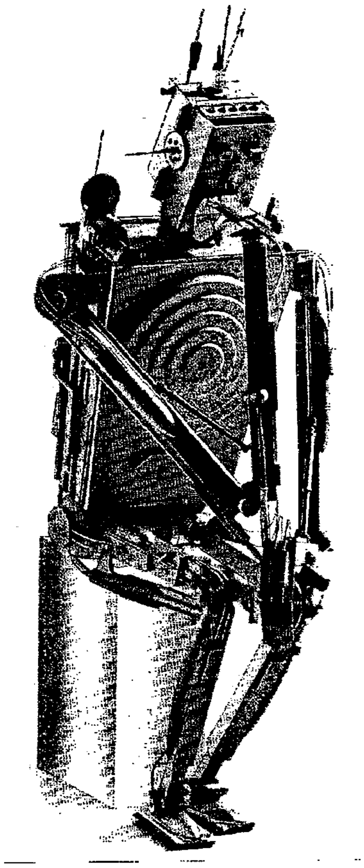
MASTER

Adaptive control of a flexible TR-robot

Doelman, R.B.

Award date:
1995

WFW
Control and Identification of Mechanical Systems
Department of Mechanical Engineering
Eindhoven University of Technology



ADAPTIVE CONTROL OF A FLEXIBLE TR-ROBOT

Ricky Doelman

wfw report 92.084

examencommissie:	prof. dr. ir. J.J. Kok, voorzitter	❑
	ir. I.M.M. Lammerts, coach	❑
	ir. J.G.M.M. Smits	❑
	dr. ir. F.E. Veldpaus	❑
	dr. S. Weiland	❑
colloquium / examenzitting:	27 aug. 1992 - 15.00 - Whoog 0.05	

ABSTRACT

In mechanical systems the presence of deformable subsystems is responsible for the increase of the number of generalized coordinates, i.e. the number of independant coordinates necessary and sufficient to adequately describe the configuration of the system under control. This means that there are more degrees of freedom to be controlled / stabilized than control input signals available. To tackle the control problem for systems which possess the above described property, a control algorithm is derived. The controller designed according to this algorithm is capable of accomplishing a reasonable trajectory tracking, while hereby limiting the elastic vibrations, which occur due to the flexibility in the system. To make the robot also able to perform satisfactorily in the presence of uncertain systemparameters, such as for example a payload with unknown mass, an adaptive version of the control algorithm is derived.

The design of the adaptive controller is based on a 'Basic Algorithm' presented by Slotine and Li in [2] and [3]. However, their method requires the robot to be *rigid*, in other words, there have to be as many control input signals as generalized coordinates. The adaptive controller presented in this report, called **Adaptive Computed Torque Computed Reference Controller**, effectively deals with the tracking control problem for a *flexible* TR - robot, i.e. a Translation - Rotation - Robot with an elastic transmission between the actuator driving the robot arm and the robot arm itself. This system has three degrees of freedom and two control inputs.

To illustrate the qualities of the designed controller, several simulation results are presented in which the performance of the controller developed by Slotine and Li, as well as the Computed Torque Computed Reference Controller, both adaptive and non-adaptive, are discussed and mutually compared.

CONTENTS

Chapter One

INTRODUCTION TO ADAPTIVE CONTROL OF FLEXIBLE ROBOTS

To conform to modern standards, robot manipulators have to maintain high accuracy in trajectory control for a large range of speeds. In order to operate at high speeds, industrial robots have to be lightweight constructed. However, a lightweight manipulator has flexibility in the link structure and in the transmissions between actuators and links. It has been shown that joint elasticity is the dominant source of compliance in most manipulator designs. This joint elasticity may arise from several sources, such as elasticity in the gears, belts, bearings etc. As a result of this compliance in the manipulator, the control system not only has to take account for achieving a reasonable trajectory tracking, but also for a certain stabilization of the occuring vibrations! Now also powerful, low-cost microprocessors necessary for real-time manipulator control have become available, an extensive (re)search for control algorithms designed assuming irrigidity at the joints is justified. Several techniques for designing controllers, able to govern flexible manipulators, have already been developed (e.g. feedback linearisation, singularly perturbed systems, integral manifold approach).

At the Eindhoven University of Technology the approach to the control problem for flexible robots is based upon the familiar Computed Torque Control technique. Ivonne Lammerts has developed a method which is capable of tackling control problems for flexible manipulators, i.e. mechanical systems with more degrees of freedom than control input signals. This so-called **Computed Torque Computed Reference Control (CTCRC)** technique is therefore well-suited for controlling flexible robots, like for instance manipulators with elastic motor transmissions. 'Computed Torque Control' refers to the fact that the control law is designed explicitly on the basis of a model in order to compensate for robot nonlinearities and to guarantee a desired closed loop behaviour.
'Computed Reference Control' refers to the fact that an expression is derived for the reference trajectories of the motor rotors of the actuators controlling the elastically driven links, i.e. the links driven via the elastic motor transmission. This is necessary, because an explicit expression for the desired elastic motor rotor variables is not beforehand. Goal of the controller is not only to take care of the tracking of the desired link-based trajectories, but also to stabilize the occuring elastic vibrations.

As robots often carry loads of uncertain mass, the robot dynamics constantly experiences unpredictable parameter variations. Mostly, there is also the problem of other unknown systemparameters, such as inertias, friction coefficients, etc.

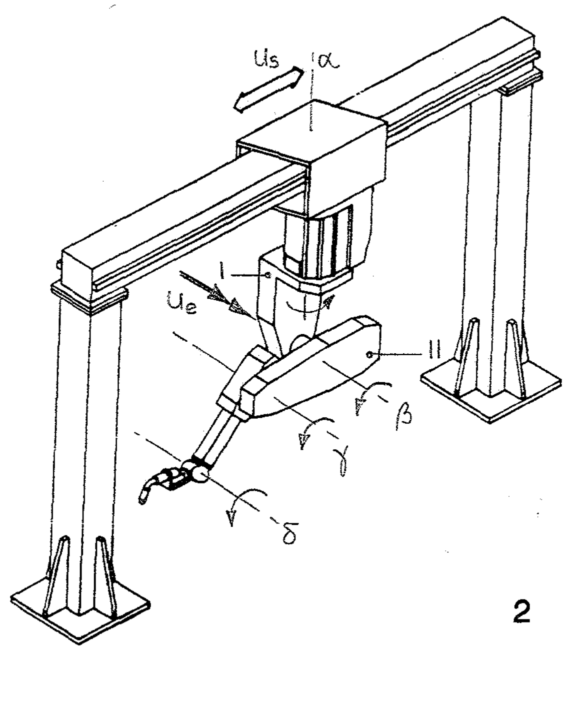
To maintain a good performance of the system in the presence of uncertain or unknown plant parameters, **adaptive control** is required. An adaptive control system is capable of adjusting one or more parameters of the control system so as to force the response of the resulting closed loop system towards a desired one. The adaptive control system estimates the uncertain plant parameters on basis of the measured system signals, and uses these on-line estimated parameters in the control input computation. The question whether adaptive control can be applied and how this could be done in the case of the newly developed **Computed Torque Computed Reference Control** technique, will be the main issue in this report.

The controller design problem can be stated as follows:

> *given a desired trajectory of the end-effector* ***x(t)****, and with some or all constant manipulator parameters* ***p*** *being unknown, derive an adaptive version of the Computed Torque Computed Reference Control technique applied to a flexible TR-robot. This means, derive a CTCRC control law and an adaptation algorithm for the parameter estimation vector* $\hat{\mathbf{p}}(t)$*, such that after an initial adaptation process:*
>
> - *system stability is guaranteed for all* $t>t_0$
> - *the goal of trajectory tracking is achieved*
> - *the occuring elastic vibrations remain bounded*

In this report we focus our attention to the adaptive control of a flexible TR-robot. As a model we use a carriage on which an inverted pendulum is fixed. The carriage is driven by an actuator with a stiff transmission, while the pendulum is driven by a motor with an elastic transmission. The elasticity in the revolute joint is modeled as a linear, torsional spring with known characteristics. This mechanical system has three degrees of freedom and two control inputs.

To illustrate what is meant by a *flexible TR-robot*, we consider the following figure (figure extracted from [7]).



For simplicity we suppress the rotation around the α-, γ- and δ-axis; we then obtain a *TR-robot*.
Now suppose that the shaft between body I and body II is not torsional rigid; we then obtain a *flexible TR-robot*.
If we model the elastic revolute joint as a linear, torsional spring, we obtain a *TR-robot with 2 inputs and 3 degrees of freedom*.

Chapter Two

ADAPTIVE CONTROL OF A RIGID TR - ROBOT

This chapter serves as an introduction to the adaptive control problem for a flexible TR - robot. We start with how to design adaptive control laws based upon Lyapunov's second method and designed according to the Basic Algorithm of Slotine and Li. After derivation of Lagrange's equations of motion and specification of the desired trajectories, an investigation of the control problem for the rigid TR - robot is presented including simulation results.

2.1 Designing the control law and adaptation algorithm

According to Lyapunov's stability analysis, a globally stable adaptive controller can be obtained if there exists a scalar function V of the state **x**, V(**x**), which meets the following requirements:

$$V(\mathbf{x}) > 0 \quad \text{for } \mathbf{x} \neq \mathbf{0} \quad \text{positive definite}$$

$$\dot{V}(\mathbf{x}) < 0 \quad \text{for } \mathbf{x} \neq \mathbf{0} \quad \text{negative definite}$$

$$V(\mathbf{x}) \rightarrow \infty \quad \text{for} \quad \|\mathbf{x}\| \rightarrow \infty$$

$$V(\mathbf{0}) = 0$$

Suppose the equations describing the dynamics of the *rigid* manipulator (in the absence of friction and other disturbances) can be written as

$$\mathbf{H}(\mathbf{q},\mathbf{p})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}},\mathbf{p})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q},\mathbf{p}) = \mathbf{u}$$

with	$\mathbf{q}$:	*nx1 vector of generalized coordinates*
	$\dot{\mathbf{q}}$:	*nx1 vector of generalized velocities*
	$\ddot{\mathbf{q}}$:	*nx1 vector of generalized accelerations*
	$\mathbf{p}$:	*mx1 vector of constant systemparameters*
	$\mathbf{H}(\mathbf{q},\mathbf{p})$:	*nxn symmetric, positive definite manipulator inertia matrix*
	$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}},\mathbf{p})\dot{\mathbf{q}}$:	*nx1 vector of centripetal and Coriolis torques*
	$\mathbf{g}(\mathbf{q},\mathbf{p})$:	*nx1 vector of gravitational torques*
	$\mathbf{u}$:	*nx1 vector of applied joint torques / forces*

According to the Basic Algorithm of Slotine and Li (see [2] - §9.2.1) we choose the adaptive control law

$$u = \hat{H}(q,\hat{p})\ddot{q}_r + \hat{C}(q,\dot{q},\hat{p})\dot{q}_r + \hat{g}(q,\hat{p}) + \kappa\dot{e}_r$$

This control law consists of two parts. The first part consists of terms corresponding to inertial, centripetal and Coriolis, and gravitational torques. The second part contains terms representing PD feedback.

with	$\dot{q}_r$:	*nx1 vector of reference velocities*
	$\ddot{q}_r$:	*nx1 vector of reference accelerations*
	$\hat{p}$:	*rx1 vector of estimated parameters ($r \leq m$)*
	$\hat{H}(q,\hat{p})$:	*nxn symmetric, estimated manipulator inertia matrix*
	$\hat{C}(q,\dot{q},\hat{p})\dot{q}_r$:	*nx1 vector of estimated centripetal and Coriolis torques*
	$\hat{g}(q,\hat{p})$:	*nx1 vector of estimated gravitational torques*
	$\dot{e}_r = \dot{q}_r - \dot{q}$:	*nx1 vector of reference velocity errors*
	κ:	*nxn diagonal, constant, positive definite matrix*

The reference velocity vector $\dot{\mathbf{q}}_r$ is formed by shifting the desired velocity $\dot{\mathbf{q}}_d$ according to the position error $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$, thus

$$\dot{q}_r = \dot{q}_d + \Lambda(q_d - q)$$

with	q_d:	*nx1 vector of desired link coordinates*
	$\dot{q}_d$:	*nx1 vector of desired link velocities*
	Λ:	*nxn diagonal, constant, positive definite matrix*

For the reference velocity error we then find

$$\dot{e}_r = (\dot{q}_d - \dot{q}) + \Lambda(q_d - q)$$

$$= \dot{\tilde{q}} + \Lambda\tilde{q}$$

with	$\tilde{q}$:	*nx1 vector of position errors*
	$\dot{\tilde{q}}$:	*nx1 vector of velocity errors*

According to Slotine and Li (see [2] - §7.1.1) $\dot{\mathbf{e}}_r$ can be considered as a sliding surface for $\mathbf{q}$. The equation for $\dot{\mathbf{e}}_r$ can also be seen as a stable first-order differential equation in $\tilde{\mathbf{q}}$, with $\dot{\mathbf{e}}_r$ as an input (Λ>**0**). If we can somehow arrange for $\dot{\mathbf{e}}_r$ to tend to **0** as time t tends to infinity (assuming bounded initial conditions), then also the position error $\tilde{\mathbf{q}}$ and the velocity error $\mathring{\tilde{\mathbf{q}}}$ will tend to zero, and trajectory control is achieved.

The following question arises, and that is, how can we arrange for the reference velocity error $\dot{\mathbf{e}}_r$ to tend to **0**?
We therefore substitute the control law into the equations of motion. This yields

$$H(\ddot{q} - \ddot{q}_r) + C(\dot{q} - \dot{q}_r) = (\hat{H} - H)\ddot{q}_r + (\hat{C} - C)\dot{q}_r + (\hat{g} - g) + \kappa\dot{e}_r$$

Define the reference position error $\mathbf{e}_r = \mathbf{q}_r - \mathbf{q}$ and we find for the equivalent error equation of the closed loop system

$$H\ddot{e}_r + C\dot{e}_r + \kappa\dot{e}_r = -\,[(\hat{H} - H)\ddot{q}_r + (\hat{C} - C)\dot{q}_r + (\hat{g} - g)]$$

$$= -\,W_r(q,\ \dot{q},\ \dot{q}_r,\ \ddot{q}_r)\tilde{p}$$

with	$\tilde{p}(t) = \hat{p}(t) - p$	*rx1 vector of parameter errors*
	W_r	*nxr matrix of functions* q, $\dot{q}$, $\dot{q}_r$, $\ddot{q}_r$

It must be said that the above derivation is only valid if the error equation of the closed loop system is **linear** in terms of the adjustable parameters $\hat{\mathbf{p}}$(t) (so - called *linearly parameterized*).

To show global stability of this adaptive control system, we have to find a suitable Lyapunov - function. We use the function

$$V = \frac{1}{2}\,[\ \dot{e}_r^T H \dot{e}_r + \tilde{p}^T \Gamma^{-1} \tilde{p}\ ]$$

with	Γ:	*rxr diagonal, constant, positive definite matrix*

The first term in this expression represents a virtual mechanical energy in the error system, while the second term is a positive definite quadratic expression of the parameter error vector $\tilde{\mathbf{p}}(t) = \hat{\mathbf{p}}(t) - \mathbf{p}$.

Differentiating this Lyapunov - function with respect to time yields

$$\dot{V} = \dot{\boldsymbol{e}}_r^T[\ \boldsymbol{H}\ddot{\boldsymbol{e}}_r + \frac{1}{2}\dot{\boldsymbol{H}}\dot{\boldsymbol{e}}_r\ ] + \tilde{\boldsymbol{p}}^T\Gamma^{-1}\dot{\tilde{\boldsymbol{p}}}$$

Substituting the error equation of the closed loop system

$$\boldsymbol{H}\ddot{\boldsymbol{e}}_r = -\ [\ \boldsymbol{W}_r\,\tilde{\boldsymbol{p}} + \boldsymbol{C}\dot{\boldsymbol{e}}_r + \kappa\dot{\boldsymbol{e}}_r\ ]$$

gives

$$\dot{V} = -\dot{\boldsymbol{e}}_r^T\kappa\dot{\boldsymbol{e}}_r \ + \frac{1}{2}\dot{\boldsymbol{e}}_r^T(\dot{\boldsymbol{H}}\ -2\boldsymbol{C})\dot{\boldsymbol{e}}_r \ + \tilde{\boldsymbol{p}}^T\Gamma^{-1}\dot{\tilde{\boldsymbol{p}}} \ - \dot{\boldsymbol{e}}_r^T\boldsymbol{W}_r\,\tilde{\boldsymbol{p}}$$

In [2] - §9.1.2 Slotine and Li prove the skew-symmetry of the matrix $\dot{\mathbf{H}}$**-2C**, which means that for all $\dot{\mathbf{e}}_r$

$$\dot{\boldsymbol{e}}_r^T(\dot{\boldsymbol{H}}\ -2\boldsymbol{C})\dot{\boldsymbol{e}}_r = 0$$

Rearranging the equation for $\dot{V}$ and using the skew - symmetry of $\dot{\mathbf{H}}$-2**C** yields

$$\dot{V} = -\dot{\boldsymbol{e}}_r^T\kappa\dot{\boldsymbol{e}}_r - \tilde{\boldsymbol{p}}^T(\boldsymbol{W}_r^T\dot{\boldsymbol{e}}_r - \Gamma^{-1}\dot{\tilde{\boldsymbol{p}}})$$

To meet the requirements imposed upon the Lyapunov function and its time derivative, the gain matrix κ is chosen positive definite and the second term is chosen equal to 0. This yields the adaptation algorithm for the parameter estimation vector $\hat{\mathbf{p}}(t)$:

$$\dot{\tilde{\boldsymbol{p}}} = \Gamma\boldsymbol{W}_r^T\dot{\boldsymbol{e}}_r$$

By choosing the above adaptation algorithm and control law, we arranged for the positive definite Lyapunov-function V(x) to be a monotonically decreasing function. This means that the 'kinetic energy' in the error system constantly decreases, and hence, the 'velocity' in the error system $\dot{\mathbf{e}}_r$ reduces to zero.

We have shown in this section that with the control law

$$\boldsymbol{u} = \hat{\boldsymbol{H}}(\boldsymbol{q},\hat{\boldsymbol{p}})\ddot{\boldsymbol{q}}_r + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}},\hat{\boldsymbol{p}})\dot{\boldsymbol{q}}_r + \hat{\boldsymbol{g}}(\boldsymbol{q},\hat{\boldsymbol{p}}) + \kappa\dot{\boldsymbol{e}}_r$$

and adaptation algorithm

$$\dot{\hat{\boldsymbol{p}}} = \Gamma \boldsymbol{W}_r^T\dot{\boldsymbol{e}}_r$$

it is guaranteed that

* the steady state position error becomes zero, $\mathbf{q} \rightarrow \mathbf{q}_d$ as $t \rightarrow \infty$
* the steady state velocity error becomes zero, $\dot{\mathbf{q}} \rightarrow \dot{\mathbf{q}}_d$ as $t \rightarrow \infty$

The structure of the adaptive controller is sketched in fig. 2.1.

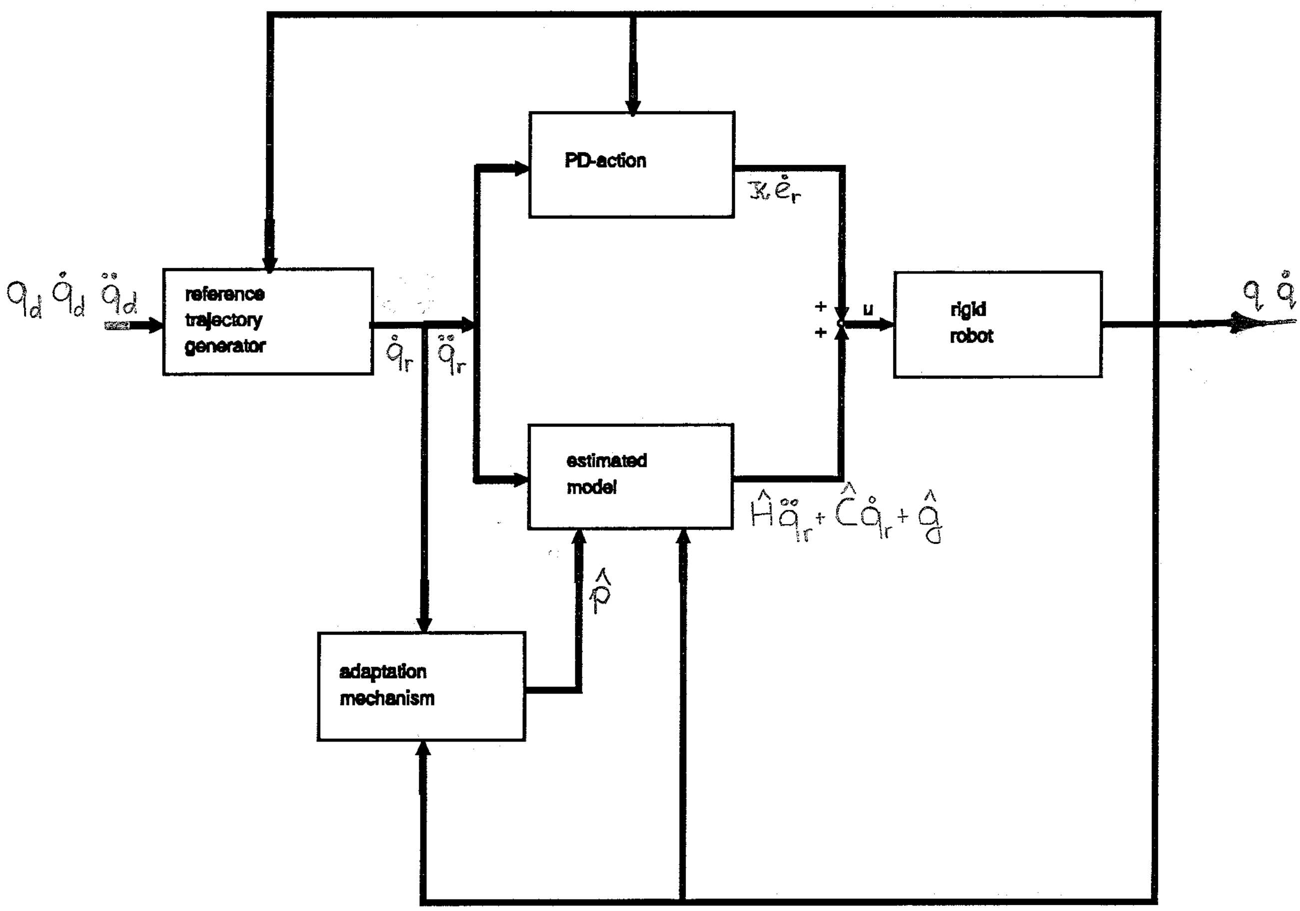


fig. 2.1 structure of the adaptive controller according to Slotine and Li

In the following section we applicate the above theory to the (adaptive) control problem of a rigid TR - robot. First, we continue with the derivation of the equations of motion and specification of the desired trajectories.

2.2 Equation of motion and desired trajectories

The following figure provides a schematical model of a rigid TR - robot

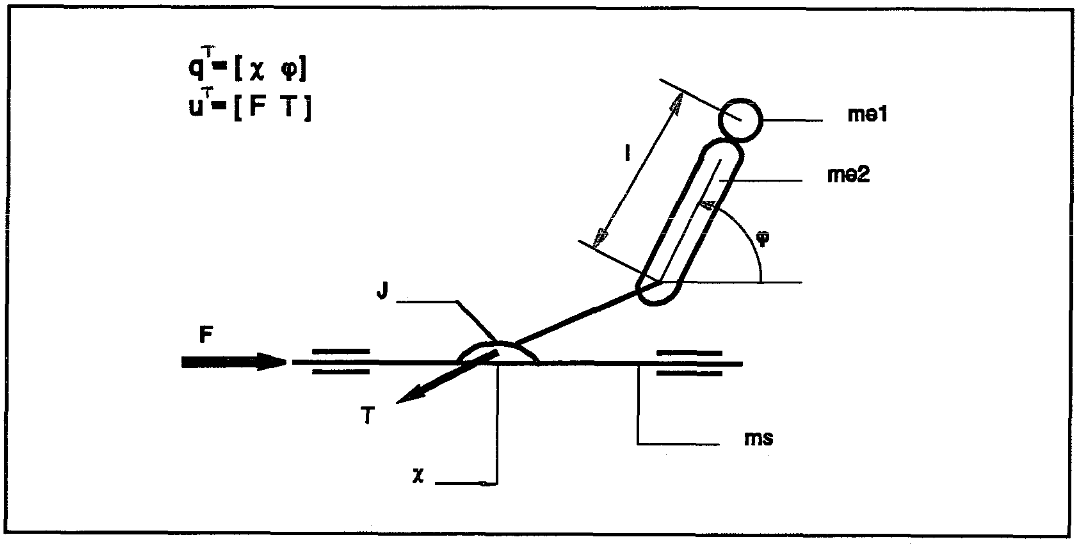


fig. 2.2 rigid TR - robot

m_s:	mass of the carriage
m_{e1}:	mass of the payload at the end of the arm
m_{e2}:	mass of the robot arm
l:	length of the robot arm
J:	inertia of the motor rotor
χ:	horizontal translation of the carriage
φ:	rotation of the robot arm
F:	force acting on the translating carriage
T:	torque acting on the rotating arm

Using Lagrange's equations it can be shown that the dynamic equations of the rigid TR - robot are:

$$\begin{bmatrix} m_1 & -m_2 l\sin(\varphi) \\ -m_2 l\sin(\varphi) & m_3 \end{bmatrix} \begin{bmatrix} \ddot{\chi} \\ \ddot{\varphi} \end{bmatrix} + \begin{bmatrix} -m_2 l\dot{\varphi}^2\cos(\varphi) \\ m_2 l g\cos(\varphi) \end{bmatrix} = \begin{bmatrix} F \\ T \end{bmatrix}$$

with

$$m_1 = m_{e1} + m_{e2} + m_s$$
$$m_2 = m_{e1} + (1/2)m_{e2}$$
$$m_3 = m_{e1}l^2 + (1/3)m_{e2}l^2 + J$$

The desired trajectories for link - motion control $[\chi_d(t)\ \varphi_d(t)]^T$ are derived from a certain trajectory of the payload at the end of the robot arm $[x_d(t)\ y_d(t)]^T$. This gripper trajectory is specified to be a constant circulation in two - dimensional space.

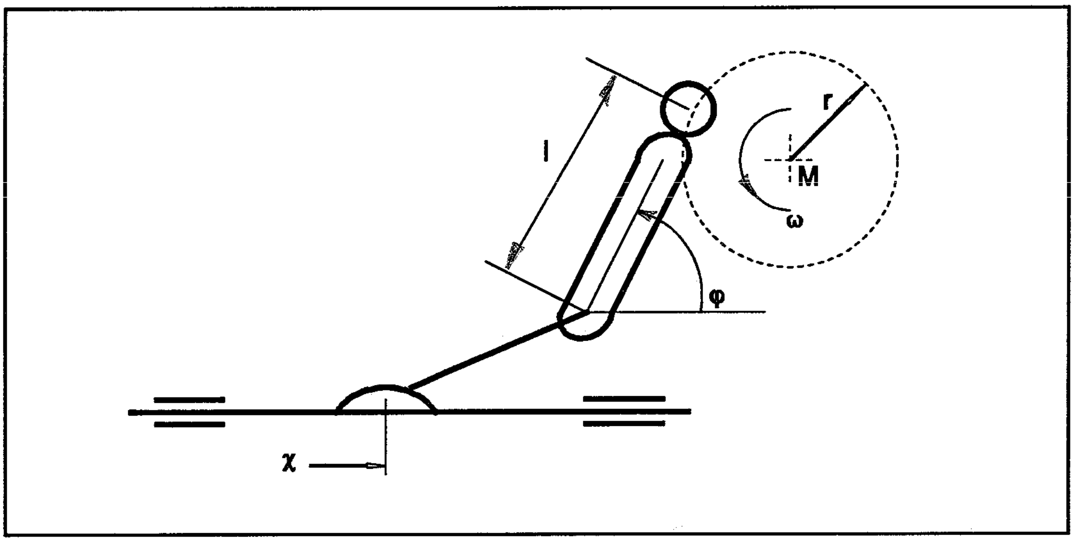


fig. 2.3 desired trajectory of the payload

l:	length of the robot arm
(x_M, y_M, r):	defining the circle to be tracked
ω:	angular velocity of the payload

The following figure depicts the desired trajectories for the carriage and pendulum

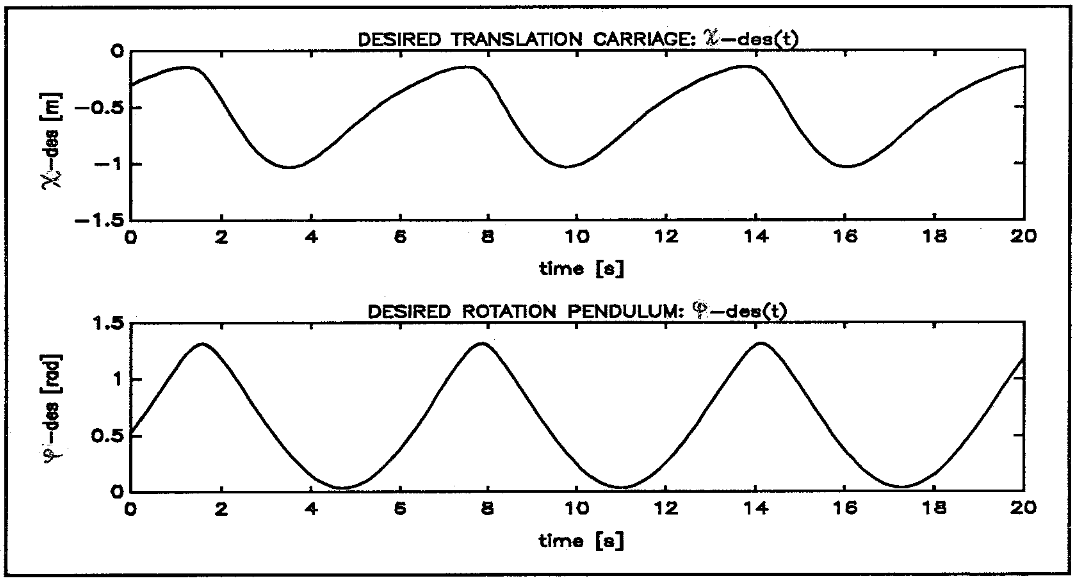


fig. 2.4a desired trajectory of the carriage $\chi_d(t)$
fig. 2.4b desired trajectory of the pendulum $\varphi_d(t)$

2.3 Adaptive control of a rigid TR - robot

In this section some simulation results are presented. We assume that the pendulum rotates in the horizontal plane (**g** = **0**).

parameters used for simulation

m_{e1}	= 2:	mass of the payload	[kg]
m_{e2}	= 3:	mass of the robot arm	[kg]
m_s	= 10:	mass of the carriage	[kg]
J	= 5:	inertia of the motor rotor	[kgm^2]
r	= 0.35:	radius of the circle to be tracked	[m]
l	= 0.75:	length of the robot arm	[m]
ω	= 1:	angular velocity payload	[rad/s]
x_M	= 0:	x-coordinate circle center	[m]
y_M	= l/2:	y-coordinate circle center	[m]

Example 2.1: *comparison of the adaptive controller with the non-adaptive controller*

In this example we consider the case in which the links of the rigid TR - robot are required to follow the desired trajectories according to fig. 2.4. We initiate the simulations with the end-effector starting from the origin with zero initial velocity ($\chi(t_0)$=0 and $\varphi(t_0)$=0). Despite of the fact that the initial link coordinates $\chi(t_0)$ and $\varphi(t_0)$ and their time derivatives are not according to the desired trajectories, after a transient, the controller is still able to reduce the position error $\tilde{\mathbf{q}}$ to zero (fig. 2.5a).

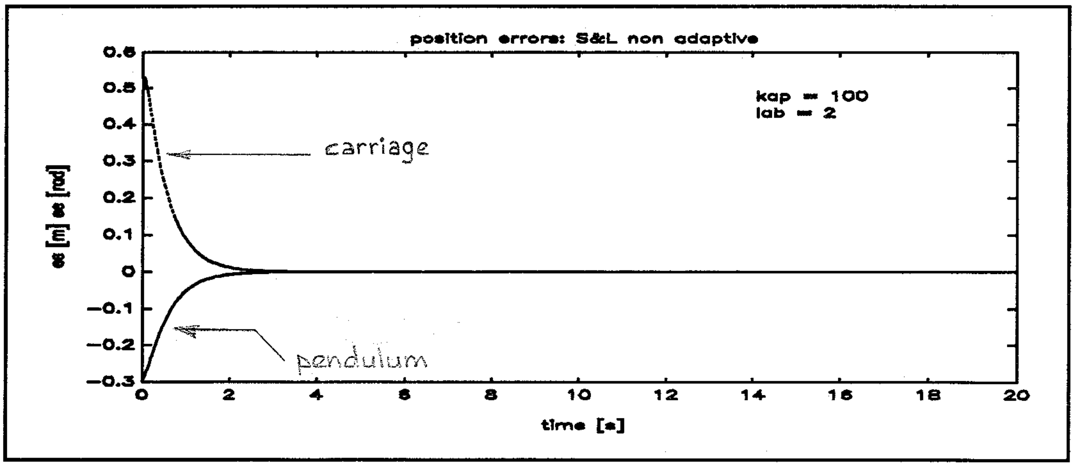


fig. 2.5a non-adaptive control of a rigid TR-robot
m_{e1}-control law = m_{e1}-model

Suppose we do not have full knowledge about the mass of the payload m_{e1}. For example, we substitute a value for the mass m_{e1} in the control law, which is 5 times larger than 'in reality'. It is easy to see that the performance of the system is beyond acceptance (fig. 2.5b); the steady state position error is much larger than the tolerance on the integration algorithm (tol = 10^{-4}).

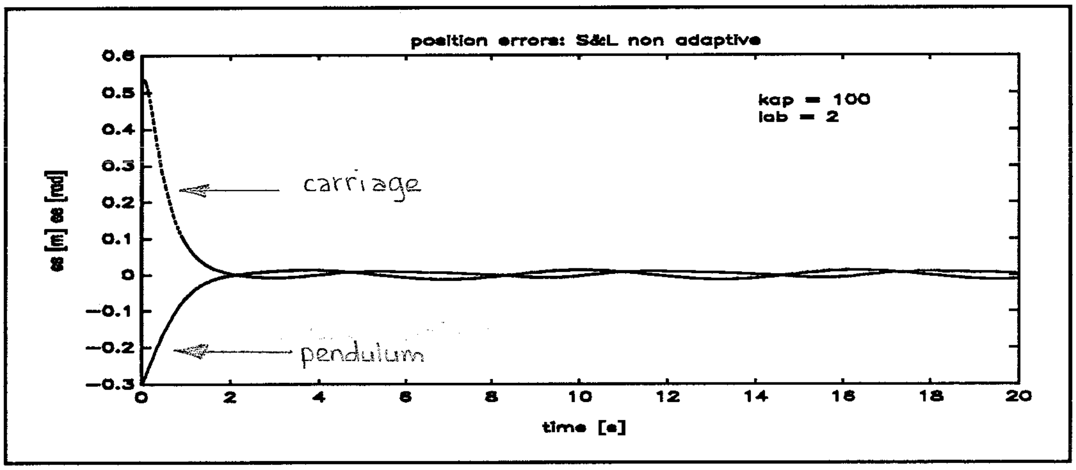


fig. 2.5b non-adaptive control of a rigid TR-robot
*m_{e1}-control law = 5*m_{e1}-model*

To deal with the problem of tracking control in the case of uncertain or unknown parameters, adaptive control is required. Fig 2.5c, see below, shows the simulation result when an adaptive control law is used to govern the rigid TR - robot. The mass of the payload m_{e1} is estimated on-line, and this estimation is used in the control input computation. The simulation is started with no a priori information about m_{e1}, thus $\hat{m}_{e1}(t_0) = 0$. Notice that tracking control is achieved. Remark: the estimated parameters converge to their true values.

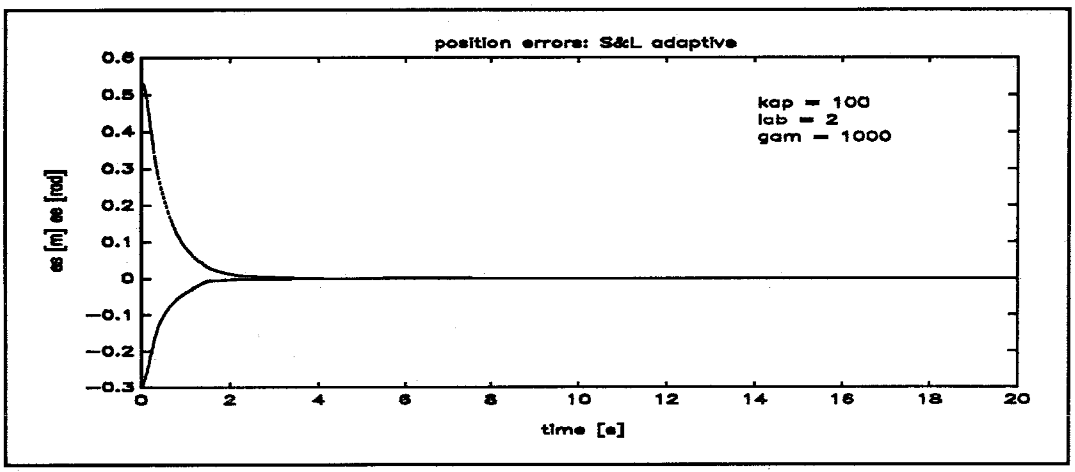


fig. 2.5c adaptive control of a rigid TR-robot
the control law is computed with an estimated value for m_{e1}, with $\hat{m}_{e1}(t_0) = 0$

Example 2.2: *finding proper values for the control gains κ, Λ and Γ*

Qualitatively, we can describe the effects of the gain matrices as follows

κ determines how fast the reference velocity error $\dot{\mathbf{e}}_r$ is reduced to **0**
Λ determines how fast the position error $\tilde{\mathbf{q}}$ and velocity error $\dot{\tilde{\mathbf{q}}}$ damp out
Γ determines at what rate the estimated parameters $\hat{\mathbf{p}}$ are adjusted

Unfortunately, properly quantifying the gain matrices is a process of trial-and-error. To illustrate the difficulty in finding a suitable value for a gain matrix, in this case Γ, we consider the adaptive control of a mass m on a frictionless surface by a motor force u, with the plant dynamics being $m\ddot{x} = u$.

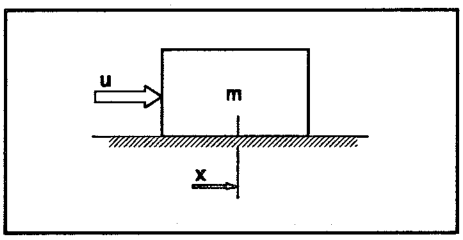


fig 2.6 mass on a frictionless surface

If we choose Γ too high, we make too great demands upon the adjusting rate for the parameter estimate $\hat{\mathbf{p}}$. The 'wild' behaviour of $\hat{\mathbf{p}}(t)$ results in a high control activity, and thus strongly influences the link positions and velocities. Observe that peaks appear in **u** and $\dot{\mathbf{q}}$ at the same time that peaks appear in $\hat{\mathbf{p}}(t)$ (fig. 2.7a).

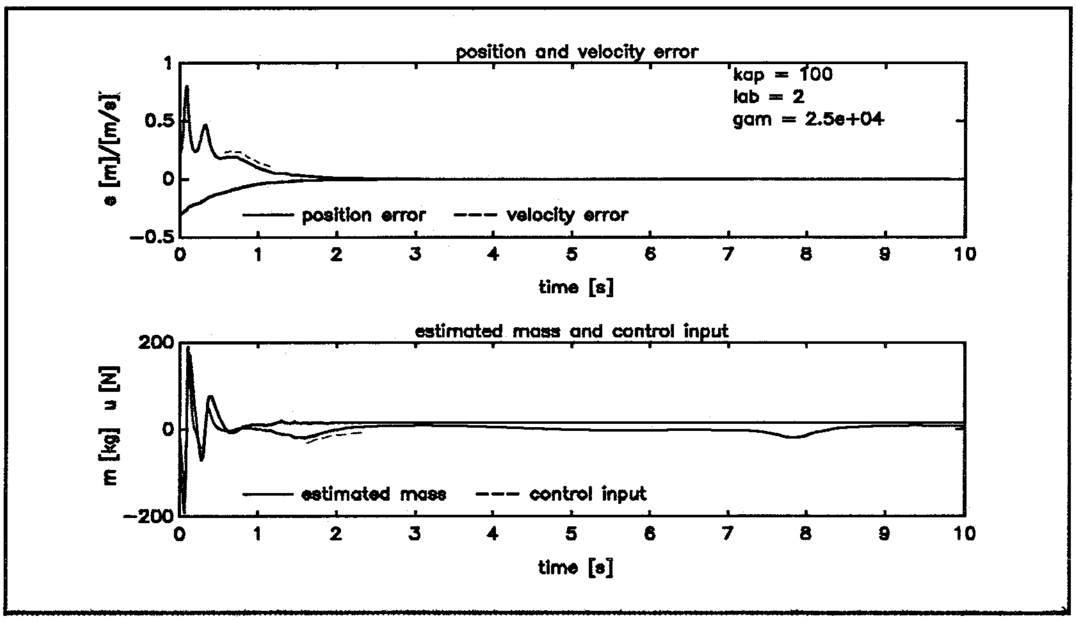


fig. 2.7a $\|\Gamma\|_2$ = 'too high'

By lowering Γ we obtain a smoother progress in time for the parameter estimate $\hat{\mathbf{p}}$, and thus a smoother progress in time for the system states $\mathbf{q}$ and $\dot{\mathbf{q}}$ (fig 2.7b). So, peaks in the system variables, occuring at places where we do not expect them at first sight, do not necessarily have to result from implementation errors !!!

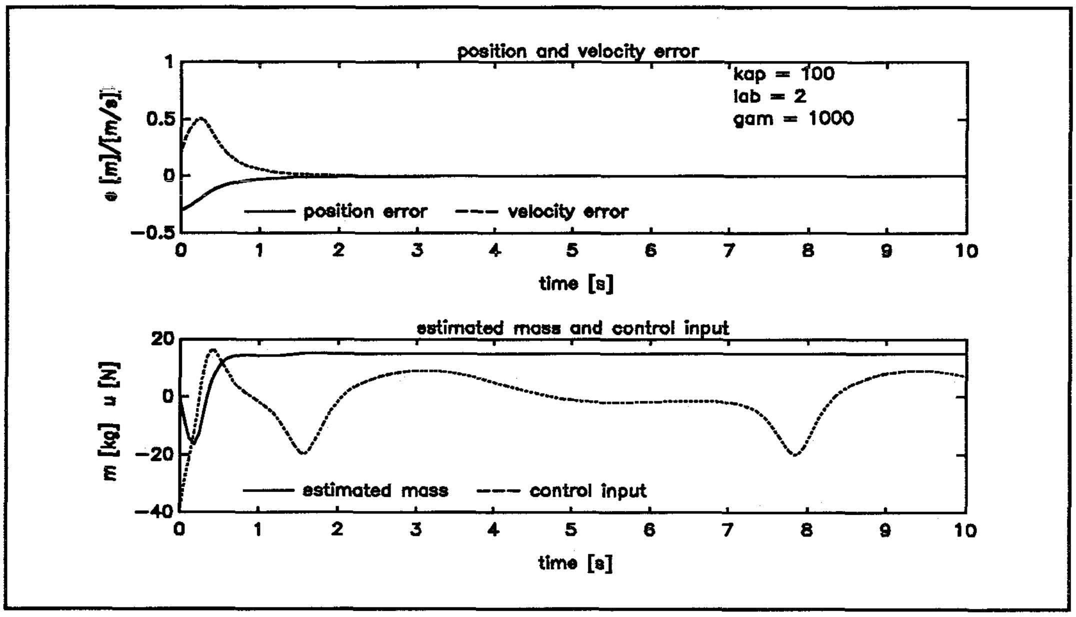


fig. 2.7b $\|\Gamma\|_2$ *= 'properly chosen'*

However, if Γ is chosen too low, the adaptive mechanism is not stimulated enough to arrange for the parameter error $\tilde{\mathbf{p}}$ to become constant within the simulation time-interval (fig. 2.7c).

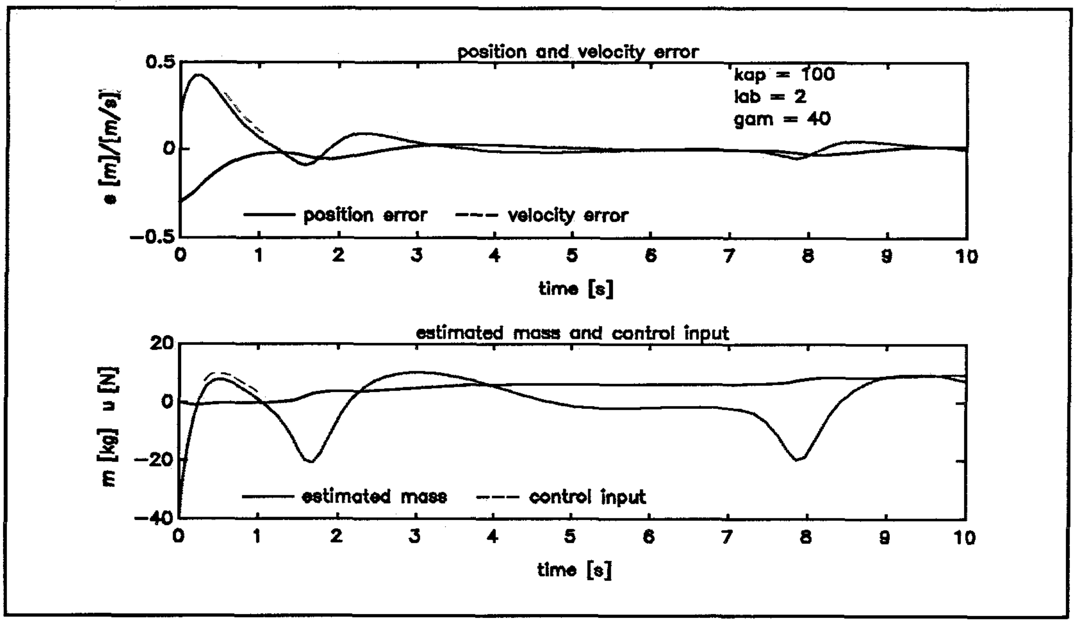


fig. 2.7c $\|\Gamma\|_2$ *= 'too low'*

Notice that in the case of an input signal of "sufficiently richness", both the tracking error $\tilde{\mathbf{q}}$, as well as the parameter error $\tilde{\mathbf{p}}$, converge to zero (fig. 2.7b). If the control input is not persistently exciting, the tracking error $\tilde{\mathbf{q}}$ converges to zero, but the parameter error $\tilde{\mathbf{p}}$ does not: see fig. 2.7d (m = 15, while $\hat{m}(t_e)$ = 8,3).

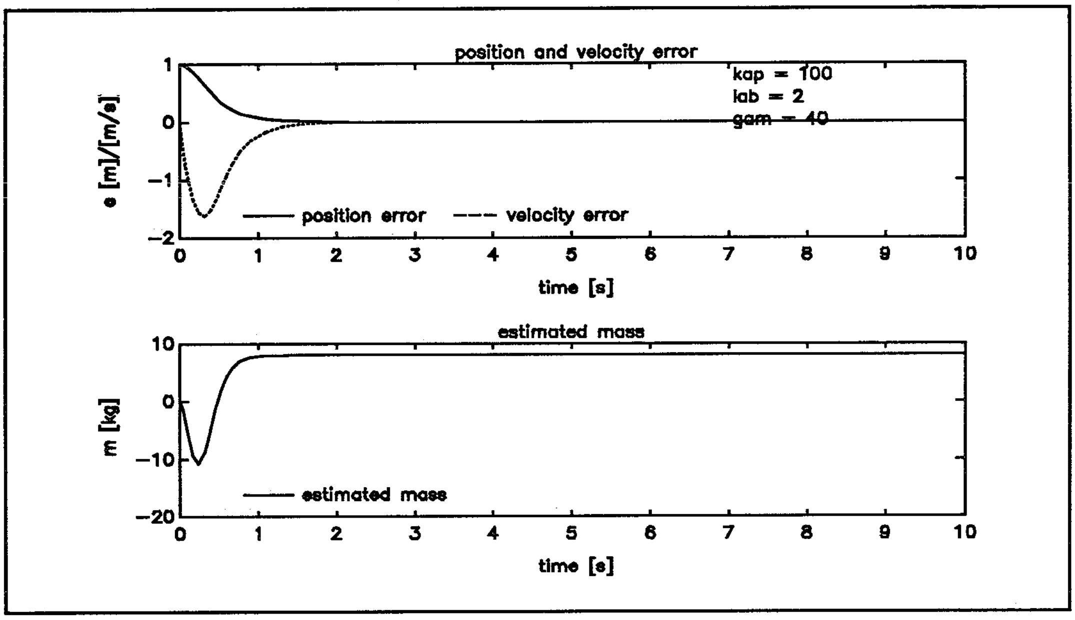


fig. 2.7d $\|\Gamma\|_2$ = 'properly chosen'
the control input is not persistently exciting

2.4 Summary

Adaptive control is a powerful method for controlling dynamic systems with uncertain or unknown parameters. An adaptive control technique based on Lyapunov's stability theory and developed by Slotine and Li, is used for controlling a rigid TR-robot manipulating a payload with an uncertain mass. Simulations show that the addition of an adaptive mechanism to a non-adaptive controller greatly improves the performance of the overall system. However, the gain matrix determining the adaptation rate for the adjustable parameters, needs to be 'properly' quantified. Also, the adaptive controller requires availability of the full state vector and the error equation of the closed loop system to be linearly parametrized.

Chapter Three

ADAPTIVE CONTROL OF A FLEXIBLE TR-ROBOT

In the previous chapter we considered a rigid TR - robot. In other words, we assumed the spring stiffness in the elastic revolute joint to be infinitely large. In this chapter, we study the system's performance in the case of an elastic transmission between actuator and link. To illustrate the effects of the flexibility in the transmission, special attention is given to simulations where the magnitude of the spring stiffness k is chosen to be 'relatively small'. One might think that the adaptive control problem for a flexible TR - robot is a straightforward extension of its rigid equivalent. This is certainly not the case. The difficulty in controlling flexible robots is that there are more degrees of freedom to be controlled / stabilized than control input signals. Especially calculation of the torque acting on the elastic transmission turns out to be quite long-winded.
The same strategy is followed as in the rigid case. First, we start with a description of the control problem. Then, we outline the methodology of how to design an adaptation algorithm and a control law. Next, Lagrange's equations describing the dynamics of the plant and the trajectories to be tracked are derived. We end this chapter with an investigation of the control problem for a flexible TR - robot including simulation results.

3.1 Problem description

To illustrate the effect of an elastic revolute joint on the control system design, we consider the following figures depicting a rigid TR - robot (fig. 3.1), respectively a flexible TR - robot (fig. 3.2).

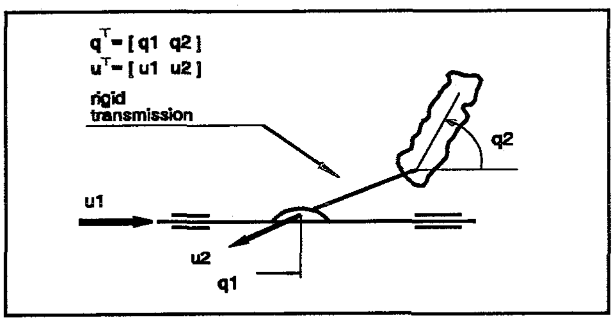


fig. 3.1 rigid TR-robot:
TR - robot with a rigid transmission between actuator and link

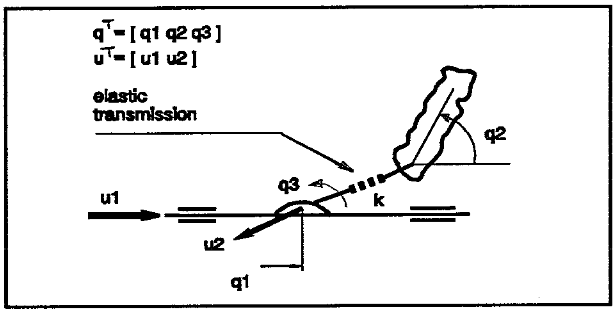


fig. 3.2 flexible TR-robot:
TR - robot with an elastic transmission between actuator and link

Remark that in the case of a rigid TR-robot (fig. 3.1) there are two degrees of freedom and two control inputs. However, if the transmission between actuator and link is modeled as being flexible instead of rigid, the set of generalized coordinates $\mathbf{q}_r^T = [q_1\ q_2]$ has to be augmented with an extra coordinate q_3 to a new set $\mathbf{q}_f^T = [q_1\ q_2\ q_3]$ (fig. 3.2). This extra coordinate defines the rotation of the motor rotor, and is necessary and sufficient in order to adequately describe the kinematics of the system. Yet, there are only two control input signals available !!! Appearantly, **in mechanical systems the presence of deformable subsystems introduces the problem that the number of degrees of freedom to be controlled / stabilized is greater than the number of control input signals.** In our case, the mechanical system is a TR-robot, and the deformable subsystem is an elastic transmission between the actuator driving the robot arm and the robot arm itself.

What we would like to have is a controller, which despite of the elasticity in the transmission, makes the end-effector of the flexible robot track the same desired trajectory as its rigid counterpart, and hereby keeping the occuring vibrations between acceptable bounds.

The **Computed Torque Computed Reference Control Technique** (presented in [1]) is a powerful tool for designing such a controller. The basic idea behind the technique is to replace the desired trajectories of the generalized coordinates by reference trajectories in order to obtain a smoother robot performance in space. In time, the reference trajectories converge to the desired trajectories. However, there is one major problem, and that is that not all the desired trajectories of the generalized coordinates are known. In the case of our TR-robot we are able to derive the desired trajectories of the carriage and pendulum from the known desired gripper path. But when the transmission is elastically modeled, we do not have the slightest idea about the desired trajectory of the motor rotor, and consequently, a reference trajectory for the rotation of the motor rotor cannot be obtained. We cope with this problem by extracting an expression for this reference trajectory from the control law, instead of obtaining the reference trajectory through an adjustment of the desired trajectory.

This extraction of the reference trajectory from the control law is typical for the Computed Torque Computed Reference Control Technique. Therefore, it is unique and it distinguishes itself from other control techniques, such as for example the 'Basic Algorithm' developed by Slotine and Li in [2] and [3]. Though the CTCRC technique has its roots in the Basic Algorithm of Slotine and Li, its main merit is that it is applicable to the control of flexible systems. One could consider Computed Torque Computed Reference Control as an Extended Version of the Basic Algorithm by Slotine and Li.

In [4], Brevoord applies the CTCRC-technique to the control of an xy-table with an *elastic* motor transmission. In many other practical applications, however, we must deal with unknown or uncertain systemparameters. In order to arrange for an acceptable performance of the system in the case of parameter uncertainty, adaptive control might be required. In [5], Vijverstra presents the theoretical and experimental results of various adaptive schemes used for controlling *rigid* manipulators.

In this report our main concern is how to tackle the adaptive control problem of a *flexible* manipulator. The complexity of the control problem contains the following aspects:

1. the system is *multivariable*
2. the system is *nonlinear*
3. the system is *flexible*
4. the system has *unknown parameters*

We deal with this complexity by

1. using the *State Space Approach*
2. applying *Computed Torque Control*
3. applying *Computed Reference Control*
4. applying *Adaptive Control*

The way we synthesize the above individual concepts into one control algorithm, **Adaptive Computed Torque Computed Reference Control**, is described in the following sections.

3.2 Designing the adaptation algorithm

In this section, we derive the adaptation algorithm for the adjustable parameter $\hat{\mathbf{p}}$ using Lyapunov's second method. We carry out the derivation under the assumption that the reference velocity and reference acceleration $\dot{\mathbf{q}}_r$, respectively $\ddot{\mathbf{q}}_r$ are totally known. This is not the case, because we do not (yet) possess an expression for the reference trajectory of the motor rotor $\mathbf{q}_{mr}$ and its time derivatives. In the next section, however, we show how with the Computed Reference Method we derive expressions for the motor rotor variables.

Suppose the equation describing the dynamics of the *flexible* system (in the absence of friction and other disturbances) can be written as

$$H(q,p)\ddot{q} + C(q,\dot{q},p)\dot{q} + g(q,p) + K_k q = Du$$

with	n:	*number of generalized coordinates necessary and sufficient to describe the kinematics of the n rigid links*
	e:	*number of generalized coordinates necessary and sufficient to describe the deformations in the elastic transmissions*
	q:	*(n+e)x1 vector of generalized coordinates*
	$\dot{q}$:	*(n+e)x1 vector of generalized velocities*
	$\ddot{q}$:	*(n+e)x1 vector of generalized accelerations*
	p:	*mx1 vector of constant systemparameters*
	$H(q,p)$:	*(n+e)x(n+e) symmetric, positive definite manipulator inertia matrix*
	$C(q,\dot{q},p)\dot{q}$:	*(n+e)x1 vector of centripetal and Coriolis torques*
	$g(q,p)$:	*(n+e)x1 vector of gravitational torques*
	K_k:	*(n+e)x(n+e) symmetric, semi-positive definite stiffness matrix*
	D:	*(n+e)xn distribution matrix*
	u:	*nx1 vector of applied joint torques / forces*

Following the Basic Algorithm of Slotine and Li (see [2] - §9.2.1), we choose the adaptive control law

$$Du = \hat{H}(q,\hat{p})\ddot{q}_r + \hat{C}(q,\dot{q},\hat{p})\dot{q}_r + \hat{g}(q,\hat{p}) + K_k q_r + \kappa \dot{e}_r$$

with	q_r:	*(n+e)x1 vector of reference positions*
	$\dot{q}_r$:	*(n+e)x1 vector of reference velocities*
	$\ddot{q}_r$:	*(n+e)x1 vector of reference accelerations*
	$\hat{p}$:	*rx1 vector of estimated parameters ($r \leq m$)*
	$\hat{H}(q,\hat{p})$:	*(n+e)x(n+e) symmetric, estimated manipulator inertia matrix*
	$\hat{C}(q,\dot{q},\hat{p})\dot{q}_r$:	*(n+e)x1 vector of estimated centripetal and Coriolis torques*
	$\hat{g}(q,\hat{p})$:	*(n+e)x1 vector of estimated gravitational torques*
	$\dot{e}_r = \dot{q}_r - \dot{q}$:	*(n+e)x1 vector of reference velocity errors*
	κ:	*(n+e)x(n+e) diagonal, constant, positive definite matrix*

This control law consists of two parts. The first part consists of terms corresponding to inertial, centripetal and Coriolis, gravitational and elastic transmission torques. The second part contains terms representing PD feedback.

The reference velocity vector $\dot{\mathbf{q}}_r$ is formed by shifting the desired velocity $\dot{\mathbf{q}}_d$ according to the position error $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$, thus

$$\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d + \Lambda(\boldsymbol{q}_d - \boldsymbol{q})$$

with $\boldsymbol{q}_d$: *(n+e)x1 vector of desired link coordinates*
$\dot{\boldsymbol{q}}_d$: *(n+e)x1 vector of desired link velocities*
Λ: *(n+e)x(n+e) diagonal, constant, positive definite matrix*

For the reference velocity error we then find

$$\dot{\boldsymbol{e}}_r = (\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \Lambda(\boldsymbol{q}_d - \boldsymbol{q})$$

$$= \dot{\tilde{\boldsymbol{q}}} + \Lambda\tilde{\boldsymbol{q}}$$

with $\tilde{\boldsymbol{q}}$: *(n+e)x1 vector of position errors*
$\dot{\tilde{\boldsymbol{q}}}$: *(n+e)x1 vector of velocity errors*

Analogous to the derivation of the adaptation algorithm in the case of a rigid TR-robot (see section 2.1), the following question arises, and that is how can we arrange for the reference velocity error $\dot{\mathbf{e}}_r$ to tend to $\mathbf{0}$?

We therefore substitute the control law into the equations of motion.This yields

$$\boldsymbol{H}(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_r) + \boldsymbol{C}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_r) + \boldsymbol{g} + \boldsymbol{K}_k\boldsymbol{q} = (\hat{\boldsymbol{H}} - \boldsymbol{H})\ddot{\boldsymbol{q}}_r + (\hat{\boldsymbol{C}} - \boldsymbol{C})\dot{\boldsymbol{q}}_r + \hat{\boldsymbol{g}} + \boldsymbol{K}_k\boldsymbol{q}_r + \kappa\dot{\boldsymbol{e}}_r$$

Define as reference position error vector: $\mathbf{e}_r = \mathbf{q}_r - \mathbf{q}$ and we find for the equivalent error equation of the closed loop system:

$$\boldsymbol{H}\ddot{\boldsymbol{e}}_r + \boldsymbol{C}\dot{\boldsymbol{e}}_r + \boldsymbol{K}_k\boldsymbol{e}_r + \kappa\dot{\boldsymbol{e}}_r = -\left[(\hat{\boldsymbol{H}} - \boldsymbol{H})\ddot{\boldsymbol{q}}_r + (\hat{\boldsymbol{C}} - \boldsymbol{C})\dot{\boldsymbol{q}}_r + (\hat{\boldsymbol{g}} - \boldsymbol{g})\right]$$

$$= -\boldsymbol{W}_f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\tilde{\boldsymbol{p}}$$

with $\tilde{\boldsymbol{p}}(t) = \hat{\boldsymbol{p}}(t) - \boldsymbol{p}$ *rx1 vector of parameter errors*
$\boldsymbol{W}_f$ *(n+e)xr matrix of functions* $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\dot{\boldsymbol{q}}_r$, $\ddot{\boldsymbol{q}}_r$

It must be said that the above derivation is only valid if the error equation of the closed loop system is **linear** in terms of the adjustable parameters $\hat{\mathbf{p}}(t)$ (so - called *linearly parameterized*).

To show global stability of this adaptive control system, we have to find a suitable Lyapunov - function. We use the function:

$$V = \frac{1}{2}[\ \dot{\boldsymbol{e}}_r^T \boldsymbol{H} \dot{\boldsymbol{e}}_r + \boldsymbol{e}_r \boldsymbol{K}_k \boldsymbol{e}_r + \tilde{\boldsymbol{p}}^T \Gamma^{-1} \tilde{\boldsymbol{p}}\ ]$$

with Γ: *rxr diagonal, constant, positive definite matrix*

The first two terms in this expression represent a virtual mechanical energy in the error system (sum of the kinetic and potential energy), while the last term is a positive definite quadratic expression of the parameter error vector $\tilde{\mathbf{p}}(t) = \hat{\mathbf{p}}(t) - \mathbf{p}$.

Differentiating the Lyapunov-function with respect to time yields

$$\dot{V} = \dot{\boldsymbol{e}}_r^T \boldsymbol{H} \ddot{\boldsymbol{e}}_r + \frac{1}{2} \dot{\boldsymbol{e}}_r^T \dot{\boldsymbol{H}} \dot{\boldsymbol{e}}_r + \dot{\boldsymbol{e}}_r^T \boldsymbol{K}_k \boldsymbol{e}_r + \tilde{\boldsymbol{p}}^T \Gamma^{-1} \dot{\tilde{\boldsymbol{p}}}$$

Substituting the error equation of the closed loop system

$$\boldsymbol{H} \ddot{\boldsymbol{e}}_r = -\ [\ \boldsymbol{C} \dot{\boldsymbol{e}}_r + \boldsymbol{K}_k \boldsymbol{e}_r + \kappa \dot{\boldsymbol{e}}_r + \boldsymbol{W}_f\, \tilde{\boldsymbol{p}}\ ]$$

gives

$$\dot{V} = -\dot{\boldsymbol{e}}_r^T \kappa \dot{\boldsymbol{e}}_r + \frac{1}{2} \dot{\boldsymbol{e}}_r^T (\dot{\boldsymbol{H}} - 2\boldsymbol{C}) \dot{\boldsymbol{e}}_r - \dot{\boldsymbol{e}}_r^T \boldsymbol{W}_f\, \tilde{\boldsymbol{p}} + \tilde{\boldsymbol{p}}^T \Gamma^{-1} \dot{\tilde{\boldsymbol{p}}}$$

In [2] - §9.1.2 Slotine and Li prove the skew symmetry of the matrix $\dot{\mathbf{H}}$-2$\mathbf{C}$, which means that for all $\dot{\mathbf{e}}_r$

$$\dot{\boldsymbol{e}}_r^T (\dot{\boldsymbol{H}} - 2\boldsymbol{C}) \dot{\boldsymbol{e}}_r = 0$$

Rearranging the equation for $\dot{V}$ and using the skew symmetry of $\dot{\mathbf{H}}$-2$\mathbf{C}$ yields

$$\dot{V} = -\dot{\boldsymbol{e}}_r^T \kappa \dot{\boldsymbol{e}}_r - \tilde{\boldsymbol{p}}^T (\boldsymbol{W}_f^T \dot{\boldsymbol{e}}_r - \Gamma^{-1} \dot{\tilde{\boldsymbol{p}}})$$

To meet the requirements imposed upon the Lyapunov-function V and its time derivative, the gain matrix κ is chosen positive definite and the second term is chosen equal to 0. This yields the adaptation algorithm for the parameter estimation vector $\hat{\mathbf{p}}(t)$:

$$\dot{\tilde{\boldsymbol{p}}} = \Gamma \boldsymbol{W}_f^T \dot{\boldsymbol{e}}_r$$

By choosing the above described adaptation algorithm and control law, we arranged for the Lyapunov function V($\mathbf{x}$) to be monotonically decreasing. This means that the 'kinetic energy' in the error system constantly decreases. It is essential to notice that then the reference velocity error $\dot{\mathbf{e}}_r$ reduces to zero, and as a consequence, the links are able to track their desired path.

3.3 Calculation of the control input signal $u = [u_s\ u_e]^T$

Assume that the dynamics of a TR-robot with elastic motor transmissions can be given in the following general form

$$H(q,p)\ddot{q} + n(q,\dot{q},p) = Du$$

with

$$n(q,\dot{q},p) = C(q,\dot{q},p)\dot{q} + g(q,p) + K_k q$$

written out

$$\begin{bmatrix} H_{ss} & H_{se} & 0 \\ H_{se} & H_{ee} & 0 \\ 0 & 0 & H_{mm} \end{bmatrix} \begin{bmatrix} \ddot{q}_s \\ \ddot{q}_e \\ \ddot{q}_m \end{bmatrix} + \begin{bmatrix} C_{ss} & C_{se} & 0 \\ C_{es} & C_{ee} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_s \\ \dot{q}_e \\ \dot{q}_m \end{bmatrix} + \begin{bmatrix} g_s \\ g_e \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & K & -K \\ 0 & -K & K \end{bmatrix} \begin{bmatrix} q_s \\ q_e \\ q_m \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u_s \\ u_e \end{bmatrix}$$

with

	$q^T = [q_s\ q_e\ q_m]$:	*(n+e)x1 vector of generalized coordinates*
	q_s:	*(n-e)x1 vector of coordinates of the links driven by a stiff transmission*
	q_e:	*ex1 vector of coordinates of the links driven by an elastic transmission*
	$q_l^T = [q_s\ q_e]$:	*nx1 vector of coordinates defining the positions and orientation of the n rigid links*
	q_m:	*ex1 vector of coordinates defining the rotations of the motor rotors of the actuators with an elastic transmission*

Following the algorithm of Slotine and Li, we choose as adaptive control law

$$Du = \hat{H}(q,\hat{p})\ddot{q}_r + \hat{C}(q,\dot{q},\hat{p})\dot{q}_r + \hat{g}(q,\hat{p}) + K_k q_r + \kappa \dot{e}_r$$

written out

$$\begin{cases} u_s = \begin{bmatrix} \hat{H}_{ss} & \hat{H}_{se} \end{bmatrix} \begin{bmatrix} \ddot{q}_{sr} \\ \ddot{q}_{er} \end{bmatrix} + \begin{bmatrix} \hat{C}_{ss} & \hat{C}_{se} \end{bmatrix} \begin{bmatrix} \dot{q}_{sr} \\ \dot{q}_{er} \end{bmatrix} + \hat{g}_s + \kappa_{ss}\dot{e}_{sr} \\ 0 = \begin{bmatrix} \hat{H}_{se} & \hat{H}_{ee} \end{bmatrix} \begin{bmatrix} \ddot{q}_{sr} \\ \ddot{q}_{er} \end{bmatrix} + \begin{bmatrix} \hat{C}_{es} & \hat{C}_{ee} \end{bmatrix} \begin{bmatrix} \dot{q}_{sr} \\ \dot{q}_{er} \end{bmatrix} + \hat{g}_e + K(q_{er} - q_{mr}) + \kappa_{ee}\dot{e}_{er} \\ u_e = \hat{H}_{mm}\ddot{q}_{mr} - K(q_{er} - q_{mr}) + \kappa_{mm}\dot{e}_{mr} \end{cases}$$

with

	q_{sr}:	*reference trajectory of* q_s
	q_{er}:	*reference trajectory of* q_e
	$q_{lr}^T = [q_{sr}\ q_{er}]$:	*reference trajectory of* q_l
	q_{mr}:	*reference trajectory of* q_m

$\dot{e}_{sr} = \dot{q}_{sr} - \dot{q}_s$ *reference velocity error of* q_s
$\dot{e}_{er} = \dot{q}_{er} - \dot{q}_e$ *reference velocity error of* q_e
$\dot{e}_{mr} = \dot{q}_{mr} - \dot{q}_m$ *reference velocity error of* q_m

Let us first consider the control law $\mathbf{u}_s$ for the **direct driven** links, i.e. the links driven by stiff transmissions. The expression for $\mathbf{u}_s$ is given by

$$u_s = \begin{bmatrix} \hat{H}_{ss} & \hat{H}_{se} \end{bmatrix} \begin{bmatrix} \ddot{q}_{sr} \\ \ddot{q}_{er} \end{bmatrix} + \begin{bmatrix} \hat{C}_{ss} & \hat{C}_{se} \end{bmatrix} \begin{bmatrix} \dot{q}_{sr} \\ \dot{q}_{er} \end{bmatrix} + \hat{g}_s + \kappa_{ss}\dot{e}_{sr}$$

Notice that for the computation of this input signal we need the reference velocities and accelerations of the links $\dot{\mathbf{q}}_{lr}^T = [\dot{\mathbf{q}}_{sr}\ \dot{\mathbf{q}}_{er}]$, respectively $\ddot{\mathbf{q}}_{lr}^T = [\ddot{\mathbf{q}}_{sr}\ \ddot{\mathbf{q}}_{er}]$.
We obtain the expression for the reference velocity by shifting the desired velocity according to the position error, thus

$$\begin{bmatrix} \dot{q}_{sr} \\ \dot{q}_{er} \end{bmatrix} = \begin{bmatrix} \dot{q}_{sd} \\ \dot{q}_{ed} \end{bmatrix} + \Lambda \begin{bmatrix} (q_{sd} - q_s) \\ (q_{ed} - q_e) \end{bmatrix}$$

By differentiating with respect to time we find for the reference accelerations of the links

$$\begin{bmatrix} \ddot{q}_{sr} \\ \ddot{q}_{er} \end{bmatrix} = \begin{bmatrix} \ddot{q}_{sd} \\ \ddot{q}_{ed} \end{bmatrix} + \Lambda \begin{bmatrix} (\dot{q}_{sd} - \dot{q}_s) \\ (\dot{q}_{ed} - \dot{q}_e) \end{bmatrix}$$

and so, the input signal $\mathbf{u}_s$ controlling the direct driven links can be computed.

If we want to compute the control law $\mathbf{u}_e$ for the **elastically driven** links, i.e. the links driven by elastic transmissions, we meet the following problem. The expression for $\mathbf{u}_e$ is given by

$$u_e = \hat{H}_{mm}\ddot{q}_{mr} - K(q_{er} - q_{mr}) + \kappa_{mm}\dot{e}_{mr}$$

Notice that in order to compute $\mathbf{u}_e$ we have to have an expression for the reference trajectory of the motor rotor $\mathbf{q}_{mr}$ and its first and second time derivative. Obtaining the reference velocity $\dot{\mathbf{q}}_{mr}$ by shifting the desired velocity $\dot{\mathbf{q}}_{md}$ according to the position error $(\mathbf{q}_{md}-\mathbf{q}_m)$ as shown above is not possible, simply because we do not have any notion about the desired motor rotor variables. We solve this problem by computing the reference trajectory of the motor rotor from the control law. To do this, we use the equation

$$\mathbf{0} = \begin{bmatrix} \hat{H}_{es} & \hat{H}_{ee} \end{bmatrix} \begin{bmatrix} \ddot{q}_{sr} \\ \ddot{q}_{er} \end{bmatrix} + \begin{bmatrix} \hat{C}_{es} & \hat{C}_{ee} \end{bmatrix} \begin{bmatrix} \dot{q}_{sr} \\ \dot{q}_{er} \end{bmatrix} + \hat{g}_e + K(q_{er} - q_{mr}) + \kappa_{ee}\dot{e}_{er}$$

Writing the elastic transmission torques explicitly yields

$$-\,\boldsymbol{K}(\boldsymbol{q}_{er} - \boldsymbol{q}_{mr}) = \begin{bmatrix} \hat{\boldsymbol{H}}_{es} & \hat{\boldsymbol{H}}_{ee} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_{sr} \\ \ddot{\boldsymbol{q}}_{er} \end{bmatrix} + \begin{bmatrix} \hat{\boldsymbol{C}}_{es} & \hat{\boldsymbol{C}}_{ee} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_{sr} \\ \dot{\boldsymbol{q}}_{er} \end{bmatrix} + \hat{\boldsymbol{g}}_{e} + \kappa_{ee}\dot{\boldsymbol{e}}_{er}$$

Writing the rotation of the motor rotor $\mathbf{q}_{mr}$ explicitly yields

$$\boldsymbol{q}_{mr} = \boldsymbol{K}^{-1}\{ \begin{bmatrix} \hat{\boldsymbol{H}}_{es} & \hat{\boldsymbol{H}}_{ee} \end{bmatrix} \ddot{\boldsymbol{q}}_{lr} + \begin{bmatrix} \hat{\boldsymbol{C}}_{es} & \hat{\boldsymbol{C}}_{ee} \end{bmatrix} \dot{\boldsymbol{q}}_{lr} + \hat{\boldsymbol{g}}_{e} + \kappa_{ee}\dot{\boldsymbol{e}}_{er} \} + \boldsymbol{q}_{er}$$

and by differentiating once, respectively twice

$$\begin{aligned} \dot{\boldsymbol{q}}_{mr} = \boldsymbol{K}^{-1} \{ & \frac{d}{dt}\begin{bmatrix} \hat{\boldsymbol{H}}_{es} & \hat{\boldsymbol{H}}_{ee} \end{bmatrix} \ddot{\boldsymbol{q}}_{lr} + \begin{bmatrix} \hat{\boldsymbol{H}}_{es} & \hat{\boldsymbol{H}}_{ee} \end{bmatrix} \boldsymbol{q}^{(III)}{}_{lr} \\ & + \frac{d}{dt}\begin{bmatrix} \hat{\boldsymbol{C}}_{es} & \hat{\boldsymbol{C}}_{ee} \end{bmatrix} \dot{\boldsymbol{q}}_{lr} + \begin{bmatrix} \hat{\boldsymbol{C}}_{es} + \hat{\boldsymbol{C}}_{ee} \end{bmatrix} \ddot{\boldsymbol{q}}_{lr} \\ & \frac{d}{dt}\hat{\boldsymbol{g}}_{e} + \kappa_{ee}\ddot{\boldsymbol{e}}_{er} \} + \dot{\boldsymbol{q}}_{er} \end{aligned}$$

$$\begin{aligned} \ddot{\boldsymbol{q}}_{mr} = \boldsymbol{K}^{-1}\{ & \frac{d^2}{dt^2}\begin{bmatrix} \hat{\boldsymbol{H}}_{es} & \hat{\boldsymbol{H}}_{ee} \end{bmatrix} \ddot{\boldsymbol{q}}_{lr} + 2\frac{d}{dt}\begin{bmatrix} \hat{\boldsymbol{H}}_{es} & \hat{\boldsymbol{H}}_{ee} \end{bmatrix} \boldsymbol{q}^{(III)}{}_{lr} + \begin{bmatrix} \hat{\boldsymbol{H}}_{es} & \hat{\boldsymbol{H}}_{ee} \end{bmatrix} \boldsymbol{q}^{(IV)}{}_{lr} \\ & + \frac{d^2}{dt^2}\begin{bmatrix} \hat{\boldsymbol{C}}_{es} & \hat{\boldsymbol{C}}_{ee} \end{bmatrix} \dot{\boldsymbol{q}}_{lr} + 2\frac{d}{dt}\begin{bmatrix} \hat{\boldsymbol{C}}_{es} & \hat{\boldsymbol{C}}_{ee} \end{bmatrix} \ddot{\boldsymbol{q}}_{lr} + \begin{bmatrix} \hat{\boldsymbol{C}}_{es} + \hat{\boldsymbol{C}}_{ee} \end{bmatrix} \boldsymbol{q}^{(III)}{}_{lr} \\ & + \frac{d^2}{dt^2}\hat{\boldsymbol{g}}_{e} + \kappa_{ee}\boldsymbol{e}^{(III)}{}_{er} \} + \ddot{\boldsymbol{q}}_{er} \end{aligned}$$

However, before we come to the computation of $\ddot{\mathbf{q}}_{mr}$, we have to make some preliminary calculations. A closer look at the equation for the reference acceleration of the motor rotor $\ddot{\mathbf{q}}_{mr}$, shows that we need an expression for $\mathbf{q}^{(IV)}{}_{lr}$. The equation for $\mathbf{q}^{(IV)}{}_{lr}$ is given by

$$\boldsymbol{q}^{(IV)}{}_{lr} = \boldsymbol{q}^{(IV)}{}_{ld} + \Lambda\,(\boldsymbol{q}^{(III)}{}_{ld} - \boldsymbol{q}^{(III)}{}_{l})$$

The following steps dictate how we come to the calculation of this equation:

1. After partitioning of Lagrange's equations of motion and writing the highest derivative explicitly, we determine the *accelerations* of the links $\ddot{\mathbf{q}}_l$ via

$$\ddot{q}_l = \begin{bmatrix} \ddot{q}_s \\ \ddot{q}_e \end{bmatrix} = \begin{bmatrix} H_{ss} & H_{se} \\ H_{se} & H_{ee} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} u_s \\ 0 \end{bmatrix} - \begin{bmatrix} n_s \\ n_e \end{bmatrix} \right\}$$

then we are able to define

$$q^{(III)}_{lr} = q^{(III)}_{ld} + \Lambda\,(\ddot{q}_{ld} - \ddot{q}_l)$$

$$\ddot{e}_{lr} = \ddot{q}_{lr} - \ddot{q}_l$$

$$\dot{u}_s = \begin{bmatrix} \hat{H}_{ss} & \hat{H}_{se} \end{bmatrix} q^{(III)}_{lr} + \frac{d}{dt}\begin{bmatrix} \hat{H}_{ss} & \hat{H}_{se} \end{bmatrix} \ddot{q}_{lr} + \frac{d}{dt}(\hat{n}_s) + \kappa_{ss}\ddot{e}_{sr}$$

2. After partitioning and differentiation of Lagrange's equations of motion and writing the highest derivative explicitly, we determine the *jerks* of the links $\mathbf{q}^{(III)}_l$ via

$$q^{(III)}_l = \begin{bmatrix} q^{(III)}_s \\ q^{(III)}_e \end{bmatrix} = \begin{bmatrix} H_{ss} & H_{se} \\ H_{se} & H_{ee} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{u}_s \\ 0 \end{bmatrix} - \begin{bmatrix} \dot{n}_s \\ \dot{n}_e \end{bmatrix} - \begin{bmatrix} \dot{H}_{ss} & \dot{H}_{se} \\ \dot{H}_{se} & \dot{H}_{ee} \end{bmatrix} \begin{bmatrix} \ddot{q}_s \\ \ddot{q}_e \end{bmatrix} \right\}$$

then we are able to define

$$q^{(IV)}_{lr} = q^{(IV)}_{ld} + \Lambda\,(q^{(III)}_{ld} - q^{(III)}_l)$$

$$e^{(III)}_{lr} = q^{(III)}_{lr} - q^{(III)}_l$$

All the signal variables necessary to compute $\dot{\mathbf{q}}_{mr}$ and $\ddot{\mathbf{q}}_{mr}$ are available and are used in the computation of the control input signal for the elastically driven links $\mathbf{u}_e$. With the expression derived for $\mathbf{u}_e$ we possess the complete control input signal $\mathbf{u}^T = [\mathbf{u}_s\ \mathbf{u}_e]$. Together with the adaptation algorithm for $\hat{\mathbf{p}}(t)$ we are able to simulate the behaviour of the system under adaptive control.

The structure of the **Adaptive Computed Torque Computed Reference Controller** is sketched in fig. 3.3. The dotted lines and blocks refer to the extensions made in relation to the Basic Algorithm by Slotine and Li (see fig. 2.1).

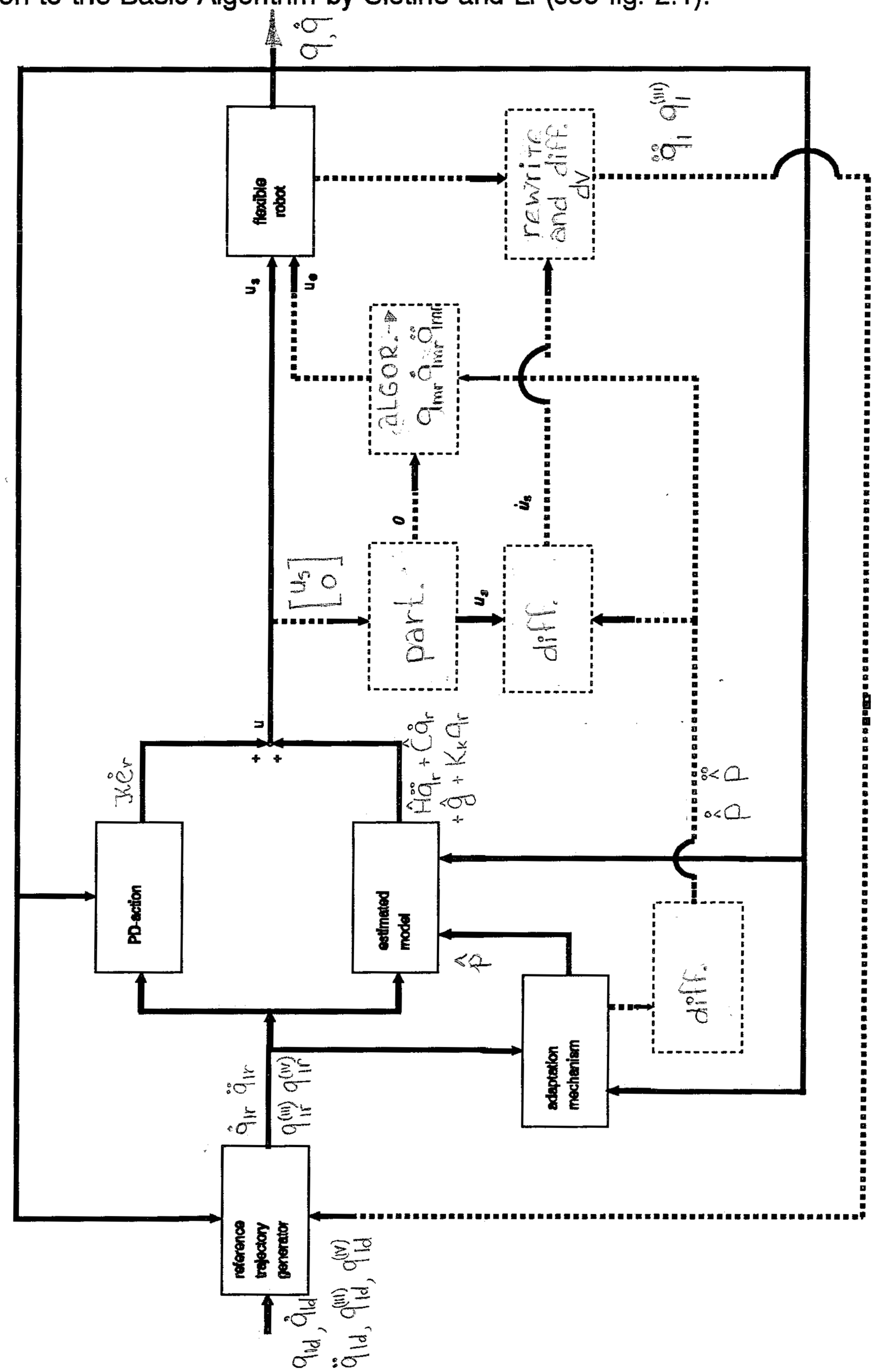


fig. 3.3 Adaptive Computed Torque Computed Reference Controller

3.4 Equation of motion and desired trajectories

The following figure schematically depicts the model of a flexible TR - robot

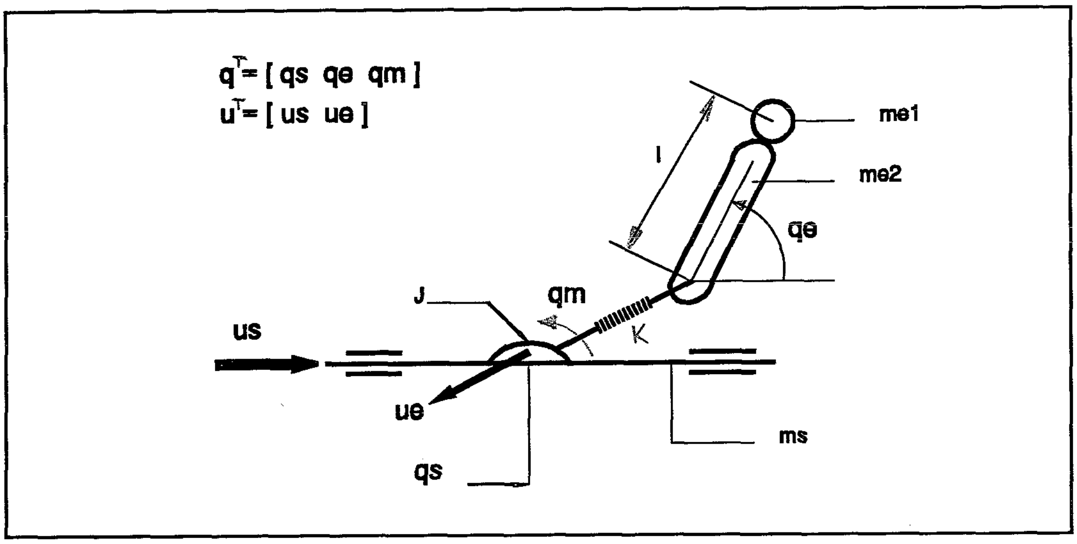


fig. 3.4 flexible TR - robot

m_s:	mass of the carriage
m_{e1}:	mass of the payload at the end of the arm
m_{e2}:	mass of the robot arm
l:	length of the robot arm
J:	inertia of the motor rotor
k:	stiffness of the linear torsional spring
q_s:	horizontal translation of the carriage
q_e:	rotation of the robot arm
q_m:	rotation of the motor rotor
u_s:	force acting on the carriage through a stiff transmission
u_e:	torque acting on the elastic transmission

Using Lagrange's equations, it can be shown that the dynamic equations of the flexible TR - robot are

$$\begin{bmatrix} m_1 & -m_2 l\sin(q_e) & 0 \\ -m_2 l\sin(q_e) & m_3 l^2 & 0 \\ 0 & 0 & J \end{bmatrix} \begin{bmatrix} \ddot{q}_s \\ \ddot{q}_e \\ \ddot{q}_m \end{bmatrix} + \begin{bmatrix} -m_2 l\cos(q_e)\dot{q}_e^2 \\ m_2 lg\cos(q_e) - k(q_m - q_e) \\ k(q_m - q_e) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_s \\ u_e \end{bmatrix}$$

with

$$m_1 = m_{e1} + m_{e2} + m_s$$
$$m_2 = m_{e1} + (1/2)m_{e2}$$
$$m_3 = m_{e1} + (1/3)m_{e2}$$

The desired trajectories to be tracked by the carriage and pendulum are the same as in the case of a rigid TR-robot (see fig. 2.4).

3.5 Adaptive control of a flexible TR - robot

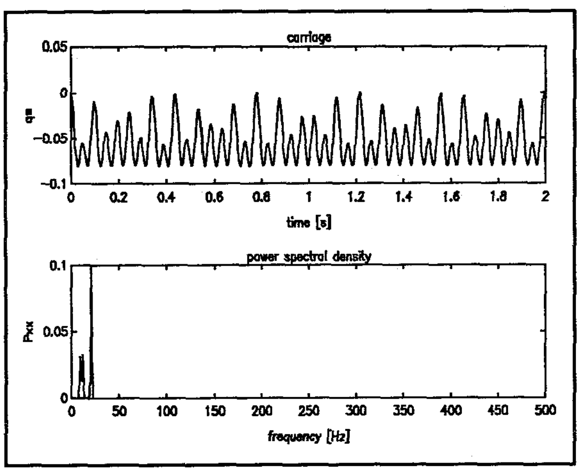
In this section some simulation results are presented. We assume that the pendulum rotates in the horizontal plane (**g** = **0**).

parameters used for simulation

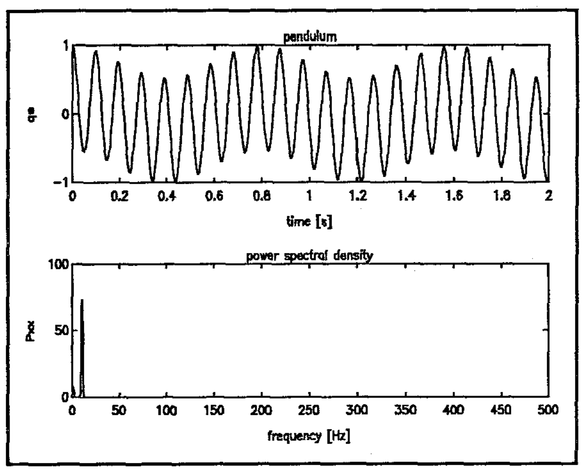
k	= 4980:	spring stiffness revolute joint	[Nm/rad]
m_{e1}	= 2:	mass of the payload	[kg]
m_{e2}	= 3:	mass of the robot arm	[kg]
m_s	= 10:	mass of the carriage	[kg]
J	= 5:	inertia of the motor rotor	[kgm^2]
r	= 0.35:	radius of the circle to be tracked	[m]
l	= 0.75:	length of the robot arm	[m]
ω	= 1:	angular velocity payload	[rad/s]
x_M	= 0:	x-coordinate circle center	[m]
y_M	= l/2:	y-coordinate circle center	[m]

Example 3.1: *uncontrolled flexible TR-robot*

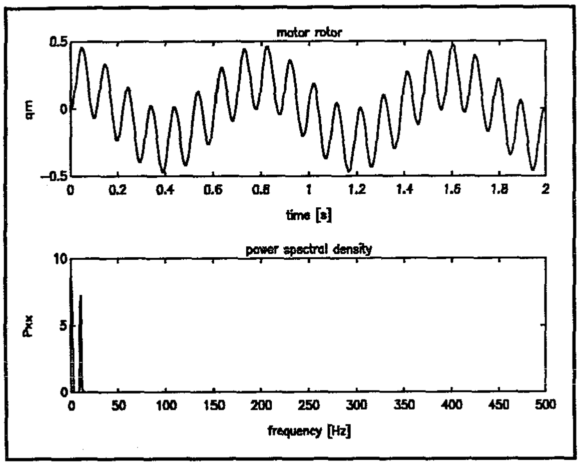
Characteristic for the system behaviour is its free response, i.e. the response generated when there is no control input signal and an initial state unequal to zero. In this example we consider the undamped, free oscillation of the flexible TR-robot. The state vector defining the positions and velocities within the system is chosen to be $\mathbf{x}^T = [q_s\ \dot{q}_s\ q_e\ \dot{q}_e\ q_m\ \dot{q}_m]$. As initial condition we choose $\mathbf{x}^T = [0\ 0\ 1\ 0\ 0\ 0]$. Figure 3.5 shows the displacement of the carriage q_s, the angular rotation of the pendulum q_e, the angular rotation of the motor rotor q_m and the flexibility $(q_m\text{-}q_e)$.



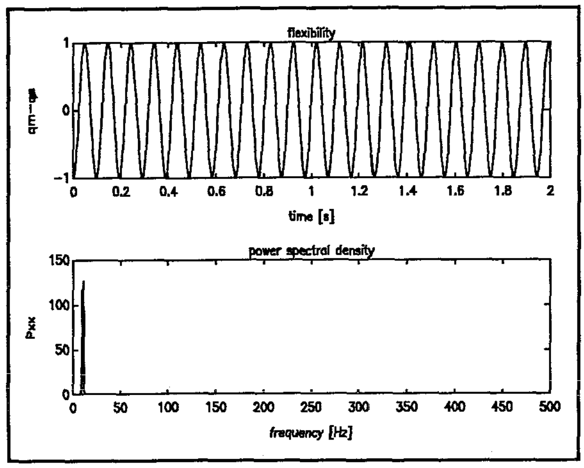
figuur 3.5a



figuur 3.5b



figuur 3.5c



figuur 3.5d

fig. 3.5 undamped, free response of the flexible TR-robot

For the value of the spring stiffness k in the revolute joint we choose: k = 4980 [Nm/rad]. This choice can be motivated as follows:

the state equation describing the dynamics of the system can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$$

with	$\boldsymbol{x}$:	*2(n+e)x1 state vector*
	$\boldsymbol{f}$:	*2(n+e)x1 nonlinear vector function*

Linearization of the original nonlinear state equation at $\boldsymbol{x} = \boldsymbol{0}$ yields

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} \quad \text{with} \quad \boldsymbol{A} = \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right]_{x=0}$$

Analytically computing the eigenvalues of A results in

$$\lambda_{1,2} = 0 \qquad \lambda_{3,4} = 0 \qquad \lambda_{5,6} = \pm j \sqrt[2]{\frac{k(I+J)}{IJ}}$$

with	$I = (m_{e1}+(1/3)me_{e2})l^2$:	*inertia robot arm + payload*
	J:	*inertia motor rotor*

This yields for the period p of the oscillation

$$p = 2\pi \sqrt[2]{\frac{IJ}{k(I+J)}}$$

By chosing k = 4980 [Nm/rad] we obtain for the period p = 0.1 [s], which is equivalent to an undamped frequency f = 10 [Hz].

Simulations of the dynamical behaviour of the *linearized system* show that the numerically obtained period of the oscillation equals the analytically computed period. Simulations of the *local behaviour of the nonlinear system*, i.e. near the equilibrium point $\mathbf{x} = \mathbf{0}$, $\dot{\mathbf{x}} = \mathbf{0}$ ($\| \mathbf{x}(t_0) \| << 1$), show consistency with its linear approximation. As can be seen in fig. 3.5, the response of the system when the initial condition is some distance away from $\mathbf{x} = \mathbf{0}$, $\dot{\mathbf{x}} = \mathbf{0}$ ($\| \mathbf{x}(t_0) \| = 1$), displays that there are more frequency components in the system signals besides the frequency of 10 [Hz]. However, an FFT analysis shows that the highest frequency occuring in the free response is about 20 [Hz] (see fig. 3.5a).

Example 3.2: *controlling a flexible system with a 'rigid' control law*

It is evident that in order for the payload to track a prespecified path, the TR-robot needs to be controlled. In Chapter Two we showed how this control problem for a *rigid manipulator* can be tackled using the Basic Algorithm by Slotine and Li. To be able to cope with the tracking control problem in the case of *flexible manipulators*, the Basic Algorithm was modified into a 'flexible' control algorithm: Computed Torque Computed Reference Control. In this example, we demonstrate what happens if we mistakenly assume that the manipulator is rigid. In other words, we apply a 'rigid' control law to a flexible system. Therefore, we consider the following system consisting of two masses m1 and m2, coupled by a linear elastic spring with stiffness k, and under control by one actuator force u. The control law is designed as if the two-mass-spring system in fig. 3.6 were completely rigid (like in fig. 2.6 with $m = m_1 + m_2$). Goal of the controller is to achieve for m2 to track the desired trajectory according to fig. 2.4b.

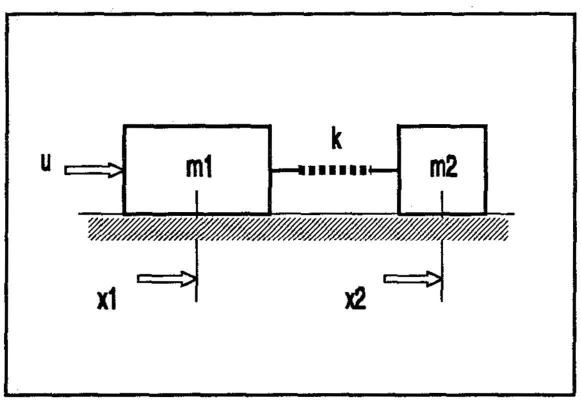


fig. 3.6 flexible system

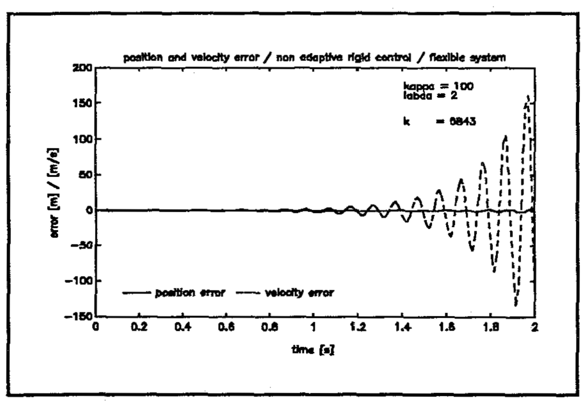


fig. 3.7 rigid control of a flexible system

Notice that the amplitude constantly grows in time; the system under control is unstable. This was to be expected, as the control law totally ignores the extra degree of freedom in the system.

Example 3.3: *control of a flexible system with the non adaptive CTCRC control law*

In this example, we take the elasticity in the transmission into account, in other words, we control the flexible system with the CTCRC control law (see fig. 3.6). Figure 3.8 depicts the simulation result when we possess exact knowledge about the value of the spring stiffness k, i.e there is no parameter uncertainty about k.

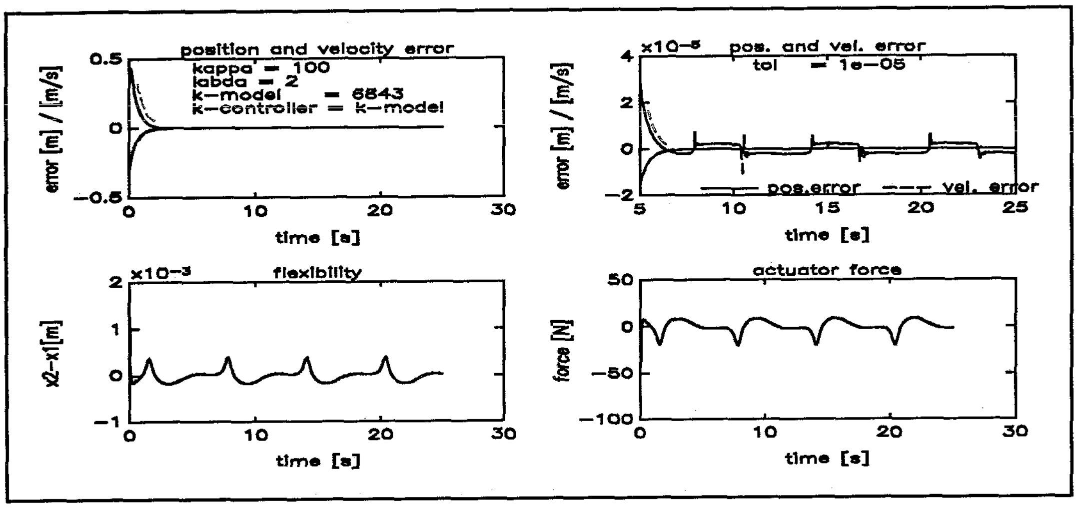


fig. 3.8 k-controller = k-model

Figure 3.9 depicts the result in the case of parameter uncertainty in k. The control law is computed under the assumption that the spring is 5 times more rigid than 'in reality'.

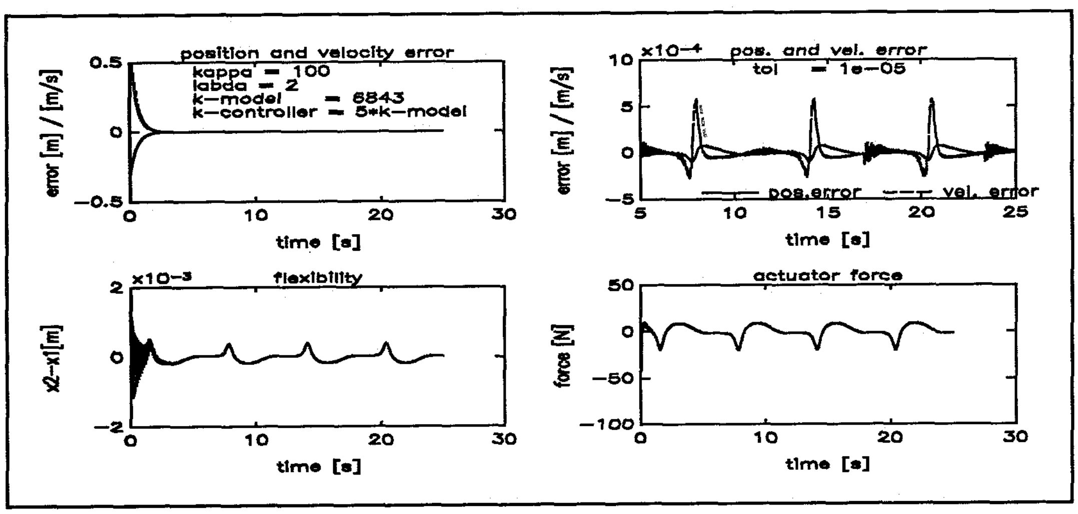
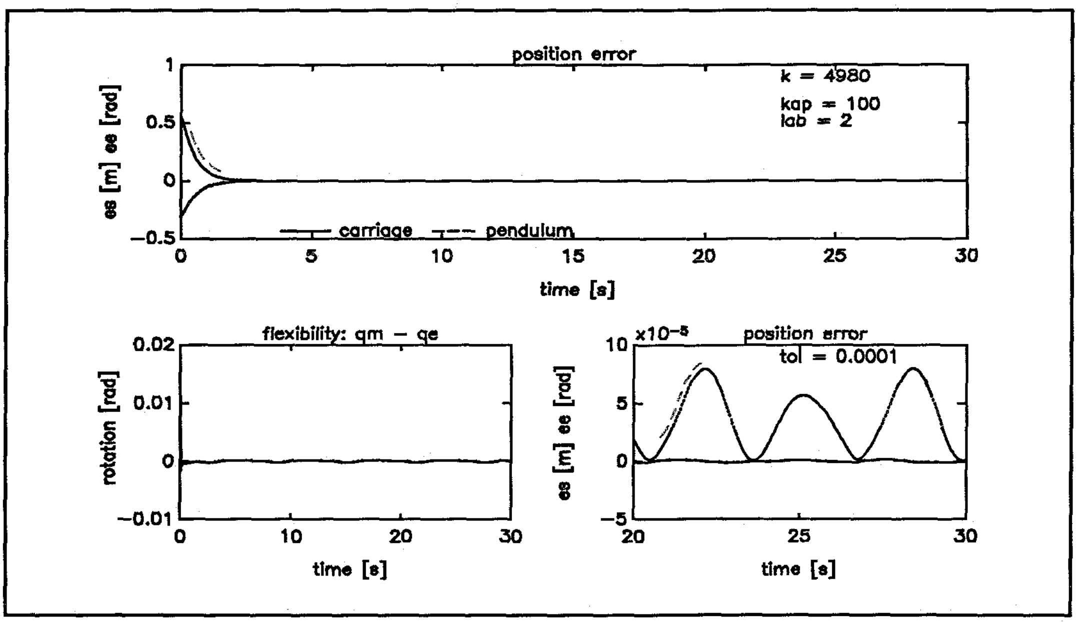


*fig. 3.9 k-controller = 5*k-model*

Compared to the performance in the case of no parameter uncertainty, the result may be considered acceptable; the steady state position error is in the order of magnitude of the tolerance of the integration algorithm. Application of a control law with an adaptation mechanism working on m2, gives similar results as shown in fig. 3.7, 3.8 and 3.9.

Example 3.4: *Non Adaptive Computed Torque Computed Reference Control of a Flexible TR-robot*

In this example, we consider the flexible TR-robot under non-adaptive CTCRC-control. The robot is initially at rest at $q_s=0$ and $q_e=0$ (the desired link trajectories are acc. to fig. 2.4). Notice that the controller effectively handles the trajectory tracking of the carriage and pendulum. Now let us zoom in on the last 10 seconds of the simulation time interval. We see that there remains a 'rest oscillation' whose amplitude is smaller than the tolerance of the integration algorithm (tol = 10^{-4}). Reducing the tolerance to 10^{-6} has no effect on the amplitude of the oscillation. The vibration can be narrowed down by increasing the control gain κ. The flexibility in the system, and thus the torque in the spring $z_e=k(q_m-q_e)$, remains between acceptable bounds.



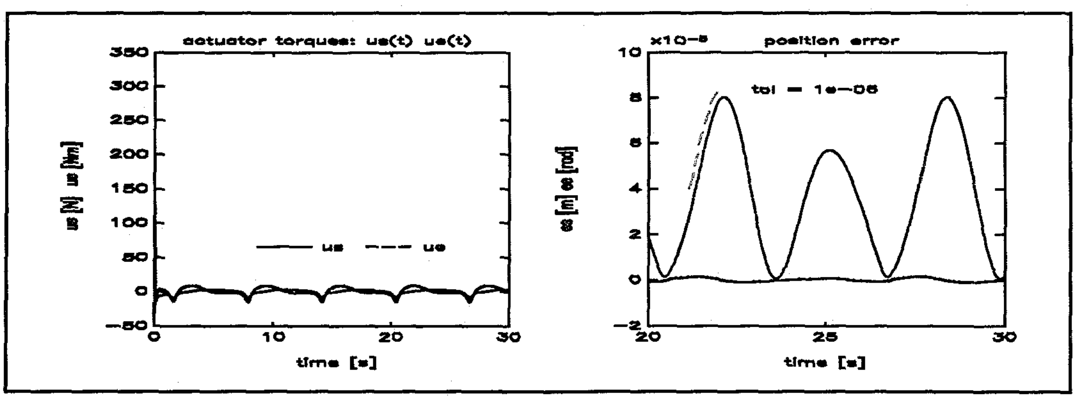
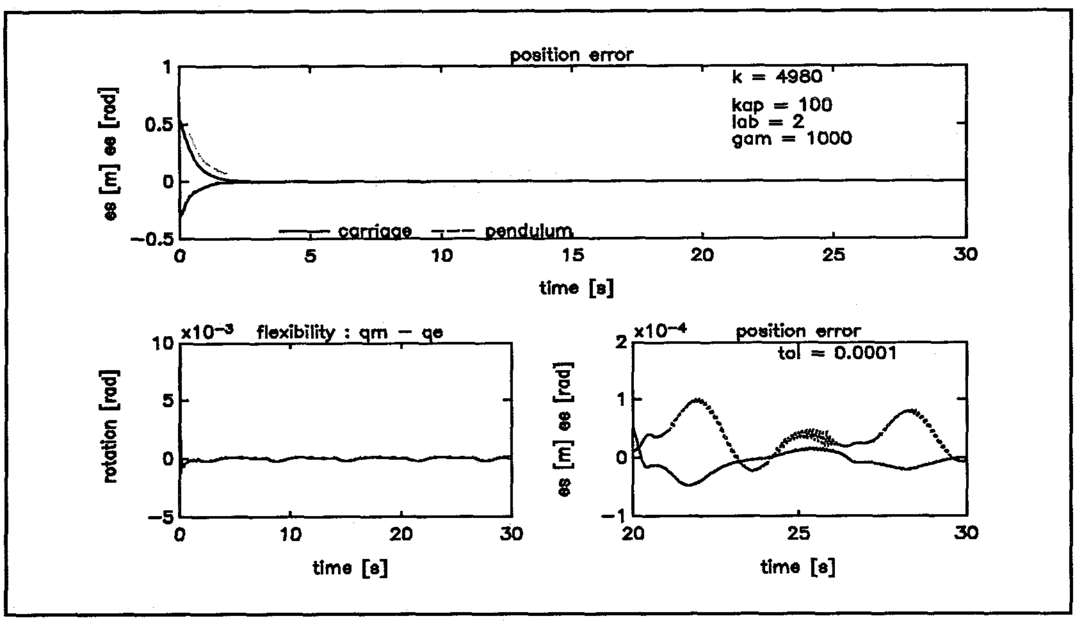


fig. 3.10 / 3.11 non adaptive CTCRC-control of a flexible TR-robot

Example 3.5: *Adaptive Computed Torque Computed Reference Control of a Flexible TR-robot*

Now suppose that we have no a priori knowledge about the mass of the payload m_{e1}. Choosing a value for m_{e1}, and substituting this choice in the non-adaptive control law, would surely lead to an unacceptable system performance (see fig. 2.5b), or even instability. We deal with this parameter uncertainty by applying the adaptive control law developed in section 3.3. Based on the measured system signals (position and velocity of the carriage, pendulum and motor rotor), the mass m_{e1} is estimated on line, and this estimate is used in the control input computation (with $\hat{m}_{e1}(t_0) = 0$). Notice that the adaptive CTCRC controller successfully deals with the tracking control problem of the carriage and pendulum, and that the estimated parameter $\hat{\mathbf{p}}$ converges to the manipulator paramater $\mathbf{p}$ ($\tilde{\mathbf{p}} = \hat{\mathbf{p}} - \mathbf{p} := \mathbf{0}$).

note: bear in mind that the mass of the payload m_{e1} is the *only* unknown systemparameter. Yet, three parameters have to be estimated. Examine the mass matrix in the Lagrange's equations (see section 3.4-page 27), and you will notice that m_{e1} occurs in three expressions in the mass matrix ($\hat{\mathbf{p}}^T = [\hat{m}_1\ \hat{m}_2\ \hat{m}_3]$).



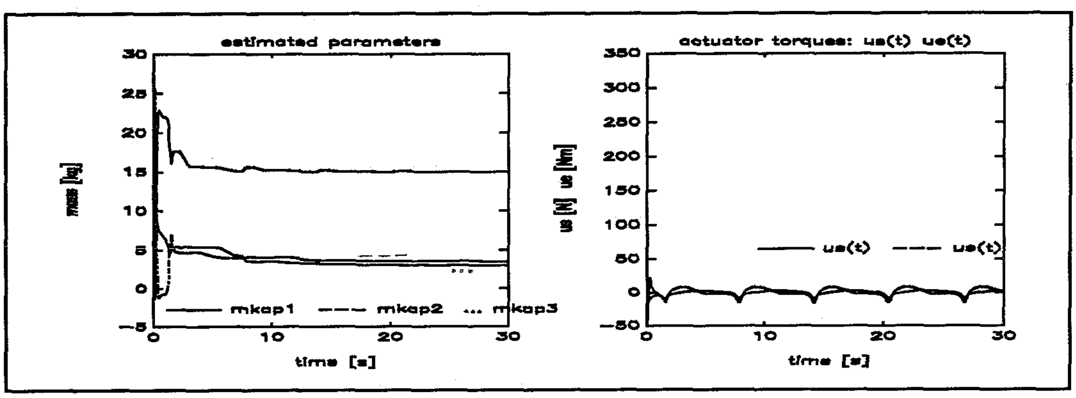


fig. 3.12 / 3.13 adaptive CTCRC control of a flexible TR-robot

3.6 Summary

In Chapter Two, we considered an adaptive control algorithm as proposed by Slotine and Li. The restriction of this algorithm is that it requires the plant to have as many control input signals as there are degrees of freedom to be controlled / stabilized. In the case of robots with elastic motor transmissions, the number of generalized coordinates is greater than the number of control inputs.
In this chapter, we modified the Basic Algorithm by Slotine and Li such that it can be used for controlling flexible robots: **Adaptive Computed Torque Computed Reference Control**. The method has a number of limitations:

1. the error equation of the closed loop system must be linear in terms of the adjustable parameters, but far more important is that
2. the full state vector has to be available

Nevertheless, the new adaptive robot control algorithm is capable of effectively dealing with the tracking control problem of a TR - robot with an elastic transmission between actuator and link. Though the trajectory tracking is not perfect, we consider it as acceptable, because the steady state position error remains in the order of magnitude of the tolerance of the integration algorithm. Due to the elasticity in the system, vibrations occur. These vibrations also remain stabilized.

We may conclude that both the adaptive, as well as the non-adaptive CTCRC controller are capable of making a flexible TR-robot display a prespecified, desired system behaviour. However, the non-adaptive controller requires more or less precise, a priori knowledge about the system parameters. The adaptive controller is able to perform equivalently, but with less information! There is, however, a price we have to pay: the adaptive controller is slower, because the control algorithm is computationally more extensive.

Chapter Four

FURTHER INVESTIGATION INTO ADAPTIVE CONTROL OF A FLEXIBLE TR-ROBOT

Unfortunately, control theory still lacks a general method for obtaining optimal control gains in nonlinear control system design. The simulation results presented in the previous chapters are performed with values for the gains **we** *think are most suitable. In the first section of this chapter, we 'play' with the control gains in order to study their influence on the system's performance. Subsequently, simulations are performed with a more flexible transmission between actuator and link. Then, we consider cases in which the mass of the payload is time-varying, the actuators display saturation characteristics and the spring stiffness k is estimated instead of the mass of the payload. We end this chapter with the presentation of an equivalent, but computationally less extensive algorithm. It is less extensive, because we numerically differentiate the computed references twice, instead of computing their first and second time-derivative exactly in an analytical and time-consuming way.*

4.1 'Playing' with the control gains κ, Λ and Γ

In example 2.2 we considered the adaptive control of a rigid mass. We stated that the performance of the system strongly depends on an appropriate tuning of the control gains. These statements, of course, also hold for the control of flexible systems. The following question now arises, and that is, if a general theory for providing optimal control gains is not available, how can we find 'properly chosen' control settings? First, we linearize the system around its equilibrium point $\mathbf{x} = \mathbf{0}$, $\dot{\mathbf{x}} = \mathbf{0}$. Then, we determine an optimal control law $\mathbf{u}^0$, i.e. in our case a state feedback law $\mathbf{u}^0 = -\mathbf{L}^0\mathbf{x}$, which, when applied as the system input, optimizes the system's performance with respect to some cost criterion. Next, we 'convert' the state feedback matrix $\mathbf{L}^0$ into our control gains κ and Λ, and apply these values in the control law for the nonlinear system. Seen from a mathematical point of view, this procedure is not correct, but it yields an initial value from which we can continue the process of finding 'proper' values for the control gains by means of trial-and-error. Finding a 'proper' value for Γ is purely a matter of trial-and-error.

Example 4.1: *influence of* Λ

The factor Λ determines how fast the position error $\tilde{\mathbf{q}}$ and the velocity error $\mathring{\tilde{\mathbf{q}}}$ damp out once the reference velocity error $\dot{\mathbf{e}}_r$ is zero. The following simulation results confirm this statement.

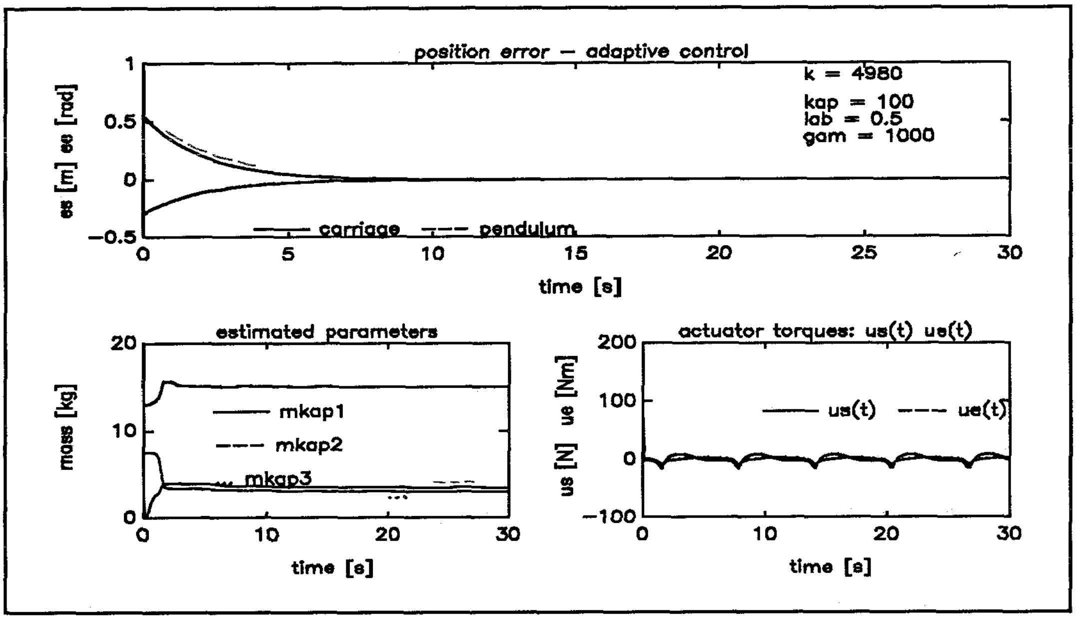


fig. 4.1 Λ *= 0.5I*

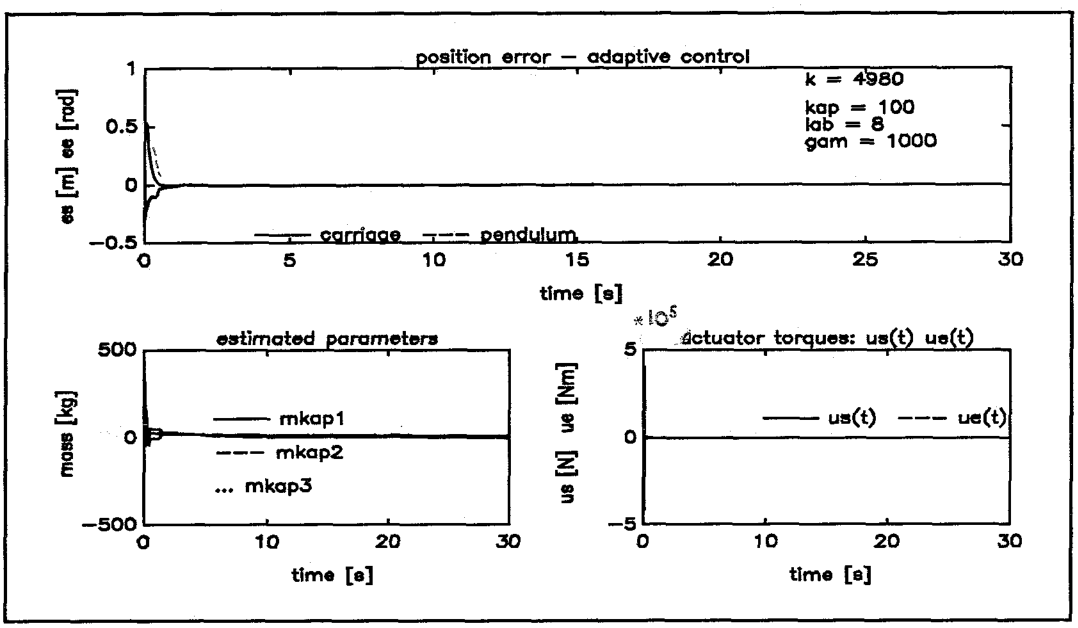


fig. 4.2 Λ *= 8I*

Notice that for Λ = 0.5**I** the convergence time for the position error lasts longer compared to the simulation in which Λ = 8**I**.

Example 4.2: *influence of* κ

By increasing κ we arrange for the system to come more quickly into sliding motion. That is just what we want, because then the position error can damp out as demonstrated in the previous example. The next simulation results show that the control gain κ determines how fast the reference velocity error $\dot{\mathbf{e}}_r$ is reduced to zero.

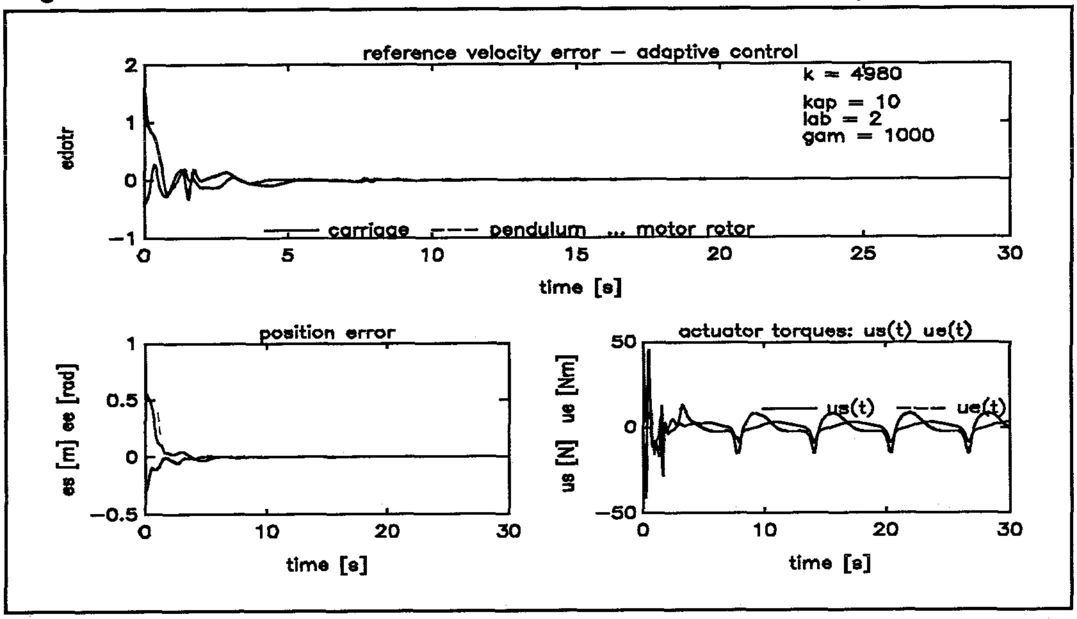


fig. 4.3 κ = *10I*

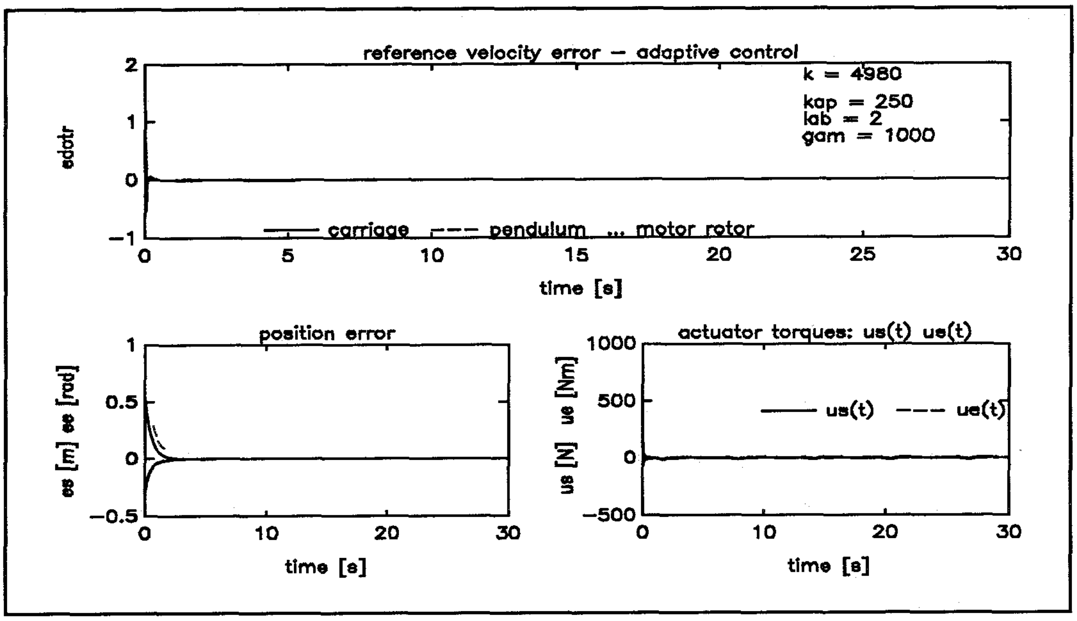


fig. 4.4 κ = *250I*

In example 2.2 we already demonstrated the effects of a change in Γ. Simulations showed that Γ determined at what rate the estimated parameters are adjusted.

4.2 An alternative value for the spring stiffness k

In example 3.1, we motivated our choice for k = 4980 [Nm/rad]. But, can the control system also cope with the trajectory tracking problem for a TR - robot with a more flexible transmission between actuator and link? To answer this question we simulate the behaviour of the system for k = 199.2 [Nm/rad].

Example 4.3: *non-adaptive / adaptive controller k = 199.2 [Nm/rad]*

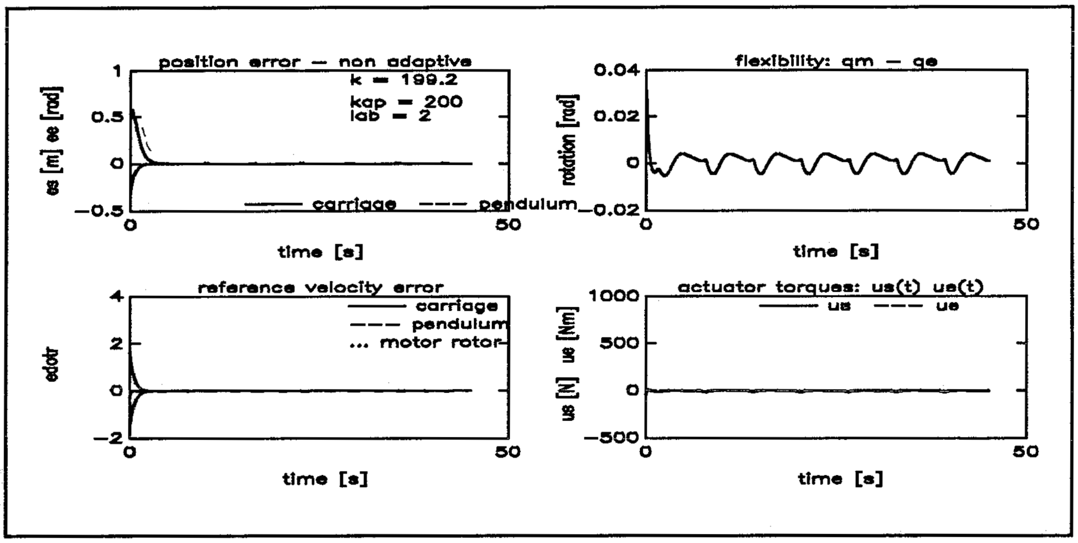


fig. 4.5 non adaptive: k = 199.2

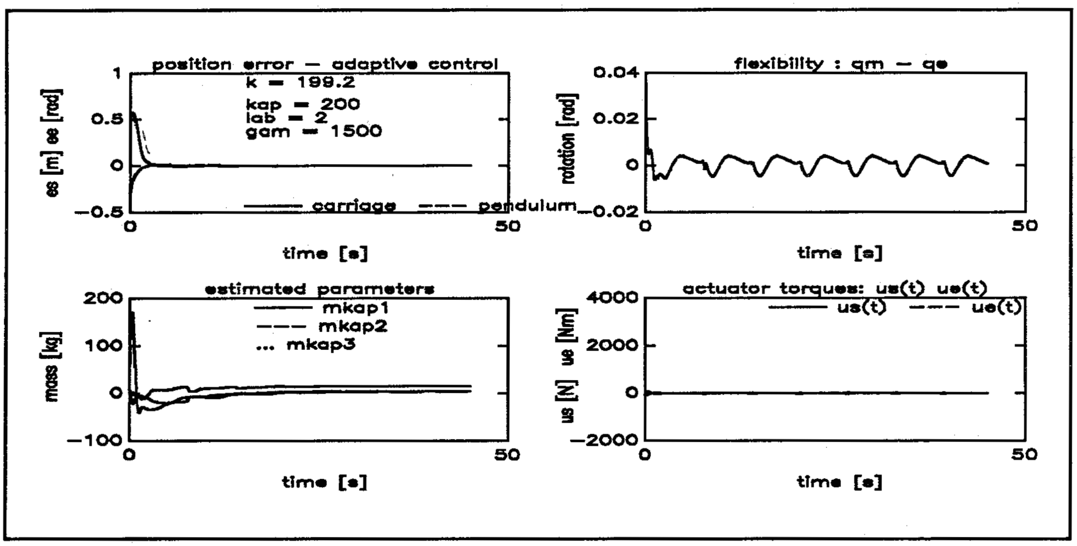


fig. 4.6 adaptive: k = 199.2

As the efforts which the actuators have to make to arrange for a satisfactory system behaviour are too great at $t = t_0$, we conclude that the more flexible the transmission between actuator and link, the harder it is to accomplish a reasonable trajectory tracking and hereby limiting the occuring elastic vibrations. The same conclusion holds for the adaptive controller.

4.3 Time - varying mass of the payload

As an adaptive controller has to process more information than its non-adaptive equivalent, it is slower. Now, if the plantparameters vary considerably faster than the parameter adaptation, it is difficult for the adaptive controller to keep up. In order for an adaptive control system to perform satisfactorily, the basic assumption that the unknown plantparameters are *constant* or *slowly time-varying* has to be made. In this section, we consider the payload to be a time-varying mass.

The mass m_{e1} progresses in time according to

$$m_{e1} = m_{e1}(1 + \cos(\omega t))$$

with $\omega_1 = 2\pi / 120$: slowly time-varying
$\omega_2 = 2\pi / 6$: rapidly time-varying

Example 4.4: *adaptive controller: slowly / rapidly time-varying mass*

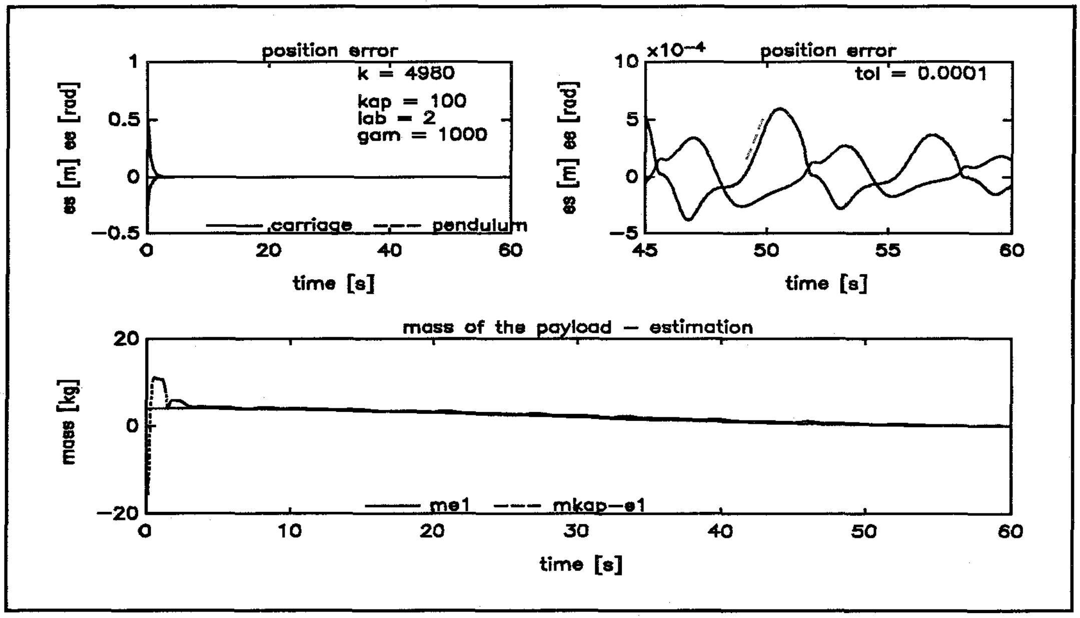


fig. 4.7 slowly-time varying mass

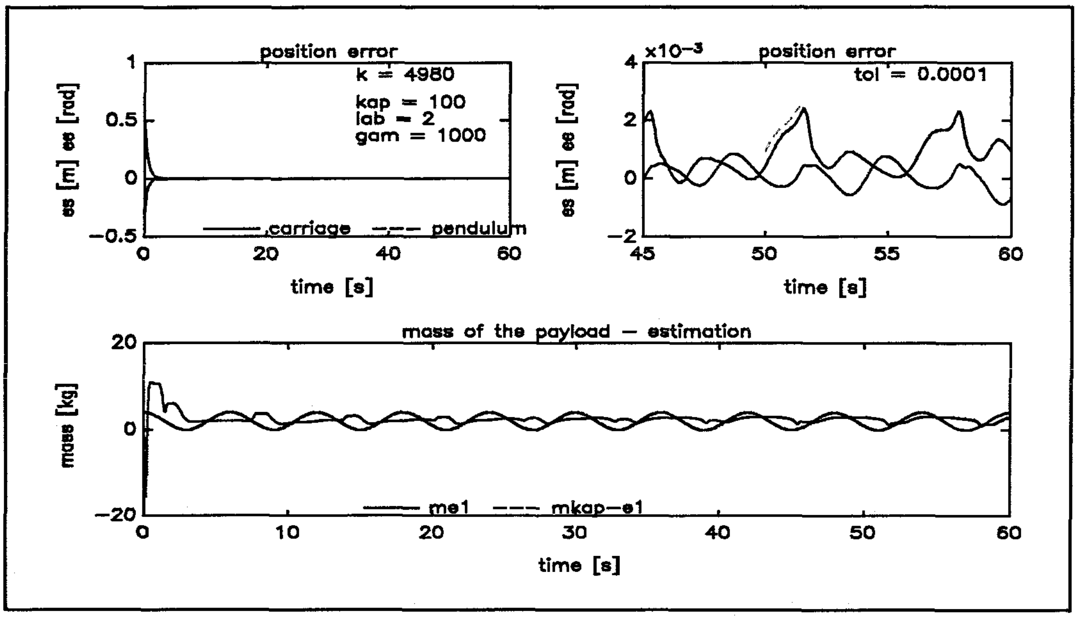


fig. 4.8 rapidly time-varying mass

Notice that the adaptation mechanism is still able to estimate the mass of the payload, despite of the fact that it varies slowly with time (fig. 4.7). However, if the mass of the payload changes more rapidly in time, the adaptation mechanism cannot keep up; a correct estimate fails to appear (fig. 4.8).

4.4 Actuator saturance

Most actuators display saturation characteristics. This means that when the input signal for the actuator reaches a certain level, an extra increase of the input hardly leads to an extra increase of the output. The output remains around a certain maximum value. In this section, we simulate the behaviour of the TR-robot when the actuator torques are limited to

$$|u_{s\text{-}max}| = 15 \text{ [N]}$$
$$|u_{e\text{-}max}| = 10 \text{ [Nm]}$$

Example 4.5: *non-adaptive / adaptive controller with actuator saturance*

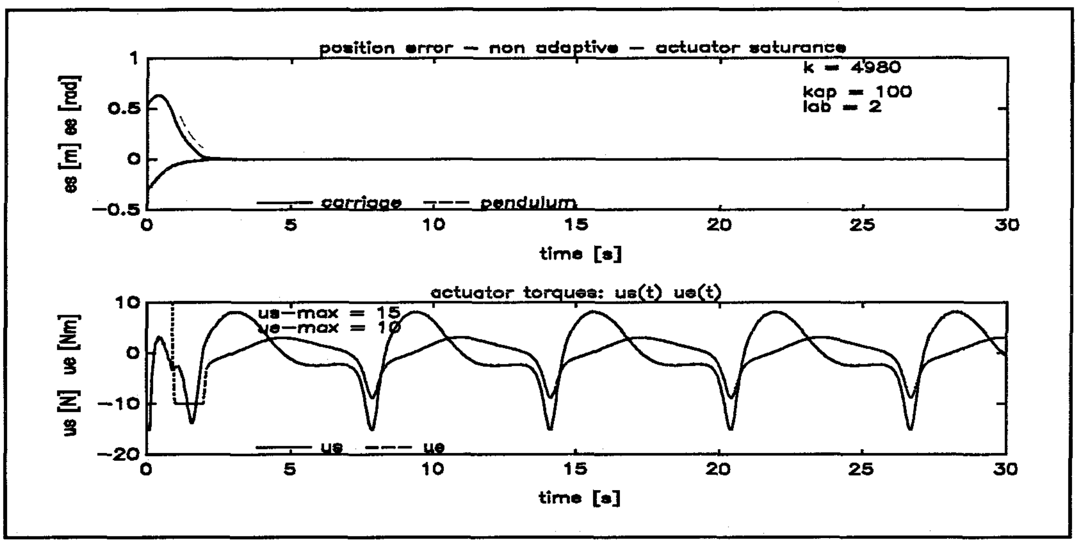


fig. 4.9 non-adaptive controller

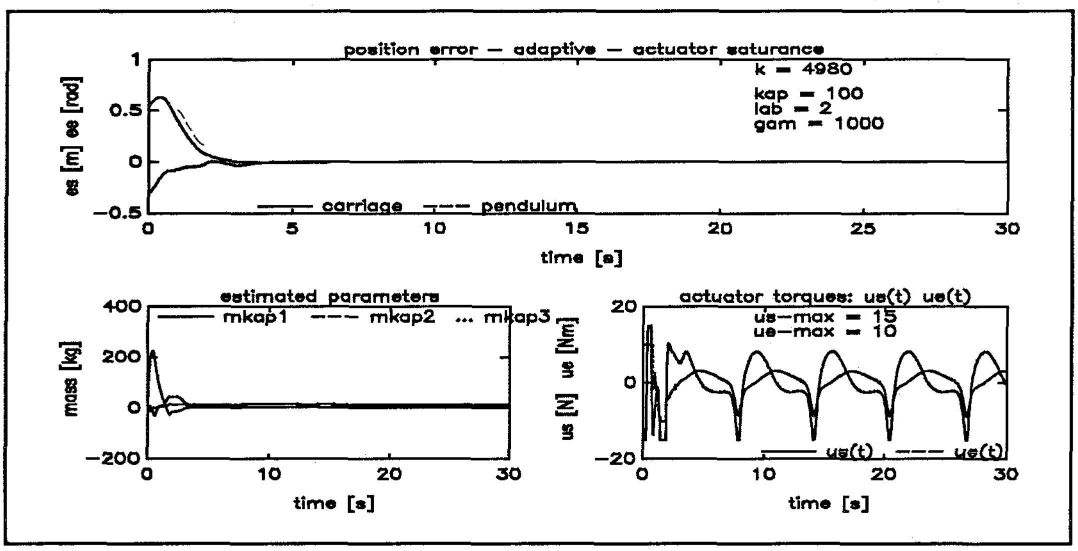


fig. 4.10 adaptive controller

Note that the problem of tracking control is effectively handled in both the adaptive, as well as in the non-adaptive case. The difference in the amount of time necessary for the position error to reduce to zero is negligible compared to the case with no actuator saturance (compare fig. 4.9 vs. fig. 3.10, and fig. 4.10 vs. fig. 3.12).

4.5 Estimation of the spring stiffness k

Up to now, we regarded the mass of the payload m_{e1} as the unknown system parameter. Suppose there is no uncertainty on the mass properties, but that the spring stiffness k of the elastic revolute joint is not exactly known. Is the controller then able to accomplish a reasonable trajectory tracking? Does the adaptation mechanism yield a correct estimate for the unknown parameter? To answer these questions, we consider the trajectory control of the flexible TR-robot in the case of uncertainty in k.

An extra complication arises if we want to tackle the above described control problem. The adaptation algorithm for $\hat{\mathbf{m}}$ yielded

$$\mathring{\hat{m}} = \alpha\,(q_e, \dot{q}_e, \dot{q}_{er}, \ddot{q}_{er}, \ddot{q}_{sr})$$

We simply found the reference velocity by shifting the desired velocity according to the position error, and the reference acceleration by shifting the desired acceleration according to the velocity error. If we derive the adaptation algorithm for $\hat{k}$ we find

$$\mathring{\hat{k}} = \beta\,(q_{er}, \dot{q}_{er}, \dot{q}_e, q_{mr}, \dot{q}_{mr}, \dot{q}_m)$$

Deducing the expression for $\dot{\mathbf{q}}_{mr}$ yields

$$\dot{q}_{mr} = \gamma\,(\hat{k}, \mathring{\hat{k}}, \ddot{q}_e, \dot{q}_{er}, \ddot{q}_{er}, q^{(III)}_{er}, q^{(III)}_{sr})$$

So in order to compute $\mathring{\hat{\mathbf{k}}}$ we need $\dot{\mathbf{q}}_{mr}$, but to compute $\dot{\mathbf{q}}_{mr}$ we need $\mathring{\hat{\mathbf{k}}}$!!! To avoid the issue of (in)stability, we neglect $\mathring{\hat{\mathbf{k}}}$ in the computation of $\dot{\mathbf{q}}_{mr}$, thus

$$\dot{q}_{mr} = \delta\,(\hat{k}, \ddot{q}_e, \dot{q}_{er}, \ddot{q}_{er}, q^{(III)}_{er}, q^{(III)}_{sr})$$

We consider this neglect as allowed, because *in the long run* $\hat{\mathbf{k}}$ must become constant, thus $\mathring{\hat{\mathbf{k}}} := \mathbf{0}$.

Example 4.6: *estimation of k*

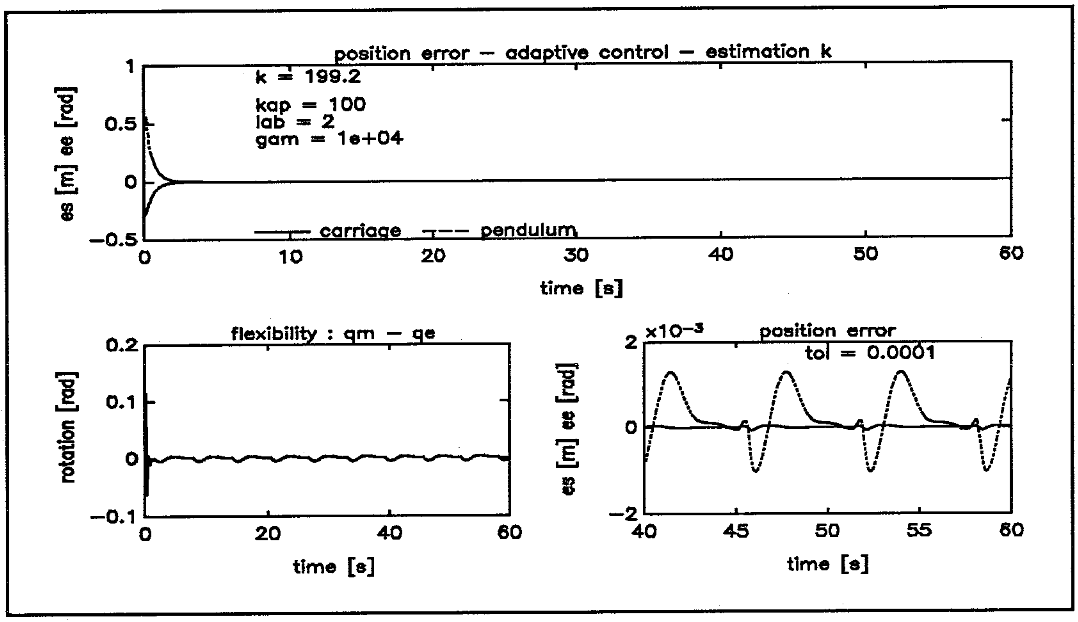
As state vector **x** we choose

$$\mathbf{x}^T = [\, q_s \; \dot{q}_s \; q_e \; \dot{q}_e \; q_m \; \dot{q}_m \; \hat{k} \; q_{mr} \; q_{er} \; q_{sr} \,]$$

We initiate the simulation with

$$\mathbf{x}^T(t_0{=}0) = [\, 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1000 \; 0 \; 0 \; 0 \,]$$

Simulation result:



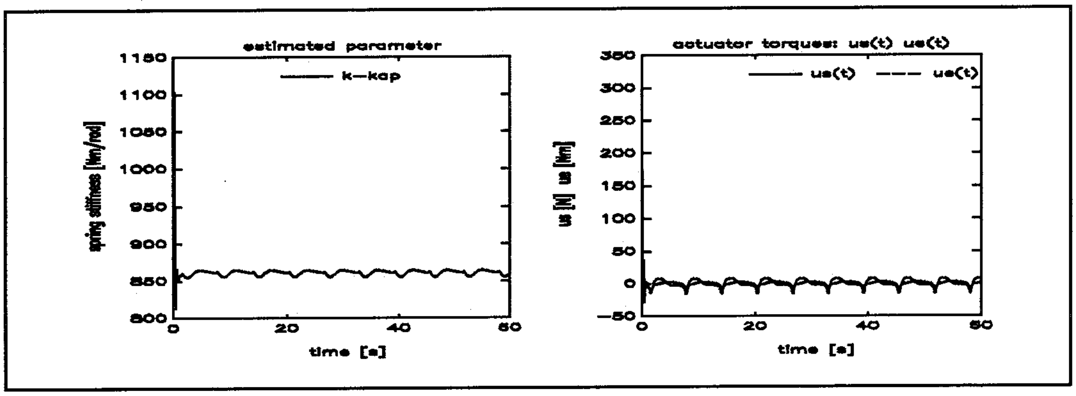


fig. 4.11 / 4.12 adaptation mechanism working on k

Notice that the trajectory control problem is effectively dealt with and that the flexibility in the system remains limited. For a controller tackling trajectory control problems, this is the decisive criterion. However, a correct estimate of the spring stiffness k fails to appear. This is in contrast with the simulation in which the mass of the payload is estimated. Appearantly, the control input signal does not contain sufficient information to provide for a correct estimate (the input signal is not persistently exciting).

4.6 Simplification of the control algorithm

To be able to apply the control algorithm presented in Chapter Three, we have to derive an expression for the accelerations and jerks of the links from the differential equation describing the dynamics of the plant. Often, however, the model describing the system under control is so complex, that analytically deducing an expression for the accelerations and jerks of the links, is practically unfeasible. In this section, we demonstrate how the (non-)adaptive control problem for a flexible TR-robot is tackled, using an approximation scheme for the calculation of some system variables.

simplified algorithm

The full state vector, i.e. the positions and velocities of the links, has to be available at all times and the desired trajectories have to be known up to a sufficiently high order. The adaptation algorithm remains unmodified.

The time derivative of a signal is computed with the following two-points differentiation scheme:

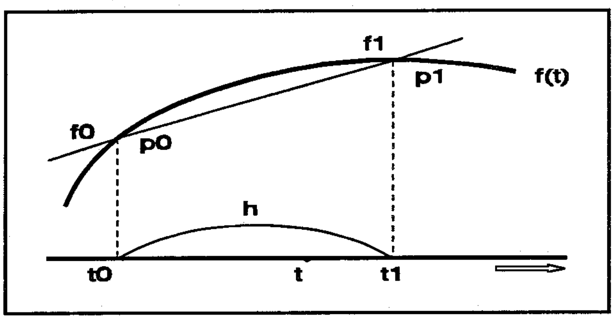


fig. 4.13 linear approximation

Lagrange's interpolation polynom [4]:

$$p(t) = \frac{t - t_1}{t_0 - t_1} f_0 + \frac{t - t_0}{t_1 - t_0} f_1 \qquad with \quad t_0 \leq t \leq t_1$$

After differentiation we find

$$\dot{p}(t) = \frac{1}{h} (f_1 - f_0) \qquad with \quad h = t_1 - t_0$$

The control input signal is computed as follows:
We obtain the *reference velocity of the links* by shifting the desired velocity according to the position error

$$\begin{bmatrix} \dot{q}_{sr} \\ \dot{q}_{er} \end{bmatrix}_{t=t_n} = \begin{bmatrix} \dot{q}_{sd} \\ \dot{q}_{ed} \end{bmatrix}_{t=t_n} + \Lambda \begin{bmatrix} (q_{sd} - q_s) \\ (q_{ed} - q_e) \end{bmatrix}_{t=t_n}$$

For the *reference acceleration of the links* we find

$$\begin{bmatrix} \ddot{q}_{sr} \\ \ddot{q}_{er} \end{bmatrix}_{t=t_n} = \begin{bmatrix} \ddot{q}_{sd} \\ \ddot{q}_{ed} \end{bmatrix}_{t=t_n} + \Lambda \begin{bmatrix} (\dot{q}_{sd} - \dot{q}_s) \\ (\dot{q}_{ed} - \dot{q}_e) \end{bmatrix}_{t=t_n}$$

The *accelerations of the links* are computed via the two-points differentiation scheme:

$$\ddot{q}_s(t_n) = \frac{\dot{q}_s(t_n) - \dot{q}_s(t_{n-1})}{t_n - t_{n-1}}$$

$$\ddot{q}_e(t_n) = \frac{\dot{q}_e(t_n) - \dot{q}_e(t_{n-1})}{t_n - t_{n-1}}$$

The *reference jerks of the links* are given by

$$\begin{bmatrix} q^{(III)}{}_{sr} \\ q^{(III)}{}_{er} \end{bmatrix}_{t=t_n} = \begin{bmatrix} q^{(III)}{}_{sd} \\ q^{(III)}{}_{ed} \end{bmatrix}_{t=t_n} + \Lambda \begin{bmatrix} (\ddot{q}_{sd} - \ddot{q}_s) \\ (\ddot{q}_{ed} - \ddot{q}_e) \end{bmatrix}_{t=t_n}$$

The *reference velocity of the motor rotor* is computed through

$$\dot{q}_{mr} = K^{-1} \{ \frac{d}{dt}\begin{bmatrix} \hat{H}_{es} & \hat{H}_{ee} \end{bmatrix} \ddot{q}_{lr} + \begin{bmatrix} \hat{H}_{es} & \hat{H}_{ee} \end{bmatrix} q^{(III)}{}_{lr}$$

$$+ \frac{d}{dt}\begin{bmatrix} \hat{C}_{es} & \hat{C}_{ee} \end{bmatrix} \dot{q}_{lr} + \begin{bmatrix} \hat{C}_{es} + \hat{C}_{ee} \end{bmatrix} \ddot{q}_{lr}$$

$$\frac{d}{dt}\hat{g}_e + \kappa_{ee}\ddot{e}_{er}\} + \dot{q}_{er}$$

For the *reference acceleration of the motor rotor* we find

$$\ddot{q}_{mr}(t=t_n) = \frac{\dot{q}_{mr}(t_n) - \dot{q}_{mr}(t_{n-1})}{t_n - t_{n-1}}$$

Now, we are able to compute the control law for the *direct driven links* via

$$u_s = \begin{bmatrix} \hat{H}_{ss} & \hat{H}_{se} \end{bmatrix} \begin{bmatrix} \ddot{q}_{sr} \\ \ddot{q}_{er} \end{bmatrix} + \begin{bmatrix} \hat{C}_{ss} & \hat{C}_{se} \end{bmatrix} \begin{bmatrix} \dot{q}_{sr} \\ \dot{q}_{er} \end{bmatrix} + \hat{g}_s + \kappa_{ss}\dot{e}_{sr}$$

and for the *elastically driven* links

$$u_e = \hat{H}_{mm}\ddot{q}_{mr} - K(q_{er} - q_{mr}) + \kappa_{mm}\dot{e}_{mr}$$

The purpose of the following simulations is to examine which differences occur, when we use the approximation scheme presented above, instead of the 'exact' scheme presented in Chapter Three. We first consider the non-adaptive controller.

Example 4.7: *non adaptive control using an 'exact' algorithm*

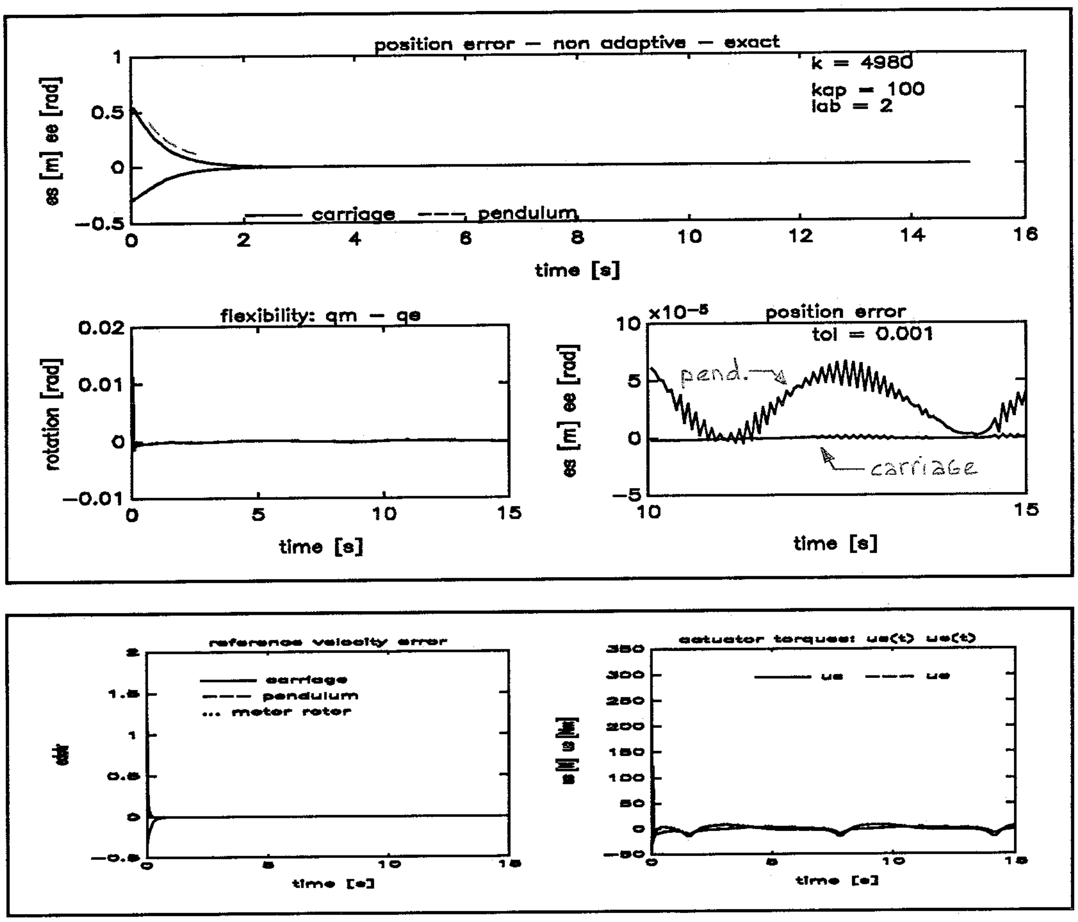
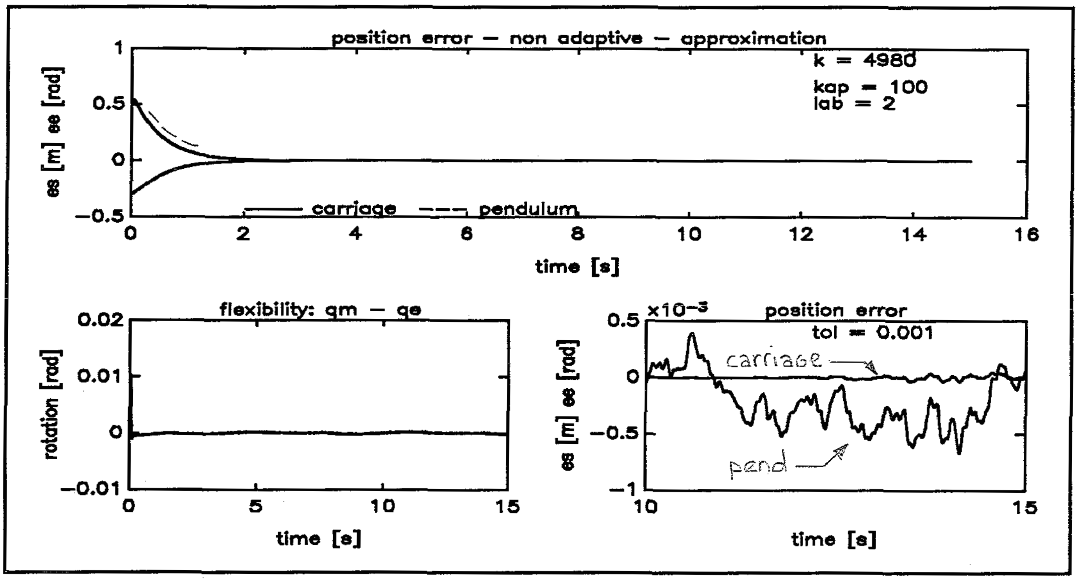


fig. 4.14 / 4.15 non adaptive exact

Example 4.8: *non adaptive control using an 'approximated' algorithm*



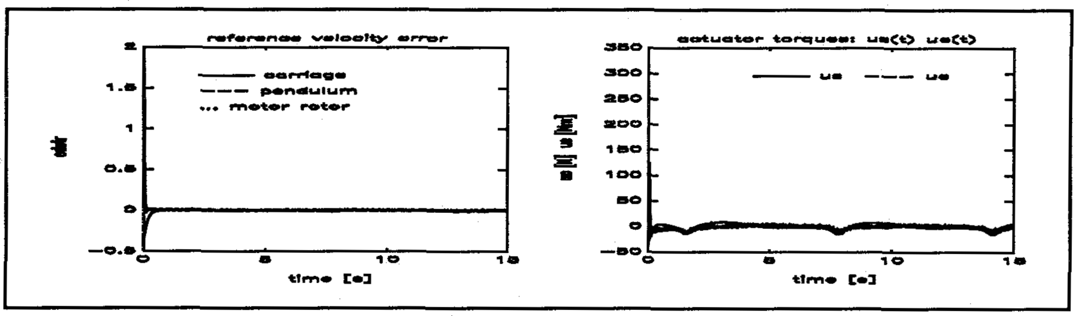


fig. 4.16 / 4.17 non adaptive approximated

At first sight, it looks like that there is no difference in the performance of the controller. However, closely examining the plots displaying the steady state position error shows that in the 'exact' case the position error remains within the tolerance range of the integration algorithm, while in the 'approximated' case the position error is about 5 times larger.

Comparing the simulation results for the adaptive controller, leads to the same conclusions as for the non-adaptive controller, namely the use of an approximation technique leads to a slight performance degradation in the sense of less smaller steady state position errors.

Example 4.9: *adaptive control using an 'exact' algorithm*

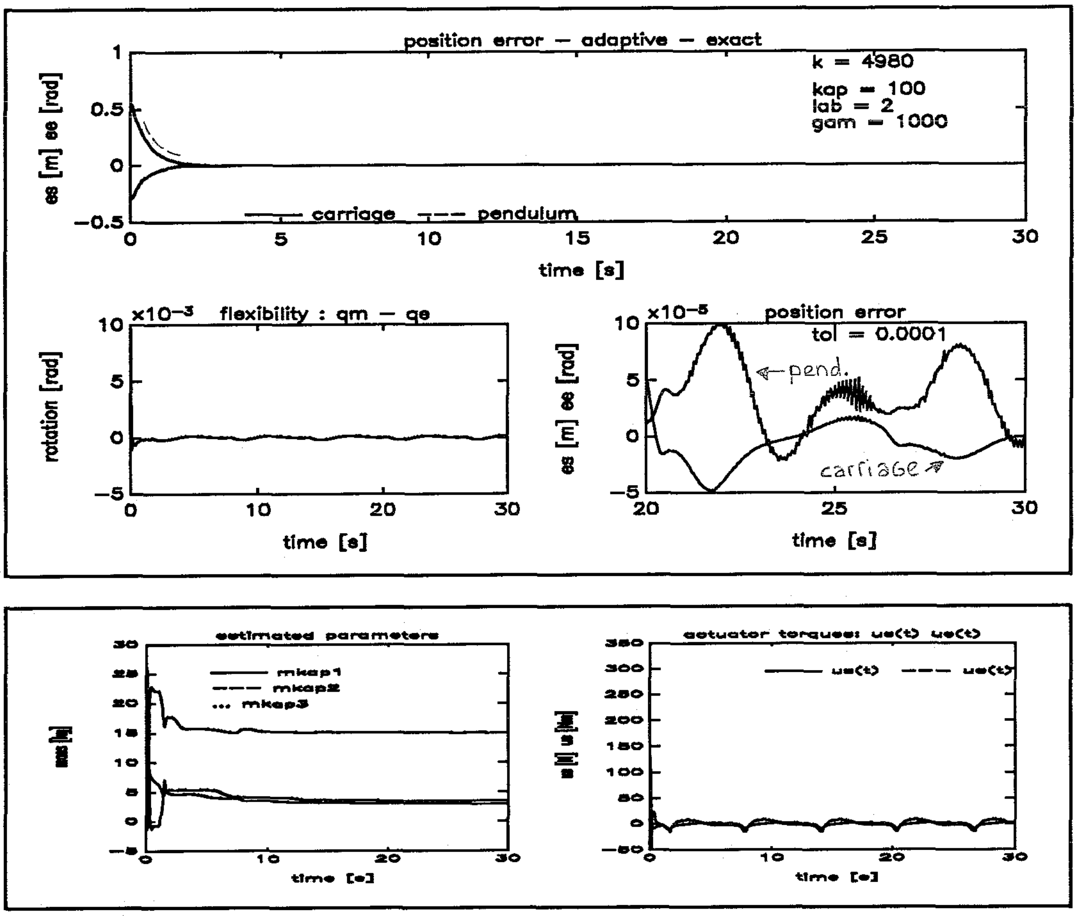


fig. 4.18 / 4.19 adaptive exact

Example 4.10: *adaptive control using an 'approximated' algorithm*

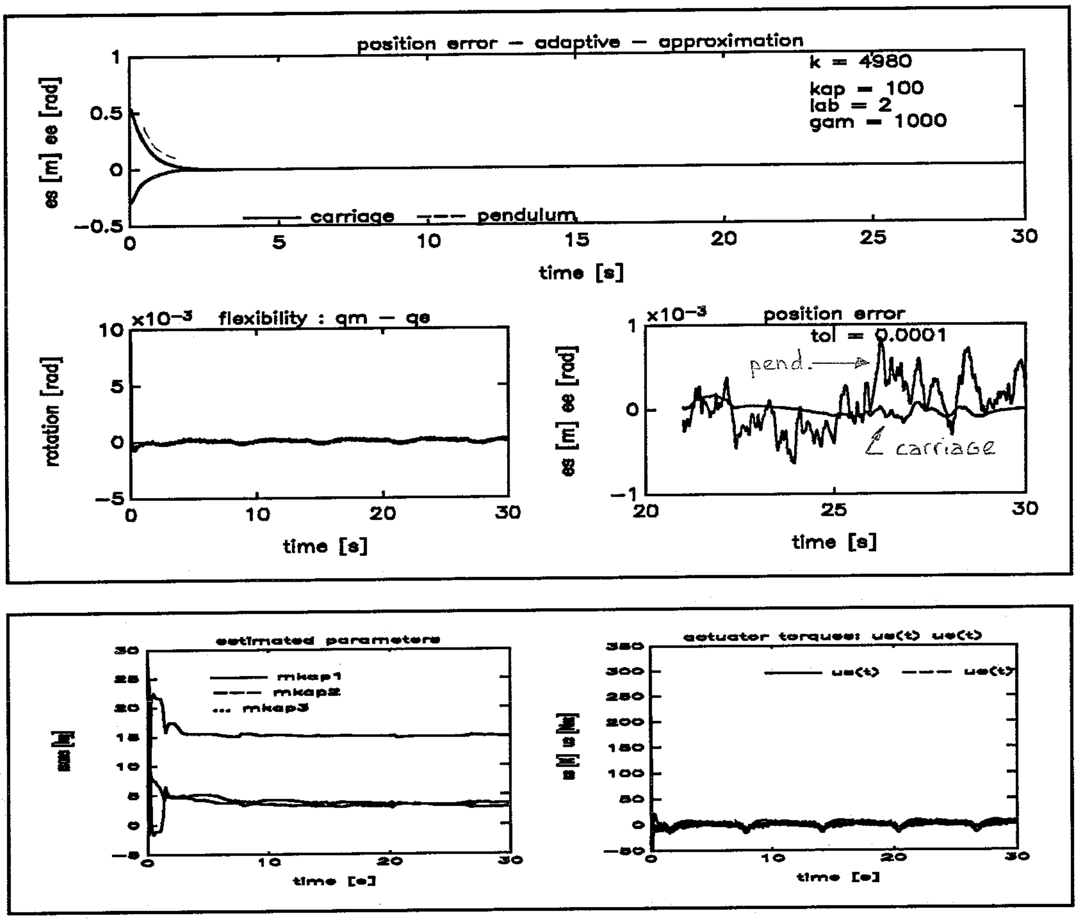


fig. 4.20 / 4.21 adaptive approximated

In practical applications, the question which choice to make between a computationally extensive algorithm with smaller errors, or a computationally less extensive algorithm with larger errors, can only be correctly answered if the demands imposed upon the performance of the overall system are taken into account.

Chapter Five

SUMMARY AND RECOMMANDATIONS

5.1 Summary

Adaptive Control is a powerful tool for controlling systems with uncertain plantparameters. In [2] and [3] Slotine and Li present a 'Basic Algorithm' which effectively deals with the trajectory control of mechanical systems, provided that the number of degrees of freedom equals the number of control input signals (so - called *rigid* robots). In a *flexible* manipulator, i.e. a manipulator with deformable elements, the number of degrees of freedom is greater than the number of control input signals. This means that there are more degrees of freedom to be controlled / stabilized than that there are input signals available. Simulations show that the controller designed according to the Basic Algorithm fails to arrange for a stable system behaviour when applied to a flexible system.

To cope with the trajectory control problem for flexible manipulators, an algorithm is derived which is capable of accomplishing a reasonable trajectory tracking, while hereby limiting the vibrations which occur due to the elasticity in the system. The algorithm is called **Adaptive Computed Torque Computed Reference Control** (abbreviated: ACTCRC).

Simulations show that the non-adaptive controller requires more or less precise, a priori knowledge about the systemparameters. The adaptive controller performs equivalently, but with less information. However, the adaptive control algorithm is computationally more extensive.

The performance of the system strongly depends on an appropriate tuning of the control gains. As a general theory for obtaining optimal control gains is not (yet) available in nonlinear control system design, the simulations are performed with values for the gain matrices **we** think are most suitable.

Actuator saturation hardly influences the performance of the control system. If the mass of the payload is time-varying, the trajectory tracking is still considered as acceptable. In some robot applications it is not the mass of the payload which has to be estimated, but other systemparameters. Simulations are included, in which is demonstrated how the ACTCRC-controller tackles the trajectory control problem in the case of an uncertain spring stiffness. It is shown that to achieve a reasonable trajectory tracking, it is not necessary for an estimated parameter to converge to its true value.

Further, a simplified algorithm is presented. In this algorithm the time derivative of some system variables is calculated by using an approximation technique, instead of deducing them analytically from the Lagrange's equations of motion. The simplified algorithm is computationally less extensive, but leads to a slight performance degradation in the sense of less smaller steady-state position errors.

5.2 Recommandations

In this report we only considered the *numerical* analysis of the control system. In order to check whether the designed controller performs satisfactorily in practice, an *experimental* analysis has to be performed. The XY-table situated in the WFW-lab is therefore suitable. In [5], Brevoord presents the experimental results of the non-adaptive CTCRC-controller. Considering the simulation results presented in this report, we do not doubt that the ACTCRC-controller will do well in governing the XY-table.

In [6], Vijverstra presents the results of various adaptive schemes used for controlling a *rigid* TR-robot. An interesting issue for further research could be to investigate if these schemes are also suitable for adaptive control of a *flexible* TR-robot.

REFERENCES

[1] Lammerts, I.M.M.,
notes on *Control of Robots with Elastic Motor Transmissions*,
Ph.D. thesis to be published

[2] Slotine, J.-J.E., and Li,W., 1987,
"On the Adaptive Control of Robot Manipulators",
The International Journal of Robotics Research, Vol. 6, No. 3

[3] Slotine, J.-J.E., and Li,W., 1991,
Applied Nonlinear Control,
Prentice-Hall International Editions,
Englewood Cliffs, New Jersey

[4] Kammer, R., 1984,
Numerieke Methoden voor Technici,
Delta Press BV,
Oudewater, The Netherlands

[5] Brevoord, G., 1992,
Composite Computed Torque Control of the XY-Table with an Elastic Motor Transmission,
M.Sc. thesis, WFW, Control and Identification of Mechanical Systems,
Department of Mechanical Engineering,
Eindhoven University of Technology,
wfw report: 92.054

[6] Vijverstra, F., 1992,
Direct, Indirect and Composite Adaptive Control of Robot Manipulators,
M.Sc. thesis, WFW, Control and Identification of Mechanical Systems,
Department of Mechanical Engineering,
Eindhoven University of Technology,
wfw report: 92.076

[7] Reijers, L.N., and de Haas, H.J.L.M., 1986,
Flexibele Produktie Automatisering - Industriële Robots,
Technische Uitgeverij De Vey Mestdagh BV,
Middelburg