivity function is defined as the transfer function from r to y. $W_2(j\omega)$ is a weighting function which represents the uncertainty profile of the system. The multiplicative uncertainty is defined as follows

$$\tilde{P}(j\omega) = P(j\omega)(1 + \Delta(j\omega)W(j\omega)) \tag{2.5}$$

with $P(j\omega)$ = The nominal plant model
$\tilde{P}(j\omega)$ = Set of possible plants
$W_2(j\omega)$ = Weighting function which represent the uncertainty profile of the plant
$\Delta(j\omega)$ = Scaling factor on the magnitude of the perturbation (varies between 0 and 1)

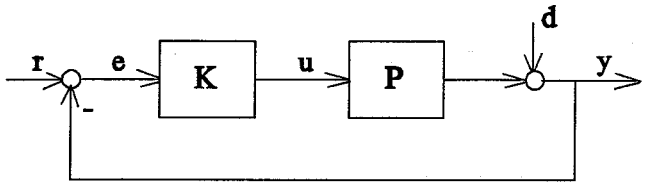


Figure 2.4 Single loop feedback system

Robust performance

Robust stability of a control system is the first requirement that has to be satisfied. However the system's performance specifications have to be also fulfilled. These specifications could be for example small tracking errors or limited control efforts. The μ-analysis requires that the performance specifications are stated in the frequency domain. The performance specifications can be handled by incorporating two frequency dependent weighting functions, W_{11} and W_{12}, which reflect these specifications, into the μ-interconnection structure, figure 2.5.

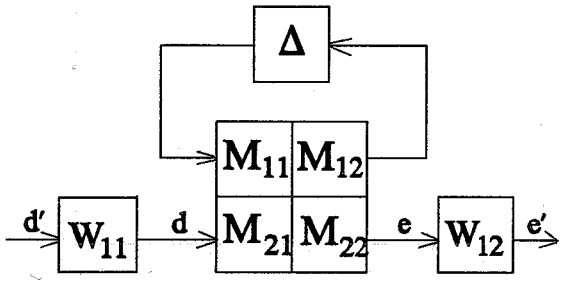


Figure 2.5 μ-interconnection structure with performance specifications.

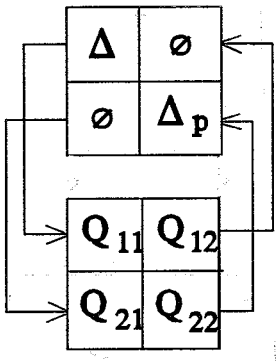


Figure 2.6 Robust performance in a robust stability framework.

Now d' and e' become signals which are a member of an unity set of signals. The set used in the μ-analysis is the set of in H_2 norm bounded signals [11].

$$\mathbf{BP} = \{x(s) | x(s) \in H_2,\ ||x(s)||_2 \leq 1\}$$

In figure 2.6 the weighting functions have been embedded into a stability framework. In this view robust performance analysis is exactly equivalent to robust stability analysis, the perturbation now consists of an uncertainty block and a performance block. The performance block Δ_p is a full block. Note that even if the uncertainty is unstructured the equivalent robust stability problem (robust performance problem) still has structure, because of the diagonal structure of the feedback loop (Δ, Δ_p).

Nominal performance (Δ=0) now is satisfied if and only if $||Q_{22}||_\infty < 1$. This means for an input signal in **BP** the output signal is also in **BP**. According to the small gain theorem this is equivalent requiring that the loop in figure 2.3 is stable for all $\Delta \in \mathbf{B}\Delta_f$ with M_{11} replaced by Q_{22}.

In the case of a SISO system a similar expression exists. For a SISO system nominal performance is achieved if $||Q_{22}|| = ||W_1(j\omega)S(j\omega)||_\infty < 1$. Where $S(j\omega)$ is the sensitivity function, the response of an error signal to a input signal. In figure 2.4 the sensitivity function is the transfer function from r or d to e. $W_1(j\omega)$ is the weighting function for the performance requirements. The weighting function has to be chosen by the designer taking into account the system specifications.

2.5 μ-Synthesis

The synthesis problem is finding a controller K for the system G such that the required performance is satisfied for all possible perturbations. This means finding a controller K that minimizes the following expression

$$||\mu(F_l(G(j\omega),K(j\omega)))||_\infty < 1 \quad \omega \in (-\infty, \infty), \Delta \in B\Delta_b, \Delta \in B\Delta_f \tag{2.6}$$

$G(j\omega)$ now also contains the weighting functions which reflect the performance requirements. There doesn't exist a completely satisfactory method to solve this problem. The D-K iteration is a method which is often used in practice. This method has been based on the properties for the μ-computation (appendix A). With D-scales expression (2.6) becomes

$$\min_{D,K} \|D(j\omega)(F_l(G(j\omega),K(j\omega)))D(j\omega)^{-1}\|_\infty < 1 \tag{2.7}$$

It has to be noted that the μ-value cannot always computed exact which means that expression (2.7) is an approximation for expression (2.6).

The D-K iteration

This method alternately minimizes expression (2.7) for either K or D, while holding the other constant. For fixed D the problem is just an H_∞ control problem. This means minimizing $||F_l(G,K)||_\infty$ for all K's. $F_l(G,K)$ is the transfer function from d to e shown in figure 2.7.

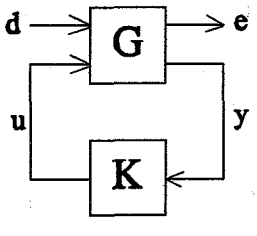


Figure 2.7 Block diagram for the H_∞-synthesis.

This minimization can only be done using an iterative scheme, the γ-iteration [3, 5]. For fixed K, (2.7) can be minimized for D at each frequency. This minimization is a convex optimization problem. The resulting D-scales can be fitted with proper stable transferfunctions. The software [3] contains several algorithm for this fitting procedure. In principle this method could be used to

obtain controllers that are arbitrary close to μ (optimal in the case of 3 or fewer perturbation blocks) and provide a nearly optimal controller for the general case. Because individual convexity in the two parameters K and D of the optimization problem does not always imply total convexity, convergence is not always guaranteed.

Chapter 3. RT-robot: controller design and simulation

3.1 Introduction

In this section a PD-controller, an H_∞-controller and an μ-controller for a RT-robot will be designed. The controller design will be carried out for two types of model errors, i.e. the motor dynamics and variations in the end-effector mass. With these controllers the RT-robot will be simulated and the controllers will be compared with each other. In section 3.3 a method is given to put the model errors caused by a not exact feedback linearization into the general interconnection structure. First, the RT-robot will be described.

3.2 Description of RT-Robot

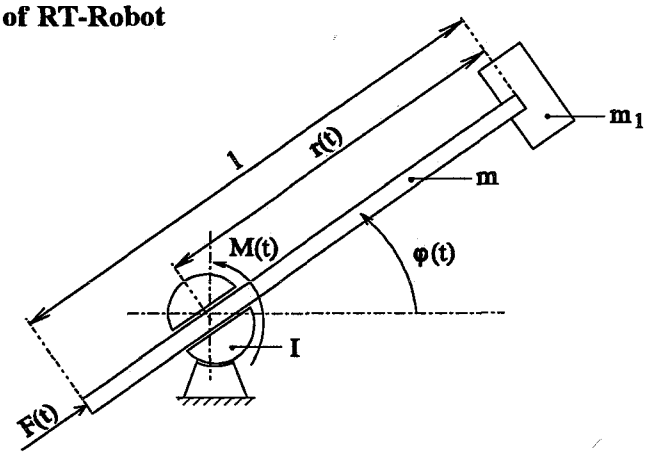


Figure 3.1 The RT-robot

Figure 3.1 illustrates the RT-robot (Rotation-Translation). The robot has two degrees of freedom, r(t) and φ(t), and moves in the horizontal plane. The robot consists of a disk with moment of inertia I and a rigid bar with length l and homogeneously distributed mass m. The load (m_1) is concentrated at the end of the bar. M(t) and F(t), the motor torque and force, are the control quantities of the robot. The model equations for this robot are

$$\begin{aligned} &P_1\ddot{r} - (P_1 r - P_2)\dot{\varphi}^2 = F \\ &(P_3 - 2P_2 r + P_1 r^2)\ddot{\varphi} + 2(P_1 r - P_2)\dot{r}\dot{\varphi} = M \end{aligned} \qquad (3.1)$$

with $P_1 = m + m_1 = 15$ [kg]

$$P_2 = \frac{1}{2}ml = 5 \text{ [kgm]}$$

$$P_3 = I + \frac{1}{3}ml^2 = 8\frac{1}{3} \text{ [kgm}^2\text{]}$$

3.3 Perturbations in state space systems

As mentioned before, the nonlinear model description cannot be used for the synthesis of the H_∞- and μ-controllers. A linear description, which matches the general interconnection structure

(figure 2.1) is required. The model (3.1) is linearized by a nonlinear state feedback. Figure 3.2 is a block diagram of this linearization. It has to be noted that the equations (3.1) have been written in another notation. In appendix B this notation is given. The exact feedback linearization is also described in this appendix.

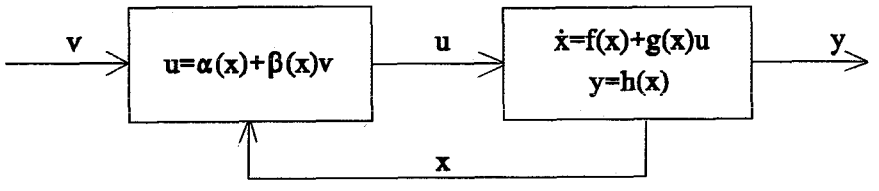


Figure 3.2. Block diagram of the state linearization.

If there are no errors in the model, the linearization is exact and results in a decoupled system of two second order differential equations. Now, the response of the output y to the input v is linear. Generally, the model contains model errors as a consequence of unmodelled dynamics and parameter variations. In that case the linearization is not exact and does not result in a decoupled system of two second order differential equations. After the feedback linearization the system will still be nonlinear and linearization along a trajectory is necessary to get the desired linear model description for the synthesis. The system matrix and the input matrix now contain extra terms comparing with the matrices obtained by an exact feedback linearization. The additional terms can be considered as perturbations on the nominal system. The perturbations have to be put in a perturbation matrix Δ, so they enter the system in a feedback form. This can be done as follows [7, 8]:

Define the nominal state space model of the system as

$$\begin{aligned} \dot{x} &= Ax + Bv \\ y &= Cx + Dv \end{aligned} \tag{3.2}$$

Parameters variations in the matrices A, B, C and D lead to non-dynamic real perturbations. If equations (3.2) are transformed to the Laplace domain the perturbations can also be additional dynamics, like actuator or sensor dynamics. The parameter variations are denoted as dA(s), dB(s), dC(s) and dD(s). The perturbed model is

$$\begin{aligned} sx &= Ax + Bv + dA(s)x + dB(s)v \\ y &= Cx + Dv + dC(s)x + dD(s)v \end{aligned} \tag{3.3}$$

As mentioned before the H_∞- and μ-synthesis requires the general interconnection structure. To extract the uncertain part of the model, a new input v_2 and a new output y_2 have to be defined. The following equations are obtained, figure 3.3:

$$\begin{aligned} \dot{x} &= Ax + Bv + B_2v_2 \\ y &= Cx + Dv + D_{12}v_2 \end{aligned}$$

$$\begin{aligned} y_2 &= C_2x + D_{21}v + D_{22}v_2 \\ v_2 &= \Delta y_2 \end{aligned} \tag{3.4}$$

All matrices in (3.4) are constant and of appropriate dimensions. The uncertainty feedback is given as $\Delta(s) = C_z + (sI-A_z)^{-1}B_z + D_z$, refer to [7]:

$$\begin{aligned} \dot{z} &= A_z z + B_z y_2 \\ v_2 &= C_z z + D_z y_2 \end{aligned} \tag{3.5}$$

In order to apply μ-analysis and synthesis it is necessary that the real parts of the eigenvalues of A_z are negative. Model (3.3) and (3.4), (3.5) are equivalent if

$$\begin{aligned} dA(s) &= B_2(I-\Delta(s)D_{22})^{-1}\Delta(s)C_2 \\ dB(s) &= B_2(I-\Delta(s)D_{22})^{-1}\Delta(s)D_{21} \\ dC(s) &= D_{12}(I-\Delta(s)D_{22})^{-1}\Delta(s)C_2 \\ dD(s) &= D_{12}(I-\Delta(s)D_{22})^{-1}\Delta(s)D_{21} \end{aligned} \tag{3.6}$$

If the matrices dA(s), dB(s), dC(s) and dD(s) are known the matrices B_2, C_2, D_{12}, D_{21} and D_{22} can be determined. However, the determination of these matrices is not unequivocal, also refer to [7].

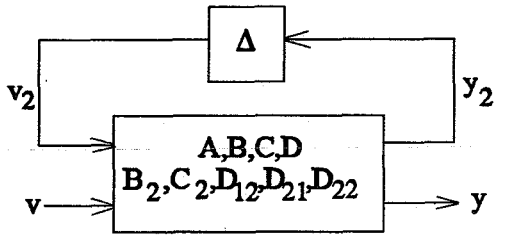


Figure 3.3 General interconnection of uncertainties.

3.4 Desired trajectory and the uncertainties

In this research two types of uncertainties are considered

1. Variations in the mass m_1. The load mass m_1 varies between 4 [kg] and 6 [kg], with a nominal value of 5 [kg].

2. Unmodelled dynamics, in this case the motor dynamics described by the following simple first order model

$$\begin{bmatrix} \tau_f & 0 \\ 0 & \tau_m \end{bmatrix} \dot{u}^* + u^* = u \tag{3.7}$$

With τ_f and τ_m the motor time constants. The motor time constants are chosen equal between

$\frac{1}{200}$ [s] and $\frac{1}{800}$ [s], with a nominal value of $\frac{1}{400}$ [s]. In figure 3.2, the motor dynamics has to be placed between linearization feedback block and the system block. Now the input to the system is u^*.

The perturbations in the system matrix A and the input matrix B don't only depend on the uncertainties above but also on the reference signal, because of the linearization along the desired trajectory, which is necessary after the not exact feedback linearization. The trajectory has been chosen as follows:

$$0 \le t \le 8.0 \text{ [s]}:$$

$$r_d = \frac{1}{2} + \frac{1}{2}\sin(\frac{1}{2}\pi t) \quad \text{[m]}$$

$$\varphi_d = \sin(\frac{1}{2}\pi t) \quad \text{[rad]} \tag{3.8}$$

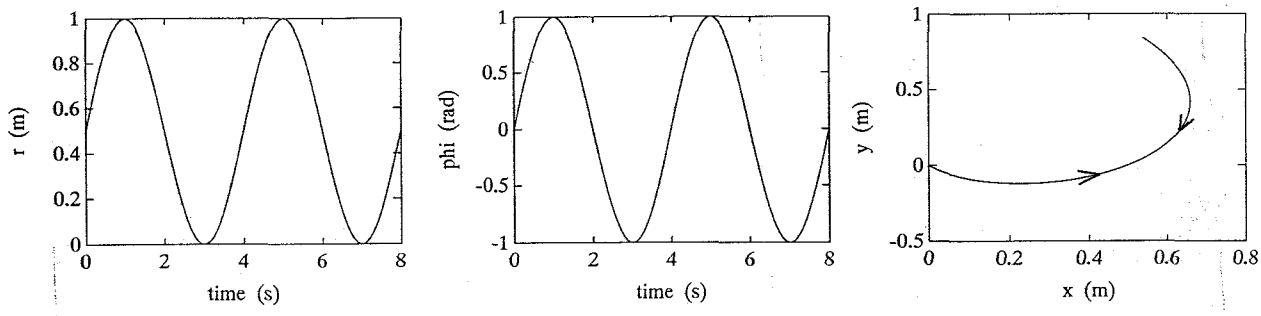


Figure 3.4 Desired trajectory.

In the case of the mass variations the system and input matrices contains at 6 places elements which differ from the nominal case. The matrices of the state space description after feedback linearization and linearization along the desired trajectory x^0, are

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & a_1 \\ 0 & 0 & 0 & 1 \\ a_2 & a_3 & 0 & a_4 \end{bmatrix} \qquad \tilde{B} = \begin{bmatrix} 0 & 0 \\ a_5 & 0 \\ 0 & 0 \\ 0 & a_6 \end{bmatrix} \tag{3.9}$$

with

$$a_1 = \frac{2(P_1-\bar{P}_1)}{P_1} x_4^0$$

$$a_2 = \frac{(2(\bar{P}_1-P_1)x_2^0x_4^0)(P_3-2P_2x_1^0+P_1(x_1^0)^2)-2(\bar{P}_1-P_1)x_1^0x_2^0x_4^0(-2P_2+2P_1x_1^0)}{(P_3-2P_2x_1^0+P_1(x_1^0)^2)^2}$$

$$a_3 = \frac{2(\bar{P}_1-P_1)x_1^0x_4^0}{P_3-2P_2x_1^0+P_1(x_1^0)^2}$$

$$a_4 = \frac{2(\bar{P}_1 - P_1)x_1^0 x_2^0}{P_3 - 2P_2 x_1^0 + P_1 (x_1^0)^2}$$

$$a_5 = \frac{\bar{P}_1}{P_1}$$

$$a_6 = \frac{P_3 - 2P_2 x_1^0 + \bar{P}_1 (x_1^0)^2}{P_3 - 2P_2 x_1^0 + P_1 (x_1^0)^2}$$

$\bar{P}_1 = m + \bar{m}_1$ with $\bar{m}_1 = 5$ [kg], the nominal load mass

Refer to appendix C for the derivation of the above expressions.
The perturbations can be considered as 6 independent scalar perturbations. In that case the perturbation Δ matrix will be a diagonal matrix (size 6x6). It could be expected that the perturbations only depend on the parameter P_1. However, they also depend on the nominal trajectory, which for example means that for one value of the parameter P_1 the perturbations can be both positive and negative. Of course the scalars $a_1 ... a_6$ are not completely independent. For this reason three new parameters are defined, see appendix C. If these parameters are used the derived perturbation matrix Δ consists of two repeated scalar block (size 2x2) and one scalar block (size 1x1). The derivation of this perturbation matrix Δ together with the matrices B_2, C_2, D_{12}, D_{21}, D_{22} and W_2 of the general interconnection structure matrix G (figure 2.1), determined with the help of the method described in section 3.3, are given in appendix C. The weighting function for the uncertainties (W_2) is obtained by scaling the perturbation matrix to unity. The concerned matrices are also given in appendix C.
In the case of the motor dynamics the uncertainties in the matrices A and B depend on the frequency.

$$\tilde{A}(s) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{2.56\tau_f s}{1+\tau_f s} & 0 & 0 & \frac{2.13\tau_f s}{1+\tau_f s} \\ 0 & 0 & 0 & 0 \\ \frac{7\tau_m s}{1+\tau_m s} & \frac{2\tau_m s}{1+\tau s} & 0 & \frac{\tau_m s}{1+\tau_m s} \end{bmatrix} \qquad \tilde{B}(s) = \begin{bmatrix} 0 & 0 \\ \frac{1}{1+\tau_f s} & 0 \\ 0 & 0 \\ 0 & \frac{1}{1+\tau_m s} \end{bmatrix} \tag{3.10}$$

The matrices B_2, C_2, D_{12}, D_{21}, D_{22}, the weighting function W_2 and the perturbation matrix Δ are also given in appendix C.

3.5 The controller design

In this section the PD-controller, the H_∞- and μ-controller are designed. The model obtained by an exact feedback linearization is used to determine the gains of the PD-feedback. As seen before the

exact linearization results in a decoupled system of two second order differential equations, which means that the controller design for the r- and φ-direction is also decoupled. The gains of the PD-feedback can be derived from the desired eigenfrequency ω_0 and the desired damping of the closed loop system. Larger values of the gains will lead to a quicker response to tracking errors. However, if the values of the gains are too large the controller will less robust with respect to unmodelled dynamics and parameter variations and the system could become unstable. The damping factor and eigenfrequency are equal for both directions and are chosen to be:

$$\omega_{0r} = \omega_{0\varphi} = 20 \quad [\text{rad/s}]$$
$$\beta_r = \beta_\varphi = 1 \quad [-]$$

The PD-feedback gains now become

$$v_1 = -K_{dr}(\dot{r}-\dot{r}_d) - K_{pr}(r-r_d)$$
$$v_2 = -K_{d\varphi}(\dot{\varphi}-\dot{\varphi}_d) - K_{p\varphi}(\varphi-\varphi_d) \qquad (3.11)$$

with $K_{pr} = K_{p\varphi} = \omega_{0r}^2$

$K_{dr} = K_{d\varphi} = 2\beta_r\omega_{0r}$

The software [3] used to design the H_∞- and μ-controllers requires a general interconnection structure matrix G (figure 2.1), which also contains the weighting functions for the uncertainty and the performance. The matrices of general interconnection structure matrix G including the weighting function for the uncertainties have been derived in appendix C. The weighting function for the performance (W_1) depend on the requirements of the system, like small tracking errors. In this research it has been tried to choose the weighting functions in such a way that the tracking errors will be as small as possible and that robust performance is guaranteed. This means that the μ-value of the closed loop system has to be less than 1 (chapter 2). The choice of the weighting function W_1 has been based on some design experiences, but it has been chosen reasonable arbitrary.

The first step in the μ-controller design is the computation of an H_∞-controller. In this case the scaling matrix D, see chapter 2, is an identity matrix of the size of the perturbation block Δ. After the H_∞-controller design the μ-value of the closed loop system and the D-scales are computed. The D-scales have to be fitted by proper stable transfer functions. The fitted transfer functions are wrapped around the original interconnection structure. This new structure is used to design a new controller. The D-K iteration continues until the μ-value does not change anymore. The last designed controller is the μ-controller.

For the analysis of the designed H_∞- and μ-controllers the following quantities are plotted:

- The μ-values of the closed loop systems, which indicate if robust performance is satisfied.
- The magnitude of the weighted nominal sensitivity functions. This quantity can be compared with $|W_1(j\omega)S(j\omega)|$ for a SISO system, see section 2.5. Because the system has two input and outputs and the r- and φ-direction have been decoupled two functions are plotted.
- The magnitude of the weighted complementary sensitivity functions. This quantity can be compared with $|W_2(j\omega)T(j\omega)|$ for a SISO system, see also section 2.5. Only the largest

functions are plotted.

- The weighted sensitivity functions of the perturbed closed loop system. The perturbations have been chosen corresponding to the type of model error.

1. *Variations in the mass m_1.*

During the controller design the perturbation matrix Δ consists of two full blocks (size 2x2) and one scalar block (size 1x1) block. The repeated scalar blocks mentioned in section 3.4 have been replaced by full blocks because the used software does still not contain an algorithm to fit full D-scales required for repeated scalar blocks. The performance block Δ_p is a full block (size 2x2). The weighting function for the performance (W_1) and the weighting function for the uncertainties (W_2), which is derived in appendix C are:

$$W_1 = \begin{bmatrix} \dfrac{250s^2+14400s+360000}{900s^2+250s+45} & 0 \\ 0 & \dfrac{250s^2+14400s+360000}{900s^2+250s+45} \end{bmatrix} \qquad W_2 = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.071 & 0 & 0 \\ 0 & 0 & 0 & 0.081 & 0 \\ 0 & 0 & 0 & 0 & 0.38 \end{bmatrix}$$

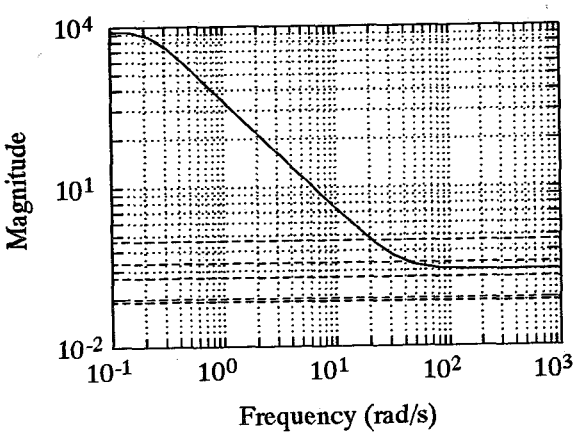


Figure 3.5 The weighting functions W_1 (solid) and W_2 (dashed)

H_∞-controller 13 states 2 inputs 2 outputs
μ-controller 41 states 2 inputs 2 outputs

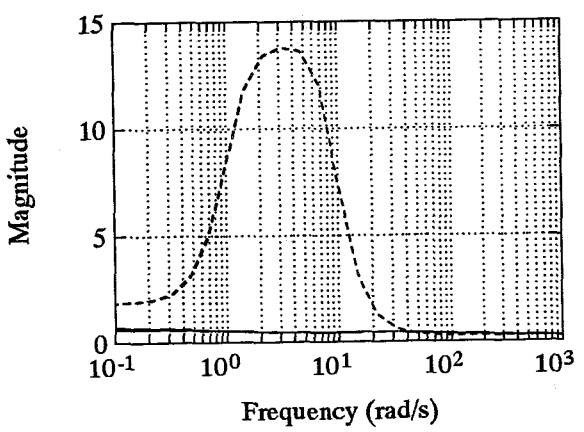


Figure 3.6 μ-values for the H_∞- (dashed) and μ-controller (solid)

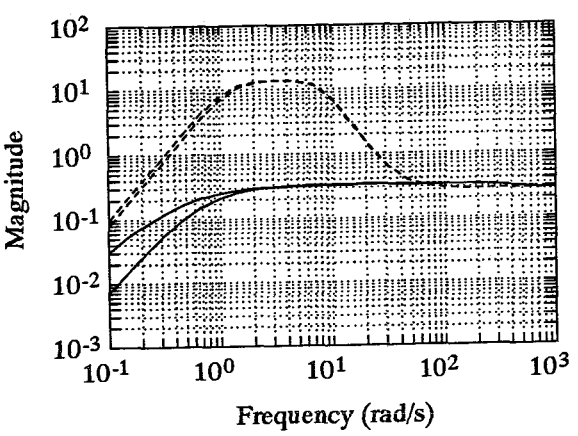


Figure 3.7 Nominal Weighted Sensitivity function for H_∞- (dashed) and μ-controller (solid)

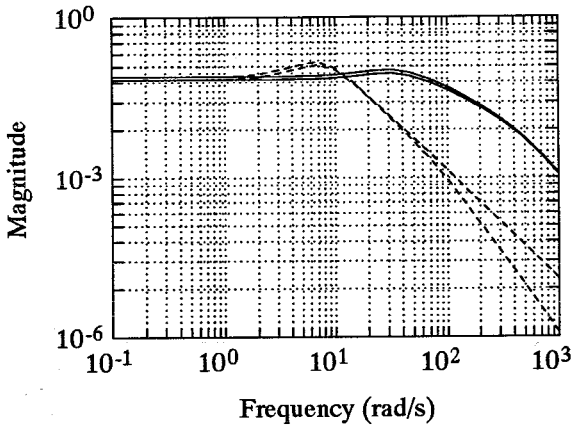


Figure 3.8 Nominal Weighted Compl. Sens. function for H_∞- (dashed) and μ-controller (solid)

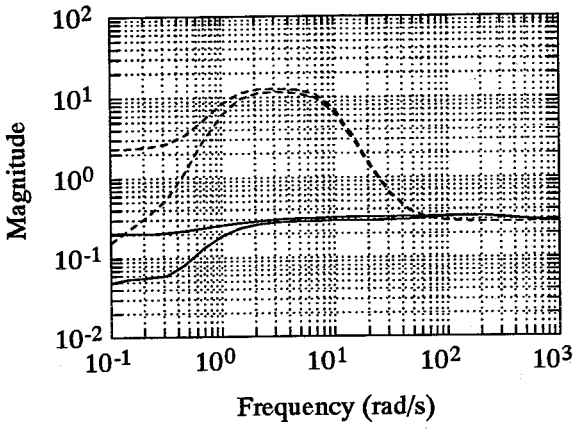


Figure 3.9 Weighted Perturbed Sens. function for H_∞- (dashed) and μ-controller (solid)

2. *Motor dynamics*

The weighting function for the uncertainties W_2, figure 3.10, consists of two complex scalar blocks (size 1x1) and is derived in appendix C. The performance block Δ_p is a full block (size 2x2). The weighting function for the performance (W_1) is also drawn in figure 3.10.

$$W_1 = \begin{bmatrix} \dfrac{s^2+80s+1600}{4s^2+0.5s+0.016} & 0 \\ 0 & \dfrac{s^2+80s+1600}{4s^2+0.5s+0.016} \end{bmatrix} \qquad W_2 = \begin{bmatrix} \dfrac{\tau s}{1+\tau s} & 0 \\ 0 & \dfrac{\tau s}{1+\tau s} \end{bmatrix} \qquad \text{with } \tau = \frac{1}{200}$$

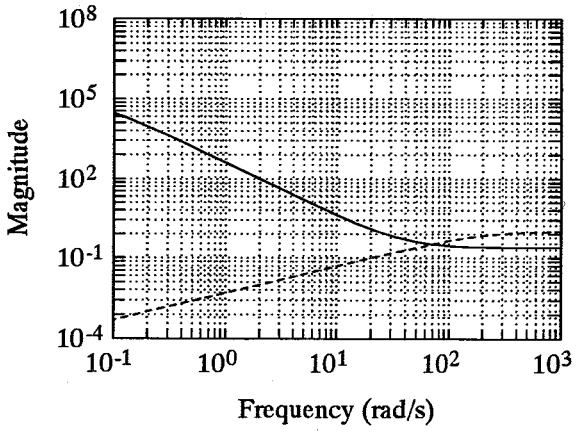


Figure 3.10 The weighting functions W_1 (solid) and W_2 (dashed)

H_∞-controller	10 states	2 inputs	2 outputs
μ-controller	26 states	2 inputs	2 outputs

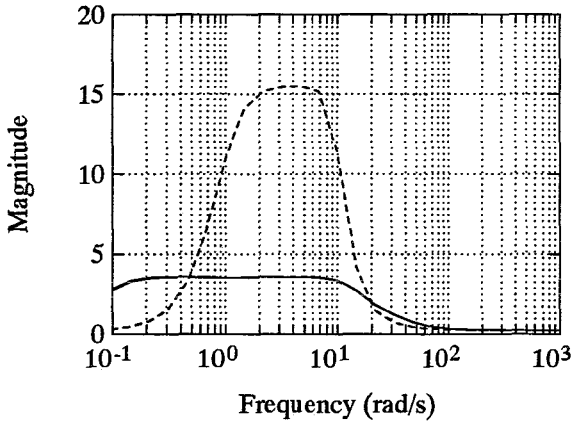


Figure 3.11 μ-values for the H_∞- (dashed) and μ-controller (solid)

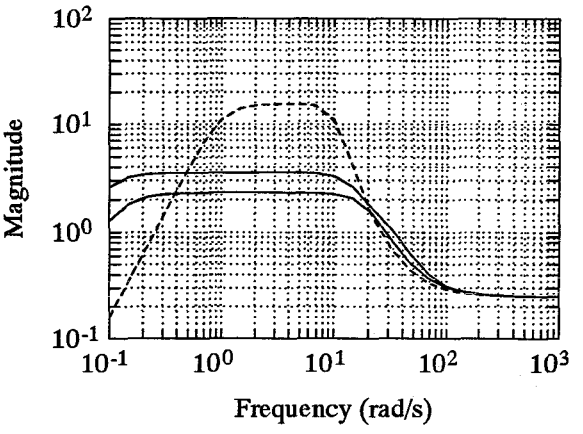


Figure 3.12 Nominal Weighted Sens. function for H_∞- (dashed) and μ-controller (solid)

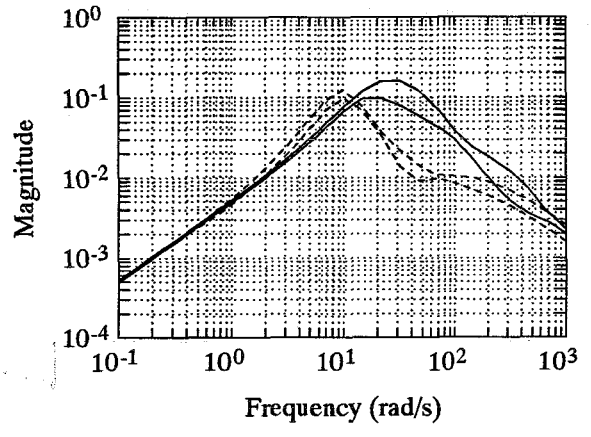


Figure 3.13 Nominal Weighted Compl. Sens. function for H_∞- (dashed) and μ-controller (solid)

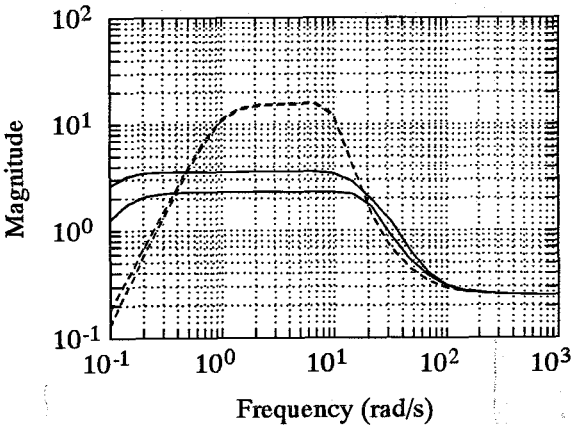


Figure 3.14 Weighted Perturbed Sens. function for H_∞- (dashed) and μ-controller (solid)

Analysis

In the case of the mass variations robust performance for the μ-controller has been satisfied, since the μ-value of the closed loop system is less than 1, figure 3.6. This does not apply for the H_∞-controller. Figure 3.9 illustrates that the perturbed sensitivity function for the H_∞-controller only differs from the nominal sensitivity function at low frequencies. This implies that the system with the H_∞-controller will probably not become unstable for the mass variations.

In the case of the motor dynamics it is not possible to design a μ-controller with a μ-value less than 1 if the weighting functions of figure 3.10 are used. This is the result of too high requirements for the performance. The nominal weighted sensitivity function, figure 3.12, shows that even in the nominal case the performance requirements are not satisfied. Although nominal and robust performance have not been satisfied it can be expected that the μ-controller performs better than the H_∞-controller which also holds in the case of the mass variations. Comparing figure 3.12 with figure 3.14 it can be concluded that for both controllers the sensitivity function doesn't change very much if uncertainty is included in the system, which implies that the system is probably robust with respect to the motor dynamics.

During the controller design some problems occurred. Some of these problems are related to the used software [3] and other problems have to be solved by the designer. Some problems are:

1. The calculation of the H_∞-optimal controller with the γ-iteration. The used algorithm for the γ-iteration requires an upper and lower bound for γ. The choice of these bounds has a large influence on the final controller design. Changing these bounds can lead to a complete other controller, which is far from the optimal controller.
2. Numerical problems if the matrices used for the H_∞-synthesis becomes too large, so that they are very close to singular. The matrices can become too large for several reasons:
   - The order of the transfer functions to fit the D-scales is chosen too high.
   - The order of the weighting functions for the performance requirements and model errors are too large.
   - There are too many requirements for the system which implies that many weighting functions have been required
3. It is not always clear which order transfer function for the D-scales has to be chosen. Sometimes a less accurate fit finally leads to a better result, because for example numerical problems does not occur.
4. An algorithm to fit full D-scales required for repeated scalar blocks has still not been implemented in the software [3], which means that in the case of the mass variations the structure of the uncertainty has been lost.

3.6 Implementation of the controllers

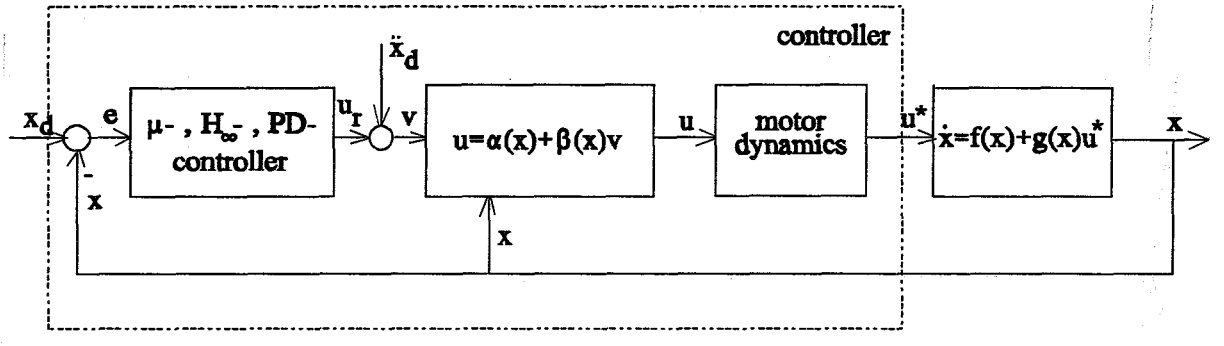


Figure 3.15 Block diagram of the closed system used for the simulation.

For the implementation of the controllers the nonlinear model of the RT-robot (3.1) is used. In the case of the motor dynamics the closed loop system is illustrated in figure 3.15. The μ-, H_∞-and PD-controller, the linearization feedback and the motor dynamics are parts of the total controller. The controller has also been extended with a feedforward of the desired acceleration, so that in the nominal case the tracking error will be zero.

For the variations in mass m_1 the 'block' motor dynamics has to be removed. The mass m_1 in the nonlinear model now is not equal to the nominal mass (15 [kg]). $\dot{x} = f(x) + g(x)u$ is the state space description of the nonlinear model described in section 3.2. For the simulations the trajectory of section 3.4 is used. In figure 3.16 and 3.17 the tracking errors for the r-direction [mm] and the φ-direction [rad] are plotted for μ- and PD-controller. The mass m_1 = 6 [kg]. The tracking error of the H_∞-controller has been left away because it is much larger. Figures 3.18 and 3.19 illustrate the

RMS-values (Root Mean Square) of these errors for different values of m_1. In the case of the motor dynamics tracking errors for both directions are plotted in figures 3.20 and 3.21. The motor time constants are $\tau = \tau_f = \tau_m = \frac{1}{200}$ [s]. Figures 3.22 and 3.23 show the RMS-values of the tracking errors for several time constants.

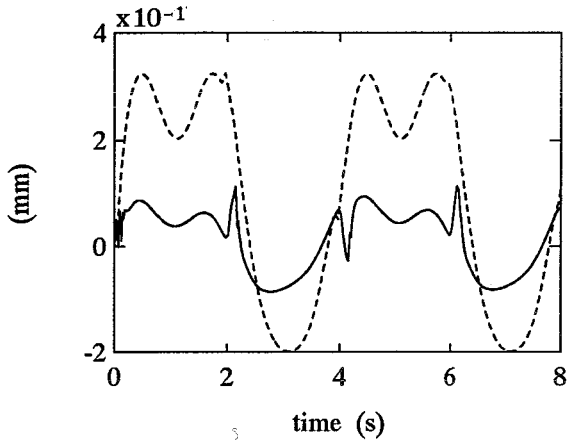


Figure 3.16 Tracking error r-direction (m_1=16 kg) μ-controller (solid) and PD-controller (dashed)

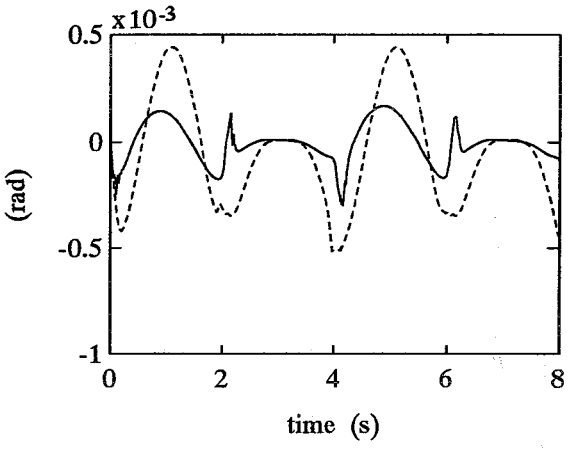


Figure 3.17 Tracking error φ-direction (m_1=16 kg)

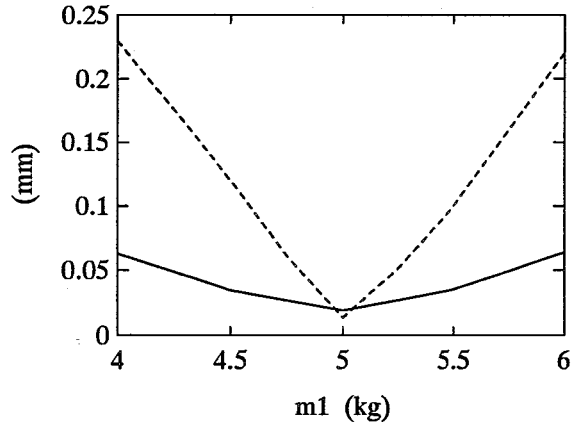


Figure 3.18 RMS-value of the tracking error in r-direction
μ-controller (solid) and PD-controller (dashed)

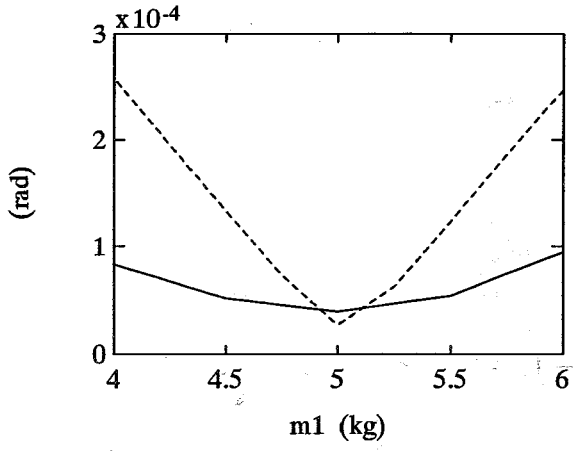


Figure 3.19 RMS-value of the tracking error in φ-direction

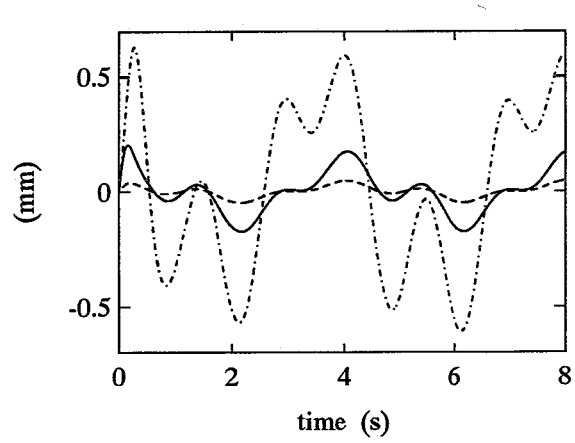


Figure 3.20 Tracking error r-direc. (τ=1/200 s) μ-controller (solid), PD-controller (dashed) and H_∞-controller (dot dashed)

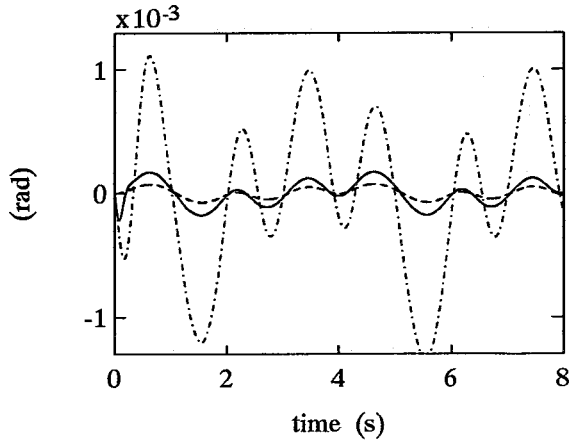


Figure 3.21 Tracking error φ-direc. (τ=1/200 s)

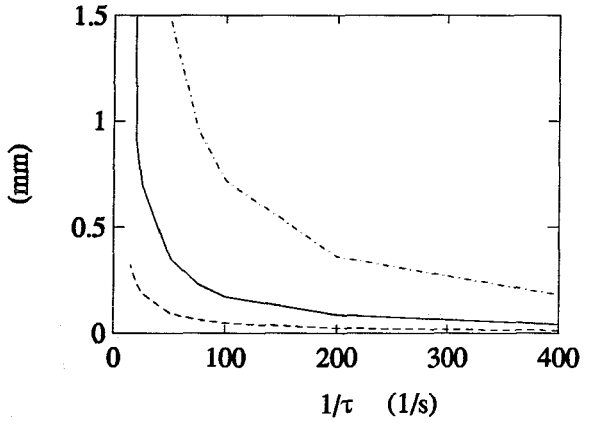


Figure 3.22 RMS-value of the tracking error in r-direction

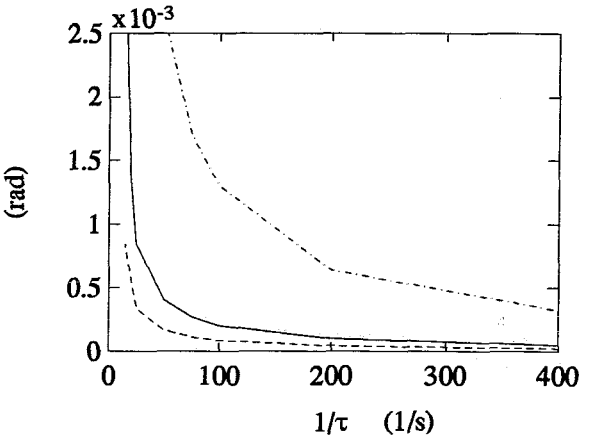


Figure 3.23 RMS-value of the tracking error in φ-direction

μ-controller (solid), PD-controller (dashed) and H_∞-controller (dot dashed)

3.7 Discussion and conclusions

From the simulations it can be concluded that in all cases the μ-controller performs better than the H_∞-controller, which completely corresponds to the expectations from the analysis. However, the μ-controller does not always perform better than the PD-controller. In the case of the motor dynamics the PD-controller performs a little better for all time constants, figures 3.22 and 3.23, and is even more robust than the μ-controller, which sooner becomes unstable.

In the case of the mass variations the PD-controller performs better in the nominal case (m_1 = 5 [kg]), but in the other cases the μ-controller is better, figures 3.18 and 3.19, which implies that the μ-controller is more robust with respect to mass variations. It has to be noted that in the nominal case the tracking error for the PD-controller had to be zero because of the feedforward of the desired acceleration. However as a consequence of numerical inaccuracies this error is not equal to zero.

For both types of model errors the H_∞- and μ-controllers are robust to a larger range of uncertainties than which the controllers have been designed for. This means that the method used to estimate the uncertainty as a consequence of the not exact feedback linearization leads to too conservative controllers. Another reason for the conservativeness is the controller design. For the μ computation all possible perturbations (real and complex) are considered. Maybe a complex perturbation can cause instability, but a real perturbation does not, so the controller design seems conservative.

Of course the weighting functions for the uncertainty can be chosen smaller so the controller is robust to a smaller range of uncertainties. The weighting function for the performance can then be chosen larger so that the tracking error decreases. In this way a better controller can be obtained. Also another choice for the weighting function which reflect the performance requirements could lead to a better controller.

Chapter 4. The xy-table: controller design and simulation

4.1 Introduction

In this section a PD-controller, a H_∞-controller and μ-controller for a flexible manipulator, the xy-table, will be designed. The controller design in x- and y-direction will be decoupled. However, no attention is paid to the controller for the y-direction, because no research has been done into possible model errors in this direction. For the x-direction the flexibility in the shaft will be considered as a perturbation. In section 4.3 a method to determine this perturbation is presented. The controllers will be simulated for several values of the stiffness of the shaft. The results are given in section 4.6. A short discussion is at the end of this chapter.

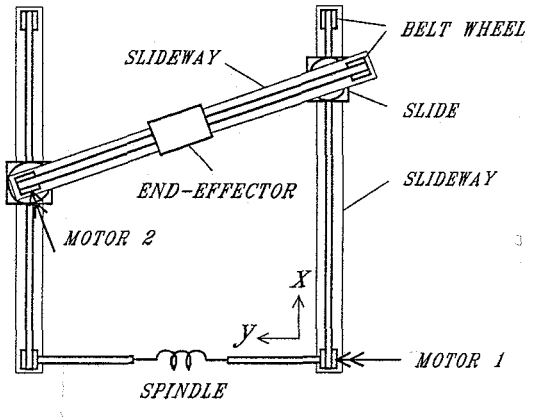


Figure 4.1a The xy-table

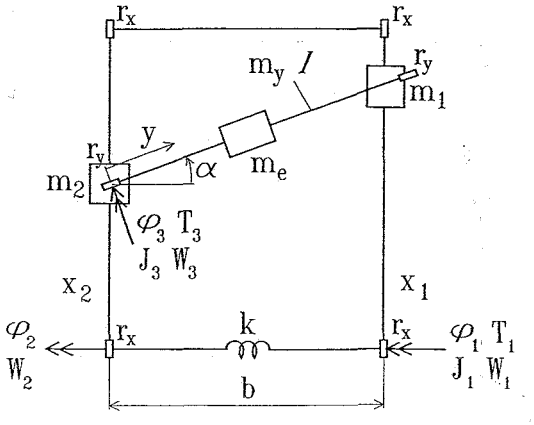


Figure 4.1b The simulated system

4.2 Description of xy-table

Figure 4.1a gives a schematic representation of the xy-table. The system used for the simulation is shown in figure 4.1b. The end-effector with mass m_e can move in the horizontal plane by means of three slideways. The system has three degrees of freedom, the rotations $\varphi_1(t)$, $\varphi_2(t)$ and $\varphi_3(t)$. The couples T_1 and T_3 generated by two servo-motors are the control quantities of the system. The coulomb friction along the slideways is represented by the torques W_1, W_2 and W_3. In appendix D the equations of motion of the model and the values of the parameters are given. If there is no torsion spring k the model can be simplified to a model with two degrees of freedom, see also appendix D.

$$P_1\ddot{\varphi}_1 = T_1 - P_3\text{sign}(\dot{\varphi}_1)$$
$$P_2\ddot{\varphi}_3 = T_3 - P_4\text{sign}(\dot{\varphi}_3) \qquad (4.1)$$

with $P_1 = 4.68*10^{-3}$ [kgm^2]
$P_2 = 4.60*10^{-4}$ [kgm^2]
$P_3 = 0.50$ [Nm]
$P_4 = 0.15$ [Nm]

This model, but without coulomb friction, will be used for the controller design.

4.3 Description of the uncertainties

In this section the weighting function which reflects the model errors in the x_1-direction is determined. The model errors resulting from the torsion spring are the only uncertainties which are taken into account during the controller design. The torsion spring only has influence on the rotations φ_1 and φ_2 and almost no influence on φ_3. The influence of the torsion spring is considered as a multiplicative perturbation on the input T_1.
The nominal transferfuntion from the motortorque T_1 to the rotation φ_1 is

$$V(s) = \frac{1}{P_1 s^2} = \frac{213.7}{s^2} \tag{4.2}$$

The transfer function above has been obtained by Laplace transformation of (4.1), the simplified model without flexibility. When the torsion spring k is not neglected the transfer function is

$$\tilde{V}(s) = \frac{0.67*10^{-3}s^2 + 3*10^{-5}s + k}{2.38*10^{-6}s^4 + 1.28*10^{-7}s^3 + 4.61*10^{-3}ks^2 + 6*10^{-5}ks} \tag{4.3}$$

with k = torsion stiffness in [Nm/rad]

This transfer function has been derived from expression (D.1), the advanced model of the xy-table. Small terms of this expression have been neglected and some viscous damping, which was not modelled, has been added. A detailed derivation is given in appendix E.
For each value of the torsion spring, a peak appears in the transfer function that will move to higher frequencies with increasing torsion stiffness. In figure 4.2 the transfer function $\tilde{V}(s)$ is plotted for some values of k. The influence of the torsion spring is treated as a multiplicative perturbation [2]. The transfer function $\tilde{V}(s)$ can be written as follows

$$\tilde{V}(s) = (1 + \Delta(s)W_2(s))V(s) \quad \text{with} \quad |\Delta(s)|_\infty < 1 \tag{4.4}$$

or in another form

$$\left| \frac{\tilde{V}(s)}{V(s)} - 1 \right| \leq |W_2(s)| \quad \forall\, s \tag{4.5}$$

For the weighting function $W_2(s)$ the following form has been chosen

$$W_2(s) = \frac{s^2}{213.7} * \frac{a}{(s^2 + 2\beta_i\omega_i + \omega_i^2)} \tag{4.6}$$

The parameters a, ω_i, and β_i have to be chosen in such a way that the left part of inequality (4.5) is fitted as good as possible, but for all frequencies the magnitude of this part has to be smaller than the magnitude of the weighting function. In figure 4.3 the weighting functions (dashed lines) are plotted for the same values of k as in figure 4.2. In all cases hold β_1=0.01 [-] and a=40 [$kg^{-1}m^2$]. For increasing stiffness the parameter ω_i is 29, 36, 41, 70, 220 [rad/s], respectively. The total weighting function which will be used for the controller design has to fit all separate weighting functions (dashed lines), because the controller has to perform well for all values of the stiffness larger than 0.46 [Nm/rad]. Two possible functions are plotted, i.e. solid lines in figure 4.3. It is not easy to say which function describes the model errors the best. However, a too large weighting function for the uncertainties will probably lead to a too conservative controller, and a too small function probably won't guarantee the desired robustness.

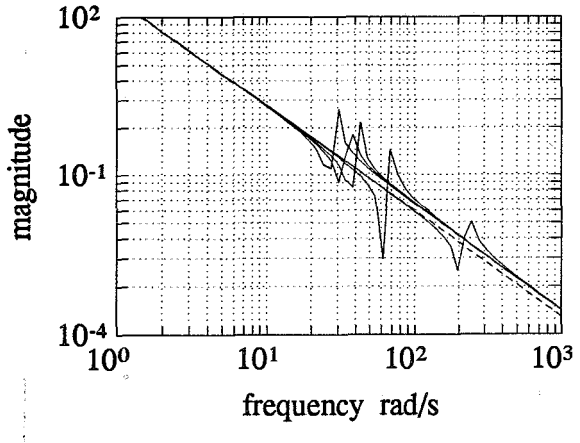


Figure 4.2 Transfer function (4.3) for several k's (0.46, 0.69, 0.93, 2.55, 28.1 [Nm/rad])

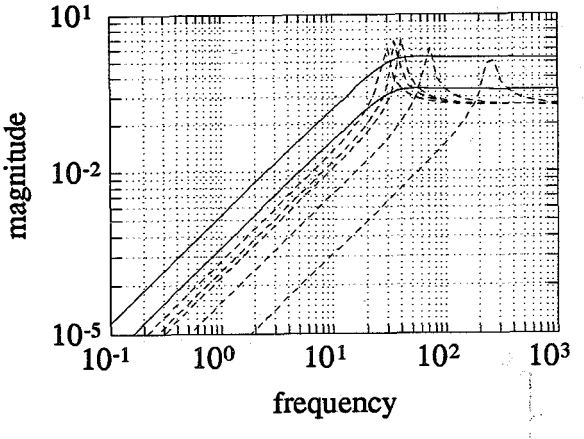


Figure 4.3 Fitted weighting functions

4.4 Controller design and analysis

As mentioned in section 4.1 the controller designs for the x- and y-direction are decoupled. Only the controller design in x-direction is considered because the flexibility of the system in this direction can be changed. The uncertainties described in the previous section are taken into account during the controller design. A PD-controller is designed with a view to compare the H_∞- and μ-controller with this controller. The model (4.1) without friction has been used to determine the gain of the PD-feedback. The gain depends on the desired eigenfrequency ω_0 and the desired damping factor β of the closed loop system. A larger gain results in a quicker response of the system and in smaller tracking errors. Because of unmodelled dynamics, e.g. the torsion spring, and measurement noise the gain may not be too large. A too large value of the gain can cause instability.
If the torsion spring has it's smallest value (k=0.46 [Nm/rad]) the peak in the transfer function (4.3) appears at about the frequency of 29 [rad/s], see figure 4.2. The eigenfrequency ω_0 of the controlled system has to be smaller than this frequency. The eigenfrequency and the damping factor are chosen to be:

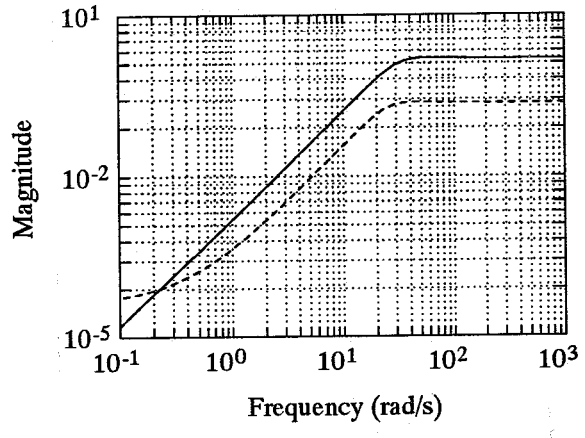
$\omega_{0\varphi 1} = 3.5*2\pi$ [rad/s]
$\beta_{\varphi 1} = 0.7$ [-]

The PD-feedback now becomes

$$T_1 = -K_{d\varphi 1}(\dot{\varphi}_1 - \dot{\varphi}_{1d}) - K_{p\varphi 1}(\varphi_1 - \varphi_{1d})$$

with $K_{p\varphi 1} = \omega_{0\varphi 1}^2 P_1$
$K_{d\varphi 1} = 2\beta_{\varphi 1}\omega_{0\varphi 1} P_1$

φ_1 and $\dot{\varphi}_1$ are the motorposition [rad] and the motor velocity [rad/s] of the servomotor, respectively.
The H_∞- and μ-controllers are designed for several weighting functions for the performance and the uncertainties. This will show the influence of the choice of the weighting functions. The weighting functions for model errors are given in figure 4.4.

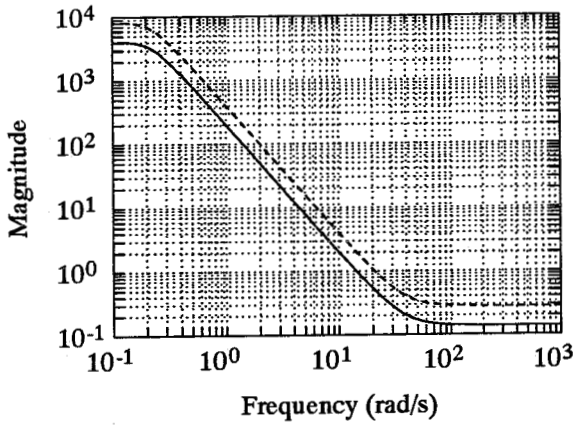


$$W_2(s) = \frac{s^2 + 0.04s + 9*10^{-4}}{0.67s^2 + 20s + 418} \quad \text{(solid)}$$

$$W_2(s) = \frac{0.15s^2 + 0.14s + 0.02}{0.67s^2 + 20s + 418} \quad \text{(dashed)}$$

Figure 4.4 Weighting functions which reflect the model errors

Of course, a lot of weighting functions are possible, but these have been chosen because they roughly describe the range of possible weighting functions. The first one is probably too conservative and the second one does probably not reflect the model errors sufficiently. Notice that the weighting functions in figure 4.4 are not the same as those in figure 4.3.
The choice of the weighting function for the performance depends on the system's requirements. In this case it has been tried to make the tracking errors as small as possible. However, the weighting function has to be chosen in such a way that robust performance (μ<1) is satisfied, which means that the requirements should not be too high.



$$W_1(s) = \frac{125s^2+7200s+180000}{900s^2+250s+45} \quad \text{(solid)}$$

$$W_1(s) = \frac{250s^2+14400s+360000}{900s^2+250s+45} \quad \text{(dashed)}$$

Figure 4.5 Weighting functions for the performance

For two sets of weighting functions the controllers have been designed and the following quantities are plotted:

- The μ-values of the closed loop system, which indicate if robust performance is satisfied.
- The magnitude of the weighted nominal sensitivity function. This function is equal to $|W_1(s)S(s)|$ because it concerns a SISO system.
- The magnitude of the weighted complementary sensitivity function. This function is equal to $|W_2(s)T(s)|$ because it concerns a SISO system.
- the magnitude of the weighted perturbed sensitivity function, which is the sensitivity function of the closed loop system if perturbations are added to the system.

1. $W_1(s) = \dfrac{250s^2+14400s+360000}{900s^2+250s+45}$ $\quad W_2(s) = \dfrac{0.15s^2 + 0.14s + 0.02}{0.67s^2 + 20s + 418}$

H_∞-controller	6 states	1 input	1 output
μ-controller	14 states	1 input	1 output

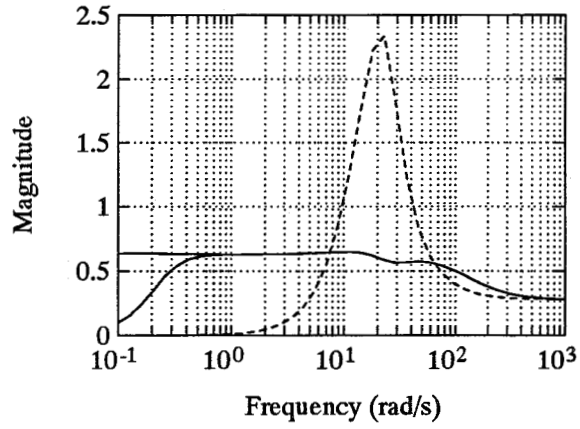


Figure 4.6 μ-values for the H_∞- (dashed) and μ-controller (solid)

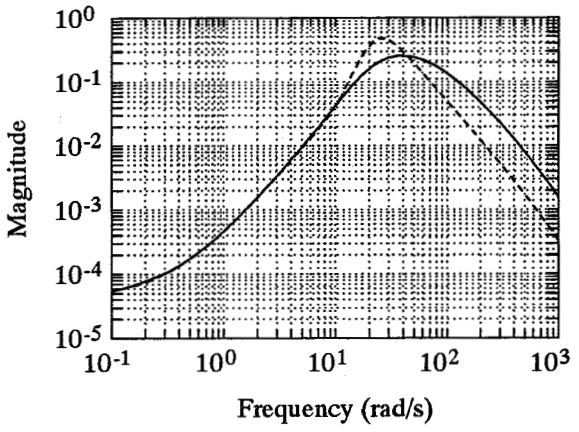


Figure 4.7 Nominal weighted compl. sens. function for H_∞- (dashed) and μ-controller (solid)

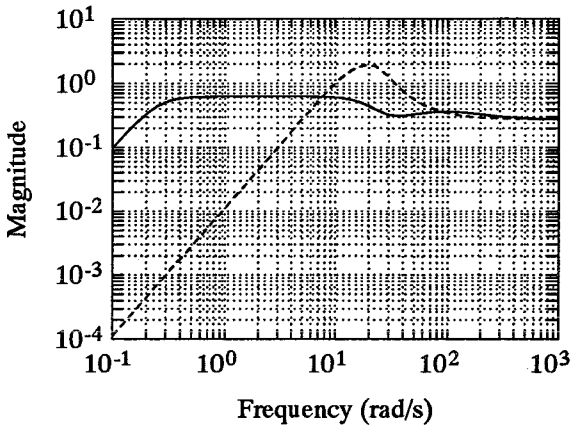


Figure 4.8 Nominal weighted sens. func for H_∞- (dashed) and μ-controller (solid)

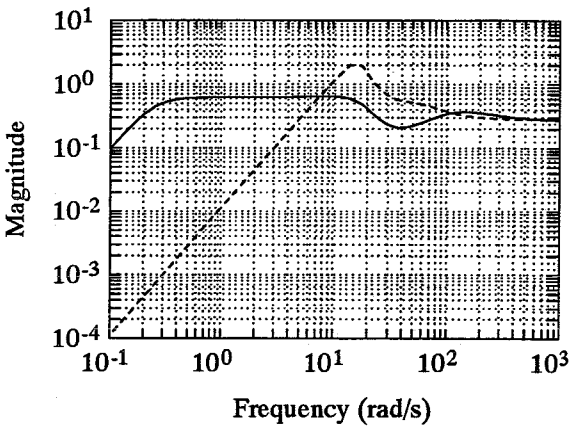


Figure 4.9 Perturbed sens. function for H_∞- (dashed) and μ-controller (solid)

2. $W_1(s) = \dfrac{125s^2+7200s+180000}{900s^2++250s+45}$ $W_2(s) = \dfrac{s^2 + 0.04s + 9*10^{-4}}{0.67s^2 + 20s + 418}$

H_∞-controller	6 states	1 input	1 output
μ-controller	14 states	1 input	1 output

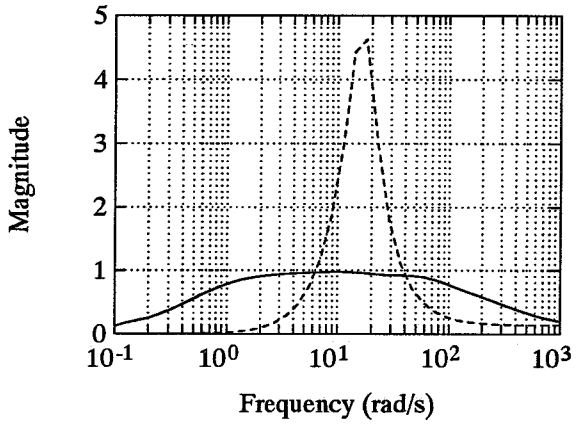


Figure 4.10 μ-values for the H_∞- (dashed) and μ-controller (solid)

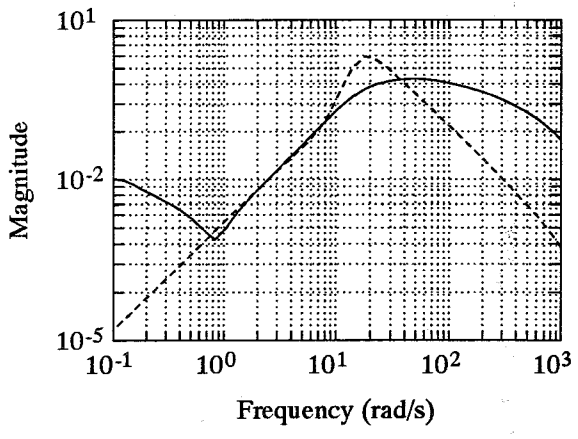


Figure 4.11 Nominal weighted compl. sens. function for H_∞- (dashed) and μ-controller (solid)

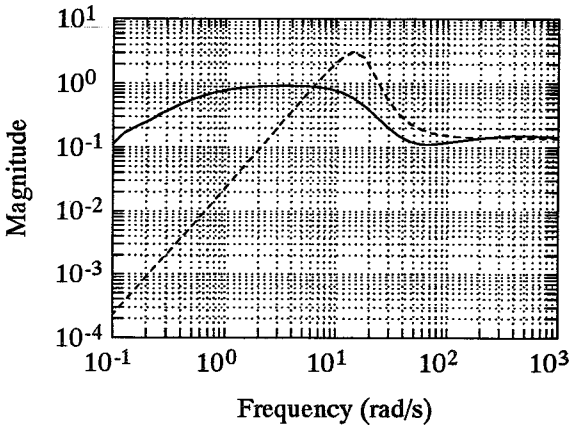


Figure 4.12 Nominal weighted sens. func for H_∞- (dashed) and μ-controller (solid)

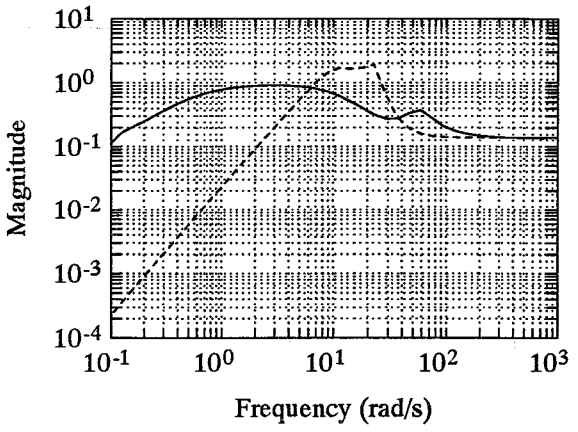


Figure 4.13 Perturbed sens. function for H_∞- (dashed) and μ-controller (solid)

Analysis

In both cases a μ-controller design with $\mu<1$ is possible. This means that robust performance has been satisfied for all permitted perturbations, which implies that the xy-table will be stable for the possible torsion springs. In the case of the H_∞-controller robust performance cannot be guaranteed. Nevertheless, from the figures 4.9 and 4.13 it can be expected that for both controller designs the system will be stable for all possible values of the stiffness, because the perturbed sensitivity functions hardly differ from the nominal sensitivity functions, figures 4.8 and 4.12, respectively. These figures also illustrate that the μ-controller will perform much better than the H_∞-controller. From figure 4.6 could be concluded that the performance requirements, represented by the weighting function W_1 could be chosen larger. Of course this function has to be chosen in such a way that robust performance is always guaranteed, thus $\mu<1$.

4.5 State reduction and discretization of the μ-controller

Before simulating the xy-table with the designed controllers the number of states of the μ-controller has to be reduced. This state reduction is necessary for two reasons:

- Because of the large number of states, the matrix of the controller often has a bad condition. This could lead to discretization and implementation problems.
- The number of states is too large for a real time implementation, see also chapter 5.

The state reduction has been carried out with the Square Root Balanced Method, which is an additive error model reduction method [4]. The number of states can be reduced by fifty percent without decrease of the performance.

The matched pole-zero method has been used for the discretization. This method has been implemented in a MATLAB-routine and is only suitable for SISO systems.

It is recommended first, to reduce the number of states of the controller and secondly, to discretize the controller because the reverse approach could lead to numerical problems.

4.6 Simulation of xy-table

The xy-table has been simulated with the PD-controller and discrete H_∞- and μ-controllers. Discrete controllers have been used to simulate the practice as good as possible. In a real time system a discrete controller has always to be used, because it takes processing time to compute the new inputs for the system. An additional advantage is that the controllers used during the simulation can be directly applied in the real system. The advanced model, described in appendix D has been used as simulation model. The simulations were done for different values of the stiffness of the torsion spring. The controllers have been extended with a compensation for the coulomb friction and a feedforward of the desired acceleration as illustrated in figure 4.14.

The sampling frequency was 143 [Hz]. Not the measured rotation has been used as input for the controller, but the rotation and the speed (only in the case of the PD-controller) one sample ahead estimated by a discrete Kalman observer [1] to cancel the time delay caused by the computational time delay.

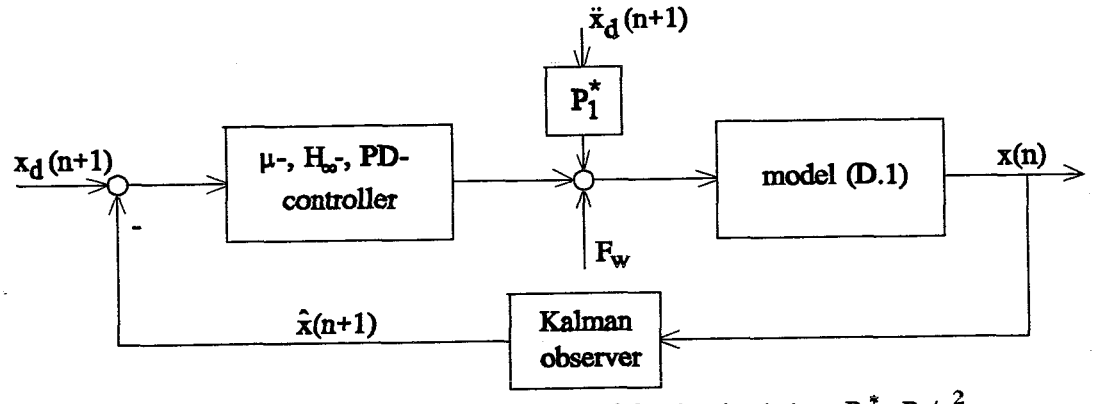


Figure 4.14 Block diagram of the closed loop system used for the simulation, $P_1^* = P_1/r_x^2$.

The desired trajectory was a circle

$$\begin{aligned} x_d &= 0.8 - 0.2\cos(2\pi ft) && [m] \\ y_d &= 0.8 + 0.2\sin(2\pi ft) && [m] \\ f &= 3.5\pi/4 && [rad/s] \end{aligned}$$

The above trajectories are the desired positions of the servomotors, not to be confused with the x- and y-position of the end-effector. It has to be noted that controllers for the x-direction have been designed for the rotation φ_1. The relationship between this rotation and the displacement above is $\varphi_{1d} = \frac{x_d}{r_x}$

As mentioned before only the x-direction has been considered. For the PD-controller and the controllers designed with the weighting functions of set 1 the tracking errors [mm] of the motor position and the tracking errors of the end-effector position are plotted, figures 4.15 and 4.16 respectively. The value of the stiffness of the torsion spring was k=0.46 [Nm/rad], the smallest value for which the controllers have been designed. In the figures 4.17 and 4.18 the RMS-values (Root Mean Square) of the motor and the end-effector position errors are plotted for several values of the stiffness.

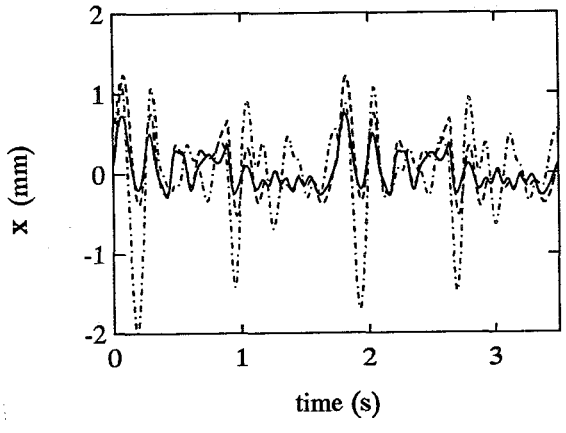


Figure 4.15 Tracking error of the motor position for k=0.46 [Nm/rad]

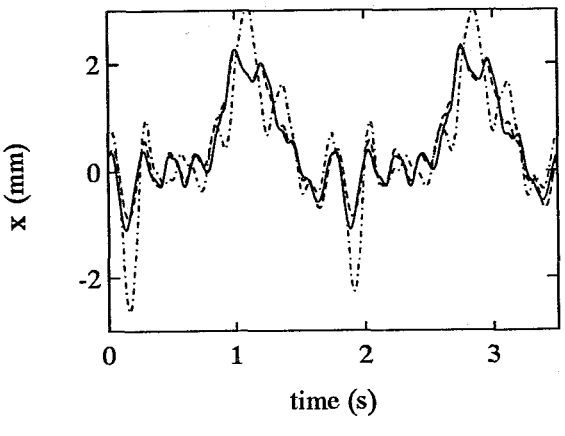


Figure 4.16 Tracking error of the end-effector position for k=0.46 [Nm/rad]

μ-controller (solid), H_∞-controller (dashed-dot) and PD-controller (dashed).

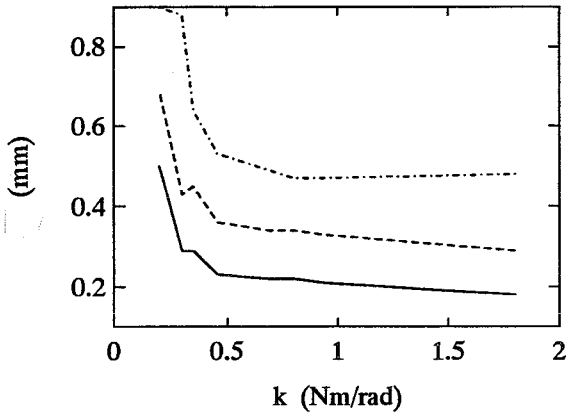


Figure 4.17 RMS-values of the tracking errors of the motor position μ-controller (solid), H_∞-controller (dashed-dot) and PD-controller (dashed).

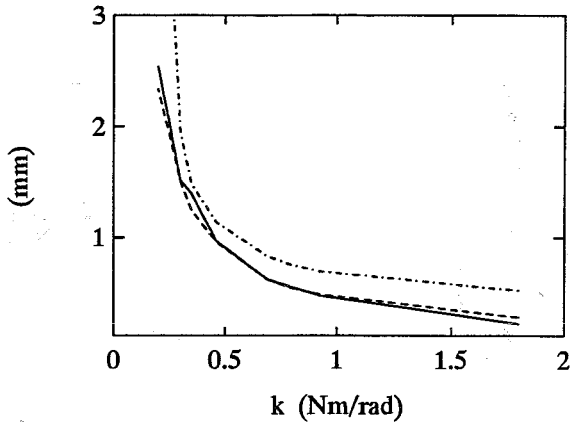


Figure 4.18 RMS-values of the tracking errors of the end-effector position μ-controller (solid), H_∞-controller (dashed-dot) and PD-controller (dashed).

For the controllers designed with the weighting functions of set 2 only the tracking error of the motor position and the RMS-values of motor position errors are plotted.

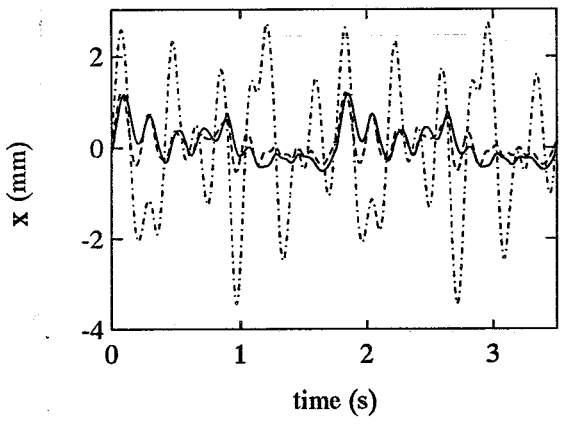


Figure 4.19 Tracking error of the motor position for k=0.46 [Nm/rad] μ-controller (solid), H_∞-controller (dashed-dot) and PD-controller (dashed).

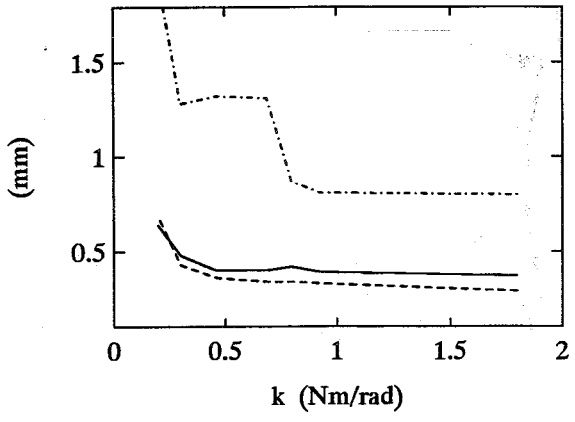


Figure 4.20 RMS-values of the tracking errors of the motor position μ-controller (solid), H_∞-controller (dashed-dot) and PD-controller (dashed).

4.7 Discussion and conclusions

From figures 4.15 and 4.17 should be concluded that the μ-controller performs better than the PD-controller. It has to be noted that PD-feedback gain can be chosen larger, so the tracking errors will be smaller, without causing instabilities during the simulations. However, former research [1] has shown that large gains cause a chattering response if the controller is used in an experiment. With a view to these experiments the PD-feedback gain has not been chosen larger. Whether the designed μ-controller will also perform in practice is investigated in chapter 5.

If the weighting function for the model errors is chosen larger and the weighting function for the performance smaller than the PD-controller has the same performance as the μ-controller, see figures 4.19 and 4.20. In both cases the μ-controller performs better than the H_∞-controller which is according to the expectations from the analysis in section 4.5. However, the H_∞-controllers do not become unstable inspite of the fact that in both cases robust performance (μ<1) is not satisfied,

figures 4.6 and 4.10. The robustness with respect to the torsion spring of the three controllers is almost the same, see figures 4.17 and 4.19. It has to be noted that the H_∞- and μ-controllers are robust for the whole range of possible values of the stiffness. For smaller values of the stiffness the motor position error increases faster, however the system stays stable. The intersection of the RMS-values of the PD- and μ-controller in figure 4.20 does not directly mean that the μ-controller is more robust. To investigate this robustness it will be necessary to do simulations with smaller values of stiffness of the torsion spring. However, it is clear that the robustness will probably not increase without reduction of the performance.

The tracking errors of the end-effector do not decrease with the μ-controller. This could be expected because the controllers were based on the motorposition error and not on the end-effector error.

Chapter 5. xy-table implementation

5.1 Introduction

The controllers of chapter 4 will now be implemented in the xy-table. First, a short description of the real xy-table and the used software are given. The experiments will be carried out for two different trajectories, both circles. In contrast to the simulations the tracking errors in the y-direction will also briefly be considered. In section 5.4 other μ- and H_∞-controllers are proposed and tested. The chapter is closed with a short discussion and some conclusions.

5.2 Description of the real xy-table

The xy-table has already been described in chapter 4. The only difference with the simulations is the position of the origin of the x- and y-axis which is now exactly in the middle of the operating area. The used control software, written in the computer language C^{++}, imposes this definition for the axis. The μ- and H_∞-controllers of chapter 4 which were simulated in Matlab have to be also converted to this language.
The rotations φ_1, φ_2 and φ_3 are measured by means of three incremental encoders fixed at both servomotors and at belt wheel 2, see figure 4.1. However, not the rotations in [rad], but the motor positions in [m] are available at every sample time. Some other quantities like the acceleration of the end-effector in x- and y-direction and the current to both servomotors are also measured, but they are not used to determine the new controller outputs. Unfortunately, the position of end-effector cannot be measured because the optical measurement system to do this was not operational during this study. With the help of the rotation φ_2 it is possible to determine the position of the end-effector. This information is not used to control the system, but is only used to calculate the end-effector position after the experiments. As for the simulations, the motor positions x_1 and y, which are in fact the rotations φ_1 and φ_3, are only used to determine the controller outputs. However, not the measured rotations φ_1 and φ_3 are used as the input for the controller, but the estimated values by the discrete Kalman observer one sample ahead are used, which is also the same as during the simulations. For all experiments the sample frequency is 143 [Hz].

5.3 Experiments

For the x-direction, the first experiments have been carried out with the same controller as during the simulations. These controllers are the PD-controller and the H_∞- and μ-controllers designed with the following weighting functions (set 1, chapter 4)

$$W_1 = \frac{250s^2 + 14400s + 360000}{900s^2 + 250s + 45} \qquad W_2 = \frac{0.15s^2 + 0.14s + 0.02}{0.67s^2 + 20s + 418}$$

For the y-direction the experiments have also been carried out with a PD-controller and an H_∞- and μ-controller designed with the following weighting functions

$$W_1 = \frac{375s^2 + 21600s + 540000}{900s^2 + 250s + 45} \qquad W_2 = \frac{0.0015s}{1+0.02s}$$

The weighting function W_1 for the performance requirements has been chosen in such a way that the bandwidth of the system with the H_∞-controller is about 6 [Hz]. The weighting function W_2 only reflects possible model errors caused by motor dynamics and mass variations. Because the model errors are not exactly known the choice for W_2 is rather arbitrary.

As for the simulations, the controllers have been extended with compensation for the coulomb friction and a feedforward of the desired accelerations. From some experiments it was concluded that in y-direction a friction compensation of 9 [N] instead of 15 [N] leads to better results. The desired trajectory was a circle with the same radius and angular velocity as used in the simulations. The results of the experiments are only given for the H_∞- and PD-controller, because the μ-controllers do not perform well for both the x-direction and the y-direction.

In figures 5.1 and 5.2 the motor position errors are plotted for the x- and y-direction, respectively. The value of the stiffness of the torsion spring is 0.46 [Nm/rad]. Figure 5.3 illustrates the RMS-values of the motorposition error in x-direction for several values of the stiffness.

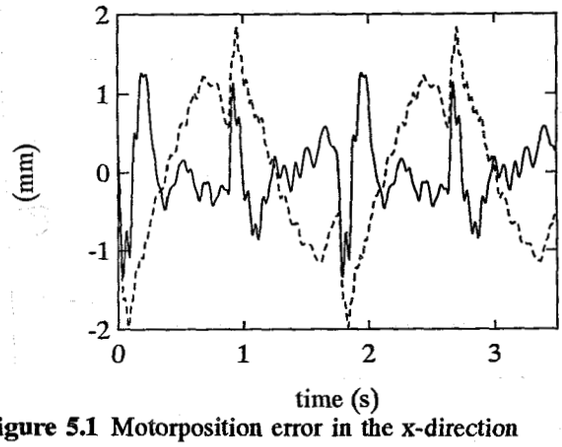


Figure 5.1 Motorposition error in the x-direction
PD-controller (dashed) and H_∞-controller (solid)

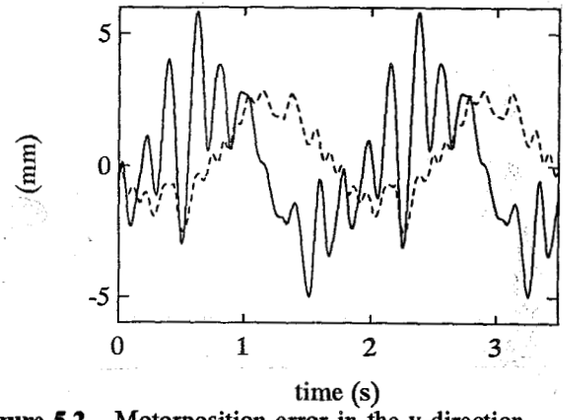


Figure 5.2 Motorposition error in the y-direction

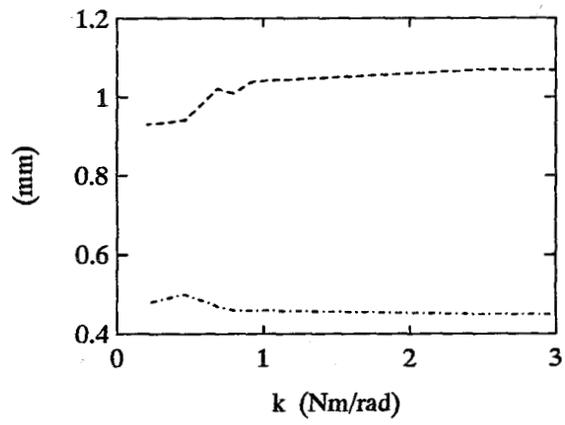


Figure 5.3 RMS-values of motorposition error in
x-direction for several torsion springs
PD-controller (dashed) and H_∞-controller (dot dashed)

Analysis

x-direction: The PD-controller does not perform as well as during simulations. The tracking error of the motorposition has been increased by about 40 percent. Some possible reasons are

- There is no exact compensation of the coulomb friction. In practice the coulomb friction does not only depend on the sign of the speed, so the coulomb friction will differ a lot from the compensated value of the friction for small speeds. Another reason is that the belts often touch the sides of the belt wheels through which the friction will change.
- The parameters, like masses, moments of inertia, etc do not exactly correspond to the model parameters, which have been used at the controller design.
- The real system contains more unmodelled dynamics caused by viscous friction, bad bearings and the little springs which connect the belts to the slides.
- The motors and the amplifier have been modelled as constant gains, which is not correct because of their own dynamics.

Contrary to the simulations the H_∞-controller now performs better than the PD-controller. A possible reason could be the bandwidth of H_∞-controller (about 20 [rad/s], figure 5.4), which is smaller than the bandwidth of the PD-controller (about 40 [rad/s]), so the above mentioned unmodelled dynamics have less influence on the system behaviour. However, to confirm this statement experiments with a PD-controller with a smaller bandwidth will be necessary. It has to be noted that the H_∞-controller even performs a little better as during the simulation, but there is no clear explanation for this. For both, the H_∞-controller and the PD-controller the RMS-value of the motor position error does not increase if the stiffness of the torsion spring decreases, figure 5.3, which is also different from the simulation results, figure 4.17 . From this it can be concluded that in the real system the torsion spring has less influence on the motor position.
The chattering response of the μ-controller during the experiments could point to a too large influence of model errors which are not taken into account during the controller design. However from figure 5.4, it can be concluded that the bandwidth of the μ-controller is about 40 [rad/s], which is equal to the bandwidth of the PD-controller, so the model errors are not the main reason for the bad performance of the μ-controller.
In [1] the estimation errors of the Kalman observer are given as another possible reason for the chattering response. Especially, the speed estimation is bad if the eigenfrequency of the used PD-controller is above 4.5 [Hz], which implies a bandwidth of about 55 [rad/s], and the sample frequency is about 125 [Hz].
However, the estimated speed is not used for the μ-controller, but probably a kind of observer has been included in the μ-controller, which may lead to problems for the determination of the new states of the controller. This has not been proved, but it is likely considering that the H_∞- and μ-controller are calculated by solving two Riccati equations [5]. One equation represents the optimal control problem and the other represents the observer. Although, the bandwidth of the μ-controller is 40 [rad/s], it may be too large for a good determination of the new states of the controller. In the next section another μ-controller is proposed and tested.

y-direction: From figure 5.2 it can be concluded that PD-controller performs better than the H_∞-controller, which means that the PD-controller is more robust to the model errors. To improve the H_∞-controller, the model errors have to be taken into account more accurately during the controller design. For example, the harmonic friction, which is clearly visible in figure 5.3, can be considered as a perturbation on the nominal system and taken into account at the controller design. For the same reasons as mentioned for the x-direction the μ-controller for the y-direction doesn't perform satisfying.

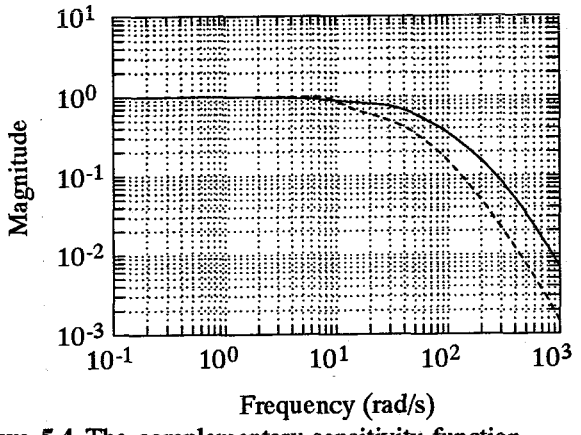


Figure 5.4 The complementary sensitivity function for the H_∞- (dashed) and μ-controller (solid)

5.4 Another H_∞- and μ-controller

As a result of the bad performance of the tested μ-controller in section 5.3 it has been tried to design another μ-controller for the x-direction. For the new controller design the weighting functions are

$$W_1(s) = \frac{125s^2 + 7200s + 180000}{900s^2 + 250s + 45} \qquad W_2(s) = \frac{0.33s^2 \bullet 0.14s + 0.02}{0.67s^2 + 20s + 418}$$

The requirements for the performance have been chosen smaller and the weighting function for the uncertainties has been chosen larger in the hope that the μ-controller performs better. With the above mentioned weighting functions a H_∞-controller has been also designed.

H_∞-controller	6 states	1 input	1 output
μ-controller	14 states	1 input	1 output

In figures 5.5 and 5.6 the μ-values and the complementary sensitivity functions for both controllers are plotted, respectively.

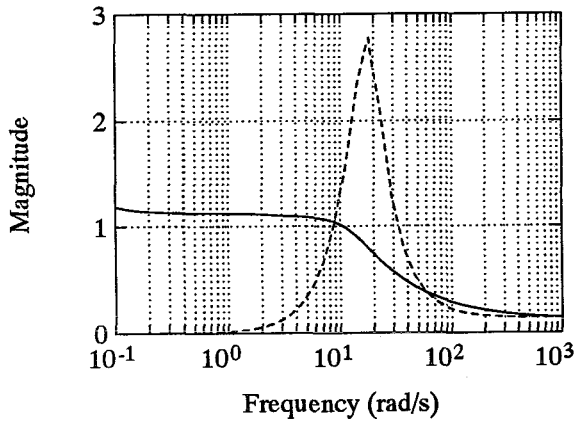


Figure 5.5 μ-values of the H_∞- (dashed) and μ-controller (solid)

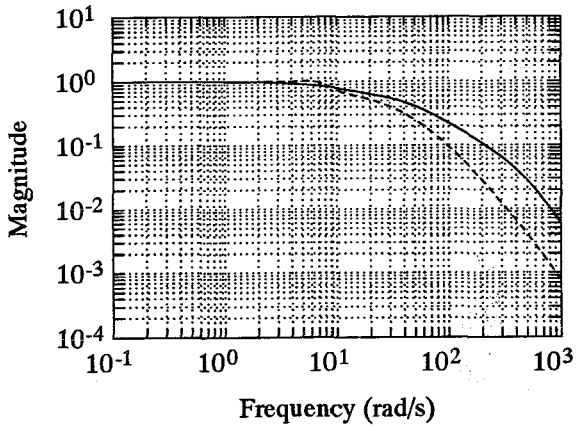


Figure 5.6 Compl. sens. function for H_∞- (dashed) and μ-controller (solid)

The controllers have been tested for two desired trajectories, both circles.

a.

$x_d = 0.2\cos(2\pi ft)$	[m]
$y_d = 0.2\sin(2\pi ft)$	[m]
$f = 4\pi/3.5$	[rad/s]
$0 \le t \le 5.25$	[s]

b. The same trajectory as above with the difference that the frequency now is $2\pi/3.5$ [rad/s].

In figure 5.7 the motorposition error is plotted. The value of the stiffness of the torsion spring was 0.46 [Nm/rad] and the reference signal was signal a. Figures 5.8 and 5.9 illustrate the RMS-values of the motorposition error and the RMS-values of the end-effector position error for several values of the stiffness, respectively. For trajectory b the RMS-values of the motorposition are only plotted, figure 5.10

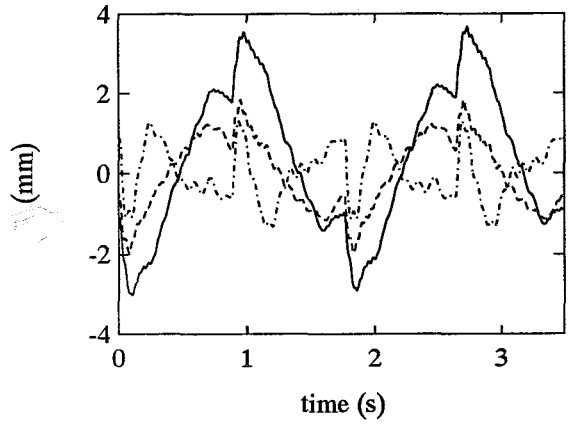


Figure 5.7 Motorposition error in x-direction for trajectory a. and k=0.46 [Nm/rad] μ-controller (solid), PD-controller (dashed) and H_∞-controller (dot dashed)

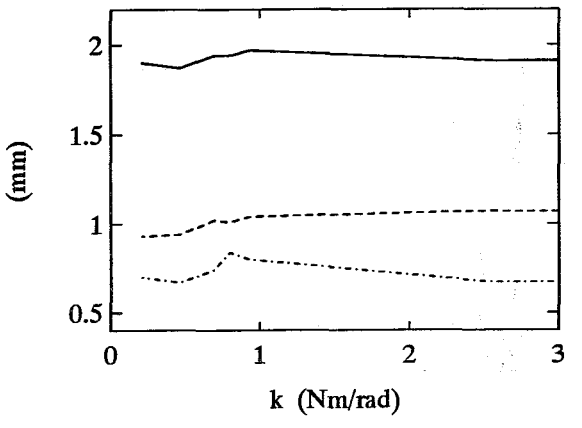


Figure 5.8 RMS-value of motorposition error for trajectory a.

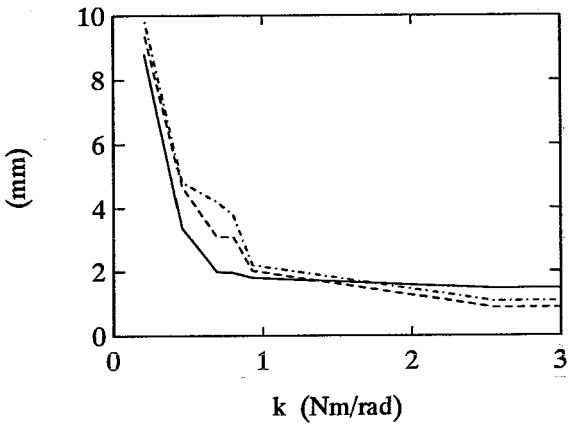


Figure 5.9 RMS-value of the end-effector pos. error for trajectory a.
μ-controller (solid), PD-controller (dashed) and H_∞-controller (dot dashed)

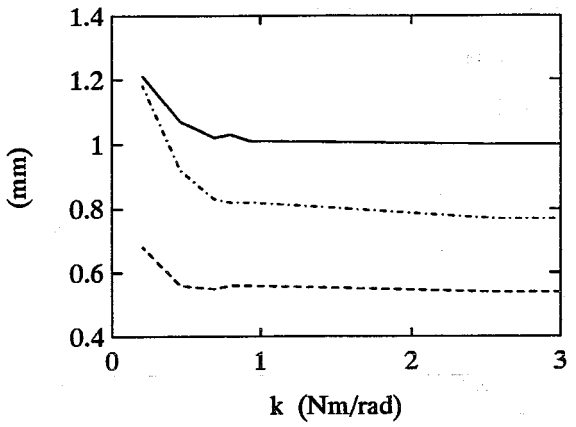


Figure 5.10 RMS-value of motorposition error for trajectory b.

Analysis

Now the μ-controller does not cause a chattering response. However, for both trajectories the H_∞- and PD-controller perform better than the μ-controller. In the case of the H_∞-controller the desired trajectory has less influence on the motorposition error. Comparing figures 5.8 and 5.10 it can be seen that the RMS-values of the motor position error hardly differ for the H_∞-controller. For the PD- and μ-controller these values increase by 50 percent for trajectory b. It has to be noted that for small values of the stiffness the H_∞-controller does even not perform as well as it does for trajectory a, but there is no clear explanation for this. If the desired trajectory is trajectory a. than the stiffness of the torsion spring has less influence on the tracking errors. Even for small values of the stiffness the tracking errors do not increase. This is probably caused by the influence of the other unmodelled dynamics mentioned in the previous section.
In the case of trajectory b. the tracking errors increase if the value of the stiffness decreases. For this trajectory the other unmodelled dynamics probably have less influence on the tracking errors, so the influence of the torsion spring is more important for the system behaviour. The robustness with respect to model errors as a result of the torsion spring is the same for the controllers. From figure 5.9 can be concluded that the controllers have no or less influence on the position error of the end-effector which could be expected because the controllers have only been designed on the motor position error.

5.5 Discussion and conclusions

During the simulations the performance of the μ-controller showed great promise. In all cases the μ-controller performed better than the H_∞- or the PD-controllers and the robustness with respect to model errors did not differ much. However, during the experiments the μ-controller becomes instable if the bandwidth of the controlled system is too large. This can be avoided by choosing other weighting functions for the performance and the uncertainties. Then the response is stable, but not as good as it is for the H_∞- and PD-controllers. The robustness with respect to the torsion spring is less clear during the experiments than it was during the simulations. Only if the frequency of the reference signal is taken smaller than the influence of the torsion spring is visible. In that case the motorposition error increases if the stiffness of the torsion spring is taken smaller than the values which the controllers have been designed for.

Chapter 6. Conclusions and recommendations

In this chapter a summary is given of the most important conclusions of this research. Some recommendations for further research and experiments will also be given.

* The design of the μ-controller sometimes causes problems. The D-K iteration carried out with the help of the software [3] doesn't always converge to the optimal μ-controller, which is often the result of numerical problems. These problems occur if the matrix which reflects the design problem is too large, so it can be ill conditioned. Two possible reasons for this bad condition are:
  - The order of the transfer functions, fitting the D-scales, is chosen too large, so the parameters of this functions nearly depend on each other.
  - The order of the weighting functions which reflect the model errors and the requirements for the system have been chosen too large. This means that the design problem has not been formulated well.

* The method used to determine the additional model errors as a consequence of the not exact state feedback linearization for the RT-robot leads to a reasonable conservative controller. Nevertheless, with regard to parameter variations (variations in the mass) the μ-controller is more robust with respect to these model errors than the PD- or H_∞-controller. In the case of the motor dynamics the PD-controller is a little more robust.
  In this study for the RT-robot model errors as a result of the parameter variations and the unmodelled dynamics have been considered separately. These model errors could be considered together during further research into the μ-control, also refer to [10].

* In the case of the simulations the μ-controller applied to the xy-table, achieves better tracking accuracy, than the PD-controller. The robustness with respect to unmodelled dynamics caused by the torsion spring is the same for both controllers. The tracking errors hardly increase until the torsion spring is chosen smaller than the minimum value, which the controller had been designed for.

* The μ-controller implemented in the real xy-table causes a chattering response. This chattering response is caused by the too large bandwidth of the controlled system, which possibly leads to a bad determination of the new states of the controller. From the experiments it can be concluded that a controller with a smaller bandwidth leads to a stable response. On the other hand the PD- and H_∞-controller perform well during the experiments. However, the stiffness of the torsion spring has no or less influence on the tracking error, because of the influence of other model errors like springs between belts and slides, harmonic friction due to bad bearings, dynamics of the motors and amplifier. It would be useful to design a μ- and H_∞-controller with taken into account the different model errors as a consequence of the above mentioned unmodelled dynamics, so the final controller will perform better. Of course, a good analysis of the possible model errors will be necessary.

* It has to be noticed that a robust controller (μ- or H_∞-controller) has to be insensitive to unmodelled dynamics and parameter variations to some extent. From the experiments and other conclusions it can be concluded that the structure of the model errors has to be determined with a reasonable accuracy to get a robust controller that performs like a PD-controller. This means that some knowledge about the system is required. However, this extra knowledge can also be used to extend the model which involves that less advanced controllers can be used. Anyway, the μ-controller design will be less complicated, so it will possible be better.

* Maybe it would be possible to combine a μ-controller design for the unmodelled dynamics with a adaption law to estimate the parameters of the system.

* Summarizing it can be concluded that in the case of the xy-table the μ-controller does not offer advantages to a simple PD-controller as long as no more model errors caused by unmodelled dynamics are taken into account during the controller design. For large systems the μ-controller may have advantages to the PD-controller because the poles of the controlled system cannot be placed. Of course, also in that case, a good analysis of the possible model errors will be necessary.

References

[1] **Gerwen, van L.J.W.**
An adaptive robot controller: Design, simulation and implementation
WFW report 90.036, Eindhoven University of Technology, 1990

[2] **Doyle, J.C., Francis, B.A., Tannenbaum, A.R.**
Feedback control theory, Macmillan Publishing Company, 1992

[3] **Balas, G.J., Doyle, J.C., Glover, K., Packard, A., Smith, R.**
μ-Analysis and Synthesis Toolbox, (μ-**Tools**), Matlab functions for the analysis and design of robust control systems
MUSYN Inc. and The MathWorks, Inc., April 1, 1991

[4] **Chiang, R.Y., Safonov, M.G.**
Robust-Control Toolbox, User's Guide, The MathWorks, Inc., June 1, 1988

[5] **Doyle, J.C., Glover, K., Khargonekar, P.P., Francis, B.A.**
State-Space Solutions to Standard H_2 and H_∞ Control Problems
IEEE Trans. Auto. Control, vol. 34, no. 8, August 1989

[6] **Doyle, J.C.**
Analysis of feedback systems with structured uncertainties
IEE Proc., vol. 129, Part D, no. 6, pp. 242-250, November 1982

[7] **Steinbuch, M., Schootstra, G., Smit, S.G.**
μ-Synthesis of a flexible mechanical servo system
Nat.Lab. Technical Note Nr. 144/90

[8] **Packard, A., Doyle, J.C.**
Excerpts from a forthcoming book on μ, 1989

[9] **Isidori, A.**
Nonlinear control systems: an introduction, 2nd edition
Berlin: Springer, 1989, 479 p.

[10] **Fan, M.K.H., Tits, A.L., Doyle, J.C.**
Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics
IEEE Trans. Auto. Control, vol. 36, no. 1, January 1991

[11] **Terlouw, J.C.**
Robust analysis of control systems with parametric uncertainty
Nat.Lab. Technical Note Nr. 099/90

Appendix A. The Structured Singular Value

The structured singular value, a matrix function denoted by $\mu(\cdot)$ will be defined in this appendix. For $M \in \mathbb{C}^{nxn}$ it is defined as [6]:

$$\mu_{\Delta_b}(M) := \frac{1}{\min\limits_{\Delta\in\Delta_b} \{\bar{\sigma}(\Delta): \Delta\in\Delta_b,\ \det(I-M\Delta) = 0\}} \tag{A.1}$$

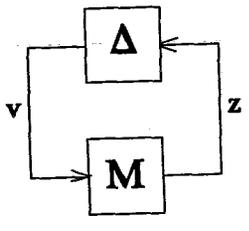
unless no $\Delta \in \Delta_b$ makes I-MΔ singular in which case $\mu_{\Delta_b}(M) := 0$.

There are two types of perturbation blocks, repeated scalar and full blocks. Two nonnegative integers, S and F, represent the number of repeated scalar blocks and the number of full blocks, respectively. The following positive integers are introduced $r_1,...,r_S$; $m_1,...,m_F$. The i'th repeated scalar block is $r_i x r_i$, while the j'th full block is $m_j x m_j$.

Define $\Delta_b \subset \mathbb{C}^{nxn}$ as $\Delta_b = \{\text{diag}\{\delta_1 I_{r_1},...,\delta_S I_{r_S}, \Delta_1,...,\Delta_F\}: \delta_i \in \mathbb{C},\ \Delta_j \in \mathbb{C}^{m_j * m_j}\}$.

Often norm bounded subsets of Δ_b will be important, so define $B\Delta_b = \{\Delta | \Delta \in \Delta_b: \bar{\sigma}(\Delta) \le 1\}$

It is instructive to consider a "feedback" interpretation of $\mu_{\Delta_b}(M)$ at this point. Let $M \in \mathbb{C}^{nxn}$ be given and consider the loop shown below.



As long as I-MΔ is nonsingular, the only solution v, z to the loop equations are v=z=0. However, if I-MΔ is singular, then there are infinitely many solutions. $\mu_{\Delta_b}(M)$ is a measure of the smallest structured Δ that causes "instability" of the constant matrix feedback loop shown above.
It has been proved that for $\mu_{\Delta_b}(M)$ the following inequality is valid

$$\rho(M) \le \mu_{\Delta_b}(M) \le \bar{\sigma}(M) \tag{A.2}$$

with $\rho(M)$: spectral radius of M, $\rho(M) := \max\limits_i |\lambda_i(M)|$, $\lambda_i(M)$ an eigenvalue of M

$\bar{\sigma}(M)$: maximum singular value of M

In [6, 8] a proof of this inequality can be found. Because the gap between the bounds in the

expression can be arbitrarily large, they must be tightened. To do this, first define the following two subsets of $\mathbb{C}^{nxn}$

$$\mathbf{Q} = \{Q \in \Delta_b : Q^*Q = I_n\}$$

$$\mathbf{D} = \{\mathrm{diag}\{D_1,..,D_s,d_1I_{m_1},...,d_{F-1}I_{m_{F-1}},I_{m_F}\}: D_i \in \mathbb{C}^{r_i*r_i}, D_i = D_i^* > 0, d_j \in \mathbb{R}, d_j > 0\}$$

Note that for any $\Delta \in \Delta_b$, $Q \in \mathbf{Q}$, $D \in \mathbf{D}$,

$$Q^* \in \mathbf{Q},\ Q\Delta \in \Delta_b,\ \bar{\sigma}(Q\Delta) = \bar{\sigma}(\Delta Q) = \bar{\sigma}(\Delta),\ D\Delta = \Delta D.$$

Consequently,

Theorem A.1 For all $Q \in \mathbf{Q}$ and $D \in \mathbf{D}$

$$\mu_{\Delta_b}(MQ) = \mu_{\Delta_b}(QM) = \mu_{\Delta_b}(M) = \mu_{\Delta_b}(DMD^{-1})$$

Refer to [6,8] for a proof of this theorem.
The bounds in (A.2) can now be tightened to

$$\max_{Q \in \mathbf{Q}} \rho(QM) = \max_{\Delta \in B\Delta_b} \rho(\Delta M) = \mu_{\Delta_b}(M) \le \inf_{D \in \mathbf{D}} \bar{\sigma}(DMD^{-1}) \qquad (A.3)$$

The lower bound is always an equality [8]. Unfortunately, the quantity $\rho(QM)$ can have multiple local maxima which are not global. The μ-software [3] uses a slightly different formulation of the lower bound as a power algorithm, which usually is an effective method to compute μ.
The upperbound has just one global minimum, but this minimum can be too conservative. This means that the upperbound in not always equal to μ. For some block structures the upperbound is always equal to $\mu_{\Delta_b}(M)$. If $2S+F \le 3$ the upperbound is equal to $\mu_{\Delta_b}(M)$. If $2S+F > 3$ then an exact value cannot be always computed. Summarized:

F= / S=	0	1	2	3	4
0		yes	yes	yes	no
1	yes	yes	no	no	no
2	no	no	no	no	no

yes = μ can be computed exact
no = μ can be only approximated
For proofs of the different boxes of this table refer to [8].

Appendix B. The state feedback linearization for the RT-robot

In this appendix the exact linearization of the RT-robot is considered. The model equations (3.1) have to be written in the following form.

$$\dot{x} = f(x) + g(x)u \qquad (B.1)$$
$$y = h(x)$$

with

$$\begin{bmatrix} x_1 = r(t) \\ x_2 = \dot{r}(t) \\ x_3 = \varphi(t) \\ x_4 = \dot{\varphi}(t) \end{bmatrix}, \quad f(x) = \begin{bmatrix} x_2 \\ \dfrac{(P_1x_1 - P_2)x_4^2}{P_1} \\ x_4 \\ \dfrac{-2(P_1x_1 - P_2)x_2x_4}{P_3 - 2P_2x_1 + P_1x_1^2} \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 & 0 \\ \dfrac{1}{P_1} & 0 \\ 0 & 0 \\ 0 & \dfrac{1}{P_3 - 2P_2x_1 + P_1x_1^2} \end{bmatrix}, \quad y = \begin{bmatrix} r(t) \\ \varphi(t) \end{bmatrix}, \quad u = \begin{bmatrix} F(t) \\ M(t) \end{bmatrix}$$

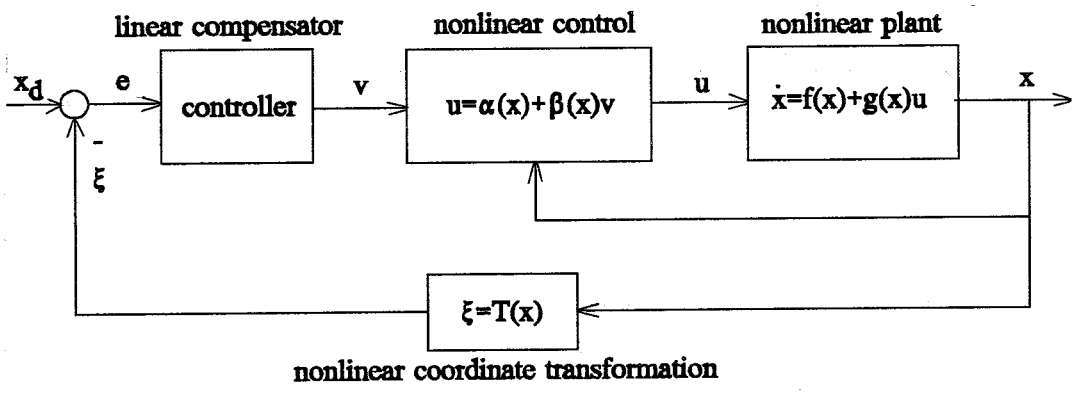


Figure B1 The feedback linearization.

The exact feedback linearization is defined as

$$u = \alpha(x) + \beta(x)v \qquad \text{with} \quad \alpha(x) = -A^{-1}(x)b(x), \quad \beta(x) = A^{-1}(x) \qquad (B.2)$$

The matrices are defined as

$$A(x) = \begin{bmatrix} L_{g_1}L_f^{r_1-1}h_1(x) & \cdots & L_{g_m}L_f^{r_1-1}h_1(x) \\ L_{g_1}L_f^{r_2-1}h_2(x) & \cdots & L_{g_m}L_f^{r_2-1}h_2(x) \\ \cdots & \ddots & \cdots \\ L_{g_1}L_f^{r_m-1}h_m(x) & \cdots & L_{g_m}L_f^{r_m-1}h_m(x) \end{bmatrix} \qquad b(x) = \begin{bmatrix} L_f^{r_1}h_1(x) \\ L_f^{r_2}h_2(x) \\ \cdots \\ L_f^{r_m}h_m(x) \end{bmatrix}$$

$r_1,...,r_m$ are elements of the vector relative degree. A multivariable nonlinear system of the form (B.1) has a (vector) relative degree $\{r_1,...,r_m\}$ at point x^0 if

1. $L_{g_j}L_f^k h_i(x) = 0$ for all $1 \le j \le m$, for all $1 \le i \le m$, for all $k \le r_i-2$ and for all x in a neighbourhood of x^0. Here $L_f^k h_i(x)$ means the k'th successive Lie derivation of the scalar function $h_i(x)$ in the direction of the vector field f.
2. the mxm matrix A(x) is nonsingular at $x = x^0$.

For the linearization the equations have to be written in the normal form. So the linearizing coordinates are defined as

$$\xi_k^{\,i}(x) = L_f^{k-1}h_i(x) \quad \text{for } 1 \le k \le r_i,\ 1 \le i \le m. \tag{B.3}$$

In the case of the RT-robot the relative degree is {2,2}. The equations already are in the normal form, so

$$\xi_1 = x_1\ ;\ \xi_2 = x_2\ ;\ \xi_3 = x_3\ ;\ \xi_4 = x_4$$

The matrices A(x) and b(x) are

$$A(x) = \begin{bmatrix} \frac{1}{P_1} & 0 \\ 0 & \frac{1}{P_3-2P_2x_1+P_1x_1^2} \end{bmatrix} \qquad b(x) = \begin{bmatrix} \frac{(P_1x_1-P_2)x_4^2}{P_1} \\ \frac{-2(P_1x_1-P_2)x_2x_4}{P_3-2P_2x_1+P_1x_1^2} \end{bmatrix}$$

The linearizing feedback becomes

$$u = \begin{bmatrix} -(P_1x_1-P_2)x_4^2+P_1v_1 \\ 2(P_1x_1-P_2)x_2x_4+(P_3-2P_2x_1+P_1x_1^2)v_2 \end{bmatrix} \tag{B.4}$$

In the new coordinates, the system appears as

$$\dot{\xi} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} v \tag{B.5}$$

The state linearization feedback results in a decoupled system of two second order differential equations. For more about the state feedback linearization refer to [9].

Appendix C. General interconnection structure for the RT-robot

In this appendix the uncertainties in the system matrix A and in the input matrix B as consequences of the not exact feedback linearization are derived for the RT-robot. This is done for two types of model errors, i.e., variations in the load mass m_1 and motor dynamics. Also the weighting functions which reflect the model errors are determined. First, the influence of variations in the mass m_1 are considered.
The model equations 3.1 are written in the following form, see also equations (B.1)

$$\dot{x} = f(x) + g(x)u \qquad (C.1)$$
$$y = h(x)$$

with

$$\begin{bmatrix} x_1 = r(t) \\ x_2 = \dot{r}(t) \\ x_3 = \varphi(t) \\ x_4 = \dot{\varphi}(t) \end{bmatrix}, \quad f(x) = \begin{bmatrix} x_2 \\ \dfrac{(P_1x_1 - P_2)x_4^2}{P_1} \\ x_4 \\ \dfrac{-2(P_1x_1 - P_2)x_2x_4}{P_3 - 2P_2x_1 + P_1x_1^2} \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 & 0 \\ \dfrac{1}{P_1} & 0 \\ 0 & 0 \\ 0 & \dfrac{1}{P_3 - 2P_2x_1 + P_1x_1^2} \end{bmatrix}, \quad y = \begin{bmatrix} r(t) \\ \varphi(t) \end{bmatrix}, \quad u = \begin{bmatrix} F(t) \\ M(t) \end{bmatrix}$$

The exact linearizing feedback has been derived in appendix B, equation (B.4), and is as follows

$$u = \begin{bmatrix} -(\bar{P}_1x_1 - \bar{P}_2)x_4^2 + \bar{P}_1v_1 \\ 2(\bar{P}_1x_1 - \bar{P}_2)x_2x_4 + (\bar{P}_3 - 2\bar{P}_2x_1 + \bar{P}_1x_1^2)v_2 \end{bmatrix} \qquad (C.2)$$

with $\quad \bar{P}_1 = m + \bar{m}_1 = 15\ [kg]\ ;\ \bar{P}_2 = \frac{1}{2}ml = 5\ [kgm]\ ;\ \bar{P}_3 = I + \frac{1}{3}ml^2 = 8\frac{1}{3}\ [kgm^2]$

Variations in the mass m_1 only have influence on P_1, so P_2 and P_3 are equal to the nominal values. For this reason the bars above P_2 and P_3 are left away during the derivation.
Substitution of the linearization feedback (C.2) in (C.1) results in

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \dfrac{(P_1 - \bar{P}_1)}{P_1}x_1x_4^2 + \dfrac{\bar{P}_1}{P_1}v_1 \\ x_4 \\ \dfrac{2(\bar{P}_1 - P_1)x_1x_2x_4}{(P_3 - 2P_2x_1 + P_1x_1^2)} + \dfrac{P_3 - 2P_2x_1 + \bar{P}_1x_1^2}{(P_3 - 2P_2x_1 + P_1x_1^2)}v_2 \end{bmatrix} \qquad (C.3)$$

To obtain a linear model description which is required for the μ-synthesis the equations (C.3) have to be linearized along a desired trajectory. The equations are written in the following form

$$\dot{x}(t) = f(x(t), v(t), t) \tag{C.4}$$

with a nominal input $v^0(t:t_0,t_e)$ and a nominal trajectory $x^0(t)$ (C.4) becomes

$$\dot{x}^0(t) = f(x^0(t), v^0(t), t) \quad \text{with } t_0 \le t \le t_e \tag{C.5}$$

Perturbations on the nominal equations can be written as

$$\dot{x}^0(t) + \tilde{x}(t) = f(x^0(t)+\tilde{x}(t), v^0(t)+\tilde{v}(t),t) \tag{C.6}$$

This equation can be approximated by a Taylor polynomial

$$\dot{x}^0(t) + \tilde{x}(t) = f(x^0(t), v^0(t), t) + J_{fx}(x^0(t), v^0(t), t)\tilde{x}(t) + J_{fu}(x^0(t), v^0(t), t)\tilde{v}(t) +... \tag{C.7}$$

The matrices J_{fx} and J_{fu} are:

(**Remark:** Unfortunately, two terms have been left out during the derivation of the matrices J_{fx} and J_{fu}. These terms are represented in italics and have been left out of consideration in the continuation of the derivation.)

$$J_{fx} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_0 & 0 & 0 & a_1 \\ 0 & 0 & 0 & 1 \\ a_2 & a_3 & 0 & a_4 \end{bmatrix} \qquad J_{fu} = \begin{bmatrix} 0 & 0 \\ a_5 & 0 \\ 0 & 0 \\ 0 & a_6 \end{bmatrix}$$

with

$$a_0 = \frac{P_1-\bar{P}_1}{P_1}(x_4^0)^2$$

$$a_1 = \frac{2(P_1-\bar{P}_1)}{P_1}x_1^0 x_4^0$$

$$a_2 = \frac{(2(\bar{P}_1-P_1)x_2^0x_4^0)(P_3-2P_2x_1^0+P_1(x_1^0)^2)-2(\bar{P}_1-P_1)x_1^0x_2^0x_4^0(-2P_2+2P_1x_1^0)}{(P_3-2P_2x_1^0+P_1(x_1^0)^2)^2} +$$

$$+ \frac{2(\bar{P}_1-P_1)P_3x_1^0-4(\bar{P}_1-P_1)(x_1^0)^2P_2}{(P_3-2P_2x_1^0+P_1(x_1^0)^2)^2}v_2^0$$

$$a_3 = \frac{2(\bar{P}_1 - P_1)x_1^0 x_4^0}{P_3 - 2P_2 x_1^0 + P_1 (x_1^0)^2}$$

$$a_4 = \frac{2(\bar{P}_1 - P_1)x_1^0 x_2^0}{P_3 - 2P_2 x_1^0 + P_1 (x_1^0)^2}$$

$$a_5 = \frac{\bar{P}_1}{P_1}$$

$$a_6 = \frac{P_3 - 2P_2 x_1^0 + \bar{P}_1 (x_1^0)^2}{P_3 - 2P_2 x_1^0 + P_1 (x_1^0)^2}$$

If the desired trajectory is as follows

$$\begin{aligned} r &= x_1^0 = 0.5 + 0.5\sin(0.5\pi t) && [m] \\ \dot{r} &= x_2^0 = 0.25\pi\cos(0.5\pi t) && [m/s] \\ \varphi &= x_3^0 = \sin(0.5\pi t) && [rad] \\ \dot{\varphi} &= x_4^0 = 0.5\pi\cos(0.5\pi t) && [rad/s] \end{aligned} \qquad (C.9)$$

and

$14 \le P_1 \le 16$ [kg], which means $4 \le m_1 \le 6$ [kg]

Define $\Delta P_1 = \bar{P}_1 - P_1$. The parameters $a_1 ... a_6$ can be approximated by the following equations

$a_1 \approx 0.13\Delta P_1 x_4^0$; $a_2 \approx 0.31\Delta P_1 x_4^0 x_2^0$; $a_3 \approx -0.19\Delta P_1 x_4^0$;

$a_4 \approx -0.093\Delta P_1 x_4^0$; $a_5 \approx 1 - 0.072\Delta P_1$; $a_6 \approx 1 - 0.081\Delta P_1$;

It has to be noticed that the scalars $a_1 ... a_6$ depends on each other. Of course, this dependence can be taken into account. The following scalars are defined:

$b_1 = \Delta P_1$; $b_2 = \Delta P_1 x_4^0$; $b_3 = \Delta P_1 x_4^0 x_2^0$

The matrices J_{fx} and J_{fu} now become

$$J_{fx} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.13b_2 \\ 0 & 0 & 0 & 1 \\ 0.31b_3 & -0.19b_2 & 0 & -0.093b_2 \end{bmatrix} \qquad J_{fu} = \begin{bmatrix} 0 & 0 \\ 1-0.071b_1 & 0 \\ 0 & 0 \\ 0 & 1-0.081b_1 \end{bmatrix}$$

J_{fx} and J_{fu} are equal to the perturbed system matrix A and the perturbed input matrix B, respectively. With the help of the method described in section 3.3 the matrices B_2, C_2, D_{12}, D_{21}, D_{22} and Δ of the general interconnection structure can be determined. However, the determination of these matrices is not unequivocal, refer to [7].

b_2-variations Δb_2:

$$dA = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.13\Delta b_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.19\Delta b_2 & -0.093\Delta b_2 \end{bmatrix} \Rightarrow B_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Delta = \begin{bmatrix} 0.13\Delta b_2 & 0 \\ 0 & 0.63\Delta b_2 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -0.3 & 0 & -0.065 \end{bmatrix}$$

b_1-variations Δb_1:

$$dB = \begin{bmatrix} 0 & 0 \\ -0.071\Delta b_1 & 0 \\ 0 & 0 \\ 0 & -0.081\Delta b_1 \end{bmatrix} \Rightarrow B_2 = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, \quad \Delta = \begin{bmatrix} 0.071\Delta b_1 & 0 \\ 0 & 0.081\Delta b_1 \end{bmatrix}, \quad D_{21} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

b_3-variations Δb_3:

$$dA = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.31\Delta b_3 & 0 & 0 & 0 \end{bmatrix} \Rightarrow B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \Delta = [\Delta b_3], \quad C_2 = [1 \ 0 \ 0 \ 0]$$

The whole perturbation matrix is $\Delta = \text{diag}(0.13\Delta b_2, 0.63\Delta b_2, 0.071\Delta b_1, 0.081\Delta b_1, 0.31\Delta b_3)$. The matrices of the interconnection structure matrix G, figure 2.1, are

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -0.071 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0.081 & 1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & -0.3 & 0 & -0.065 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad D_{21} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$D_{12} = D_{22} = \text{zeros}(6,6)$

The weighting function for the uncertainty is now obtained by scaling the perturbation matrix to unity and is as follows

$$W_2 = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.071 & 0 & 0 \\ 0 & 0 & 0 & 0.081 & 0 \\ 0 & 0 & 0 & 0 & 0.38 \end{bmatrix} \tag{C.10}$$

The matrices B_2, C_2, D_{12}, D_{21}, D_{22} and the weighting function W_2 are elements of the general interconnection matrix G, figure 2.1, which is used for the controller design in chapter 3.

In the case of the motor dynamics the equations (C.1) are first linearized along the desired trajectory and then transformed to the frequency domain. This is also done for the exact linearizating feedback (C.2). The first linearized equation of motion (3.1) now becomes

$$s^2 r(s) = (x_4^0)^2 r(s) + 2\left(\left(\frac{P_1 x_1^0 - P_2}{P_1} x_4^0\right) s\varphi(s) + \frac{1}{P_1} u_1^*(s)\right) \tag{C.11}$$

The exact linearization becomes

$$u_1(s) = -P_1 (x_4^0)^2 r(s) - 2(P_1 x_1^0 - P_2) x_4^0 s\varphi(s) + P_1 v_1(s) \tag{C.12}$$

Laplace transform of the motor dynamics, equation 3.7), gives the following equation

$$u_1^* = \frac{1}{(1+\tau_f s)} u_1(s) \tag{C.13}$$

Substitution of (C.12) in (C.13) results in

$$u_1^*(s) = \frac{-P_1 (x_4^0)^2}{1+\tau_f s} r(s) - \frac{2(P_1 x_1^0 - P_2) x_4^0 s}{1+\tau_f s} \varphi(s) + \frac{P_1}{1+\tau_f s} v_1(s) \tag{C.14}$$

This expression has to be substituted in (C.11)

$$s^2 r(s) = \left((x_4^0)^2 - \frac{(x_4^0)^2}{1+\tau_f s}\right) r(s) + \left(2\left(\frac{P_1 x_1^0 - P_2}{P-1} x_4^0\right) - 2\frac{(P_1 x_1^0 - P_2)}{P_1(1+\tau_f s)} x_4^0\right) s\varphi(s) + \frac{1}{1+\tau_f s} v_1(s) \tag{C.15}$$

If the feedback linearization is exact the equation above will be

$$s^2 r(s) = v_1(s) \tag{C.16}$$

The equation above is the result of the Laplace transform of equation (B.5). Now (C.16) has to be subtracted from (C.15) and results in the parameter variations dA(s) and dB(s) caused by the motor dynamics with time constant τ_f

$$dA(s) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{2.56\tau_f s}{1+\tau_f s} & 0 & 0 & \frac{2.13\tau_f s}{1+\tau_f s} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad dB(s) = \begin{bmatrix} 0 & 0 \\ \frac{-\tau_f s}{1+\tau_f s} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The matrices B_2, C_2, D_{12}, D_{21}, D_{22} and Δ of the general interconnection structure can be determined with the help of the method described in section 3.3.

$$B_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \Delta = \begin{bmatrix} \frac{\tau_f s}{1+\tau_f s} \end{bmatrix}, \quad C_2 = \begin{bmatrix} 2.56 & 0 & 0 & 2.13 \end{bmatrix}, \quad D_{21} = \begin{bmatrix} -1 & 0 \end{bmatrix}$$

For the second equation of motion the derivation can be done in the same way. This results in the parameter variations dA(s) and dB(s) caused by the motor dynamics with time constant τ_m

$$dA(s) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{7\tau_m s}{1+\tau_m s} & \frac{2\tau_m s}{1+\tau_m s} & 0 & \frac{\tau_m s}{1+\tau_m s} \end{bmatrix} \qquad dB(s) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{-\tau_m s}{1+\tau_m s} \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \Delta = \begin{bmatrix} \frac{\tau_m s}{1+\tau_m s} \end{bmatrix}, \quad C_2 = \begin{bmatrix} 7 & 2 & 0 & 1 \end{bmatrix}, \quad D_{21} = \begin{bmatrix} 0 & -1 \end{bmatrix}$$

The whole perturbation matrix is

$$\Delta = \begin{bmatrix} \dfrac{\tau_f s}{1+\tau_f s} & 0 \\ 0 & \dfrac{\tau_m s}{1+\tau_m s} \end{bmatrix}$$

The matrices of the interconnection structure matrix G are

$$B_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 2.56 & 0 & 0 & 2.13 \\ 7 & 2 & 0 & 1 \end{bmatrix}, \quad D_{21} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad D_{12} = D_{22} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The weighting function for the uncertainty is now obtained by scaling the perturbation matrix to unity and is as follows

$$W_2 = \begin{bmatrix} \dfrac{\tau_f s}{1+\tau_f s} & 0 \\ 0 & \dfrac{\tau_m s}{1+\tau_m s} \end{bmatrix} \tag{C.17}$$

The matrices B_2, C_2, D_{12}, D_{21}, D_{22} and the weighting function W_2 are elements of the general system G (figure 2.1), which is used for the controller design in chapter 3.

Appendix D. Derivation of the xy-table model

In this appendix the equations of motion of the xy-table model are given together with all values of the parameters which are used for the simulations. In figure D1 the system used for the simulations is shown. This figure is the same as figure 4.1b in chapter 4.

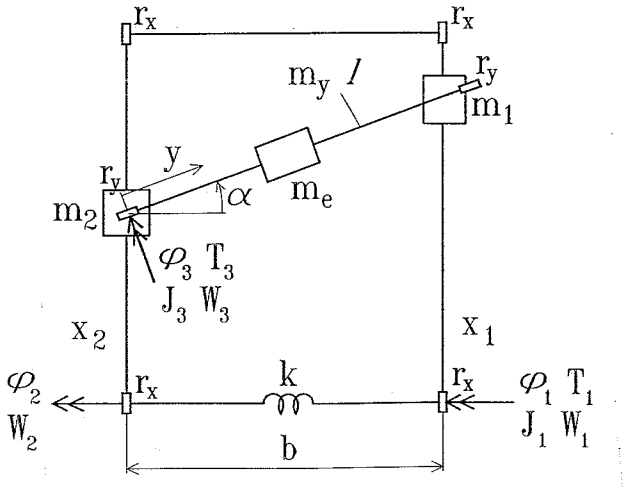


Figure D1 The simulated system.

The used symbols are:

φ_1	angular displacement of belt wheel 1	[rad]
φ_2	angular displacement of belt wheel 2	[rad]
φ_3	angular displacement of belt wheel 3	[rad]
x_1	position of x-slide 1 on slideway 1	[m]
x_2	position of x-slide 2 on slideway 2	[m]
y	position of the end-effector on the y-slideway	[m]
b	distance between slideway 1 and 2	[m]
l	length of the y-slideway	[m]
r_x	radius of the belt wheels 1 and 2	[m]
r_y	radius of belt wheel 3	[m]
m_1	mass of x-slide 1	[kg]
m_2	mass of x-slide 2	[kg]
m_e	mass of the end-effector	[kg]
m_y	mass of the y-slideway including the y-motor	[kg]
J_1	moment of inertia associated with φ_1	[kgm^2]
J_2	moment of inertia associated with φ_2	[kgm^2]
J_3	moment of inertia associated with φ_3	[kgm^2]
W_1	friction torque associated with φ_1	[Nm]
W_2	friction torque associated with φ_2	[Nm]
W_3	friction torque associated with φ_3	[Nm]
T_1	motor torque on belt wheel 1	[Nm]
T_3	motor torque on belt wheel 3	[Nm]

k torsion spring constant [Nm/rad]

$$x_1 = \varphi_1 r_x$$
$$x_2 = \varphi_2 r_x$$
$$y = \varphi_3 r_y$$
$$\alpha = \arctan(\frac{x_1 - x_2}{b})$$

The angle α is assumed to be small: $\alpha \approx \frac{x_1 - x_2}{b}$; $\cos(\alpha) \approx 1$; $\sin(\alpha) \approx \alpha$

The values of the parameters used in the simulations correspond to the values of the parameters of the xy-table. These parameters are:

$b = l = 1$	[m]
$r_x = r_y = 0.01$	[m]
$m_1 = m_2 = 3.8$	[kg]
$m_e = 2.3$	[kg]
$m_y = 6.6$	[kg]
$J_1 = 3*10^{-3}$	[kgm^2]
$J_2 = 0$	[kgm^2]
$J_3 = 2*10^{-4}$	[kgm^2]
$W_1 = W_2 = 0.25$	[Nm]
$W_3 = 0.15$	[Nm]
$k \in \{0.46...\infty\}$	[Nm/rad]

The equations of motion used for simulating the xy-table are [1]

$$M(q)\ddot{q} + h(q,\dot{q}) = f \qquad (D.1)$$

with

$$q^T = [\varphi_1 \ \varphi_2 \ \varphi_3]$$

$$M_{11} = J_1 + [m_1 + \frac{1}{3}m_y(\frac{l}{b})^2 + m_e(\frac{\varphi_3 r_y}{b})^2]r_x^2$$

$$M_{12} = M_{21} = [\frac{1}{2}m_y(\frac{l}{b}) - \frac{1}{3}m_y(\frac{l}{b})^2 + m_e(\frac{\varphi_3 r_y}{b}) - m_e(\frac{\varphi_3 r_y}{b})^2]r_x^2$$

$$M_{13} = M_{31} = 0$$

$$M_{22} = [m_2 + m_y - m_y(\frac{l}{b}) + \frac{1}{3}m_y(\frac{l}{b})^2 + m_e - 2m_e(\frac{\varphi_3 r_y}{b}) + m_e(\frac{\varphi_3 r_y}{b})^2]r_x^2$$

$$M_{23} = M_{32} = [m_e \frac{(\varphi_1 - \varphi_2) r_y}{b}] r_x^2$$

$$M_{33} = J_3 + m_e r_y^2$$

$$h_1 = 2m_e(\varphi_3 r_y)\frac{r_y r_x^2}{b^2}(\dot{\varphi}_1 - \dot{\varphi}_2)\dot{\varphi}_3 + k(\varphi_1 - \varphi_2)$$

$$h_2 = 2m_e(b - \varphi_3 r_y)\frac{r_y r_x^2}{b^2}(\dot{\varphi}_1 - \dot{\varphi}_2)\dot{\varphi}_3 - k(\varphi_1 - \varphi_2)$$

$$h_3 = -m_e(\varphi_3 r_y)\frac{r_y r_x^2}{b^2}(\dot{\varphi}_1 - \dot{\varphi}_2)^2$$

$$f_1 = T_1 - W_1 \text{sign}(\dot{\varphi}_1)$$
$$f_2 = -W_2 \text{sign}(\dot{\varphi}_2)$$
$$f_3 = T_3 - W_3 \text{sign}(\dot{\varphi}_3)$$

For the controller design a simplified model is used. In this model the torsion spring is rigid ($k=\infty$) which means that φ_1 is equal to φ_2. The equations of motion now become:

$$M(q)\ddot{q} + w(\dot{q}) = \tau \qquad (D.2)$$

with

$$q^T = [\varphi_1 \ \varphi_3]$$

$$M_{11} = J_1 + [m_1 + m_2 + m_e + m_y] r_x^2$$
$$M_{12} = M_{21} = 0$$
$$M_{22} = J_3 + m_e r_y^2$$
$$w_1 = (W_1 + W_2)\text{sign}(\dot{\varphi}_1)$$
$$w_2 = W_3 \text{sign}(\dot{\varphi}_3)$$
$$\tau_1 = T_1$$
$$\tau_2 = T_3$$

The values of the parameters used in chapter 4 are:

$P_1 = M_{11}$	$= 4.68*10^{-3}$	$[kgm^2]$
$P_2 = M_{22}$	$= 4.60*10^{-4}$	$[kgm^2]$
$P_3 = W_1 + W_2$	$= 0.50$	$[Nm]$
$P_4 = W_3$	$= 0.15$	$[Nm]$

Appendix E. Linear model of the flexible xy-table

The μ-synthesis requires a linear model of the xy-table. A linear model has been already derived in appendix D, equations (D.2). This model, which is only valid if torsion spring is stiff, will be used for the controller design. To investigate the influence of the torsion spring in this appendix a linear model is derived without neglecting the torsion spring. The nonlinear terms of the equations of motion (D.1) are estimated or neglected if these are small. To estimate the nonlinear terms the desired trajectory has to be known. The desired trajectory is the same trajectory which is used for the experiments. The definition of the axis is given in figure E.1.

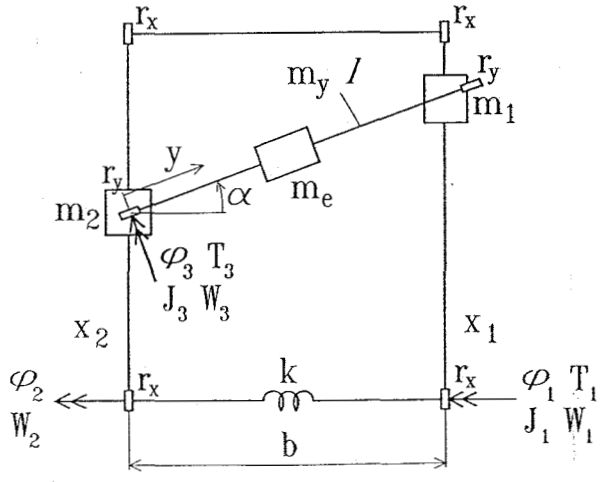


Figure E1 System used for the simulations of the xy-table.

$$\begin{aligned} x_d &= 0.5 - 0.2\cos(2\pi ft) && [m] \\ y_d &= 0.5 + 0.2\sin(2\pi ft) && [m] \\ f &= 3.5\pi/4 && [rad/s] \end{aligned} \qquad (E.1)$$

x_d and y_d are the desired motorpositions of the servomotors. The relationships between these displacements and rotations in the model (D.1) are:

$$\begin{aligned} \varphi_1 &= x_d/r_x && [rad] \\ \varphi_3 &= y_d/r_y && [rad] \end{aligned}$$

The linear equations of motion for this trajectory become

$$\tilde{M}(q)\ddot{q} + \tilde{h}(q,\dot{q}) = \tilde{f} \qquad (E.2)$$

with

$$q^T = [\varphi_1 \ \varphi_2 \ \varphi_3]$$

$$\tilde{M}_{11} = J_1 + [m_1 + \frac{1}{3}m_y(\frac{l}{b})^2]r_x^2$$

$$\tilde{M}_{12} = \tilde{M}_{21} = [\frac{1}{2}m_y(\frac{l}{b}) - \frac{1}{3}m_y(\frac{l}{b})^2 + 0.22m_e]r_x^2$$

$$\tilde{M}_{13} = \tilde{M}_{31} = 0$$

$$\tilde{M}_{22} = [m_2 + m_y - m_y(\frac{l}{b}) + \frac{1}{3}m_y(\frac{l}{b})^2 + m_e - 0.75m_e]r_x^2$$

$$\tilde{M}_{23} = \tilde{M}_{32} = 0$$

$$\tilde{M}_{33} = J_3 + m_e r_y^2$$

$$\tilde{h}_1 = k(\varphi_1 - \varphi_2)$$

$$\tilde{h}_2 = -k(\varphi_1 - \varphi_2)$$

$$\tilde{h}_3 = 0$$

$$\tilde{f}_1 = T_1$$

$$\tilde{f}_2 = 0$$

$$\tilde{f}_3 = T_3$$

The friction torques W_1, W_2 and W_3 have been omitted because the system without friction will be used for the controller design. After the substitution of the values of the parameters the linear equations of motion are

$$3.6*10^{-3}\ddot{\varphi}_1 + 0.17*10^{-3}\ddot{\varphi}_2 + k(\varphi_1-\varphi_2) = T_1 \quad \text{(E.3)}$$

$$0.17*10^{-3}\ddot{\varphi}_1 + 0.67*10^{-3}\ddot{\varphi}_2 - k(\varphi_1-\varphi_2) = 0 \quad \text{(E.4)}$$

$$0.46*10^{-3}\ddot{\varphi}_3 = T_3 \quad \text{(E.5)}$$

Because the torsion spring only has influence on the motion in the x-direction, the equations (E.3) and (E.4) only have to be considered.
These equations are extended with small terms for the viscous friction. The viscous friction has not been modelled, but in the real system this friction is about 0.6 [Ns/m] = $0.6*10^{-4}$ [Nms/rad]. The model equations with viscous friction are

$$3.6*10^{-3}\ddot{\varphi}_1 + 0.17*10^{-3}\ddot{\varphi}_2 + 3*10^{-5}\dot{\varphi}_1 + k(\varphi_1-\varphi_2) = T_1 \quad \text{(E.6)}$$

$$0.17*10^{-3}\ddot{\varphi}_1 + 0.67*10^{-3}\ddot{\varphi}_2 + 3*10^{-5}\dot{\varphi}_2 - k(\varphi_1-\varphi_2) = 0 \quad \text{(E.7)}$$

These equations have to be transformed to the frequency domain. After this transformation $\varphi_2(s)$ can be eliminated, so a transfer function from T_1 to φ_1 is obtained. Laplace transformation of the equations (E.6) and (E.7) gives

$$3.6*10^{-3}s^2\varphi_1(s) + 0.17*10^{-3}s^2\varphi_2(s) + 3*10^{-5}s\varphi_1(s) + k(\varphi_1(s)-\varphi_2(s)) = T_1(s) \quad \text{(E.8)}$$

$$0.17*10^{-3}s^2\varphi_1(s) + 0.67*10^{-3}s^2\varphi_2(s) + 3*10^{-5}s\varphi_2(s) - k(\varphi_1(s)-\varphi_2(s)) = 0 \quad \text{(E.9)}$$

Equation (E.9) can be written as follows

$$\varphi_2(s) = \frac{-0.17*10^{-3}s^2+k}{0.67*10^{-3}s^2+3*10^{-5}s+k}\varphi_1(s) \quad \text{(E.10)}$$

Substitution of (E.10) in (E.8) gives

$$\frac{2.38*10^{-6}s^4 + 1.28*10^{-7}s^3 + 4.61*10^{-3}ks^2 + 6*10^{-5}ks}{0.67*10^{-3}s^2 + 3*10^{-5}s + k}\varphi_1(s) = T_1(s) \quad \text{(E.11)}$$

The transferfunction from T_1 to φ_1 is now

$$\tilde{V}(s) = \frac{0.67*10^{-3}s^2 + 3*10^{-5}s + k}{2.38*10^{-6}s^4 + 1.28*10^{-7}s^3 + 4.61*10^{-3}ks^2 + 6*10^{-5}ks} \quad \text{(E.12)}$$

In the rigid case, equations (D.2), the transfer function from T_1 to φ_1 is

$$V(s) = \frac{1}{P_1}*\frac{1}{s^2} = \frac{213.7}{s^2} \quad \text{(E.13)}$$

With the help of these equations the model errors as a consequence of the torsion spring can be determined. This has been done in chapter 4.