Adaptive computed torque computed reference control of flexible joint manipulators

Citation for published version (APA):

Document status and date:
Published: 01/01/1993

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

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
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 12. Oct. 2019
Adaptive Computed Torque
Computed Reference Control of
Flexible Joint Manipulators

A.J. Kaats

WFW report 93.103

Professor: Prof. Dr. Ir. J.J. Kok
Coach: Ir. I.M.M. Lammerts

August 1993

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

Summary ................................................................. iii
Notation ................................................................. iv

1 Introduction .......................................................... 1

2 Robot Dynamics ....................................................... 3

3 Control of Flexible Manipulators .................................. 5
   3.1 Non-Adaptive Control of Flexible Manipulators .............. 5
   3.2 Adaptive Control of Flexible Manipulators .................... 8
      3.2.1 Direct Adaptive Control .................................. 8
      3.2.2 Composite Adaptive Control ............................ 11

4 Improving Robustness of (A)CTCR Controller .................... 14

5 (A)CTCRC with Exact Flexibility Compensation .................. 16

6 Variable feedback matrix $K_d$ .................................... 17

7 The XY-table .......................................................... 19
   7.1 Analysis of Model XY-table .................................. 20

8 Design (A)CTCR Controller for XY-table ........................ 22
   8.1 Discrete Implementation ....................................... 23
   8.2 Kalman Observer XY-table ................................... 23

9 CTCR Control of the XY-table ..................................... 24
   9.1 Results of CTCR Control ..................................... 24
   9.2 PID-Feedback .................................................. 27
   9.3 Offset of Tracking Errors ..................................... 28

10 ACTCR Control of the XY-table ................................... 32
   10.1 Results of ACTCR Control .................................... 32
   10.2 Input Vector $u$ ............................................... 36
   10.3 Direction Dependent Friction ................................ 37
   10.4 Improving Robustness ....................................... 39
Summary

Manipulators are controlled to make their end-effector (Tool Centre Point) track a pre-specified trajectory in space. This goal is achieved if the actuators of the manipulator are activated appropriately. The controller, used to calculate these inputs, can have many different forms, each with its own (dis)advantages.

In this report, a controller is discussed for manipulators consisting of a series of links interconnected by joints. One end of the manipulator is fixed to the ground and the other end is the end-effector that has to be controlled. The actuators are positioned at the joints and drive the links by a transmission. It is assumed that some or all of the transmissions are elastic, which is called joint elasticity. Consequently, there are more generalized coordinates than control inputs. Despite the flexibility, the end-effector has to track its desired trajectory.

Besides the elastic transmissions, it is also assumed that the system and load parameters are not or poorly known. In a practical situation, often only estimates of the actual values are available. However, the magnitude of the parameters are important to control the end-effector accurately. Therefore, an adaptive control strategy is used.

Manipulators with joint elasticity and unknown parameters can be controlled by the Adaptive Computed Torque Computed Reference (ACTCR) control law (see Lammerts [13]). This control strategy is discussed in this report and results of simulations and experiments with a xy-table are presented. The simulations have been executed without the presence of unmodeled dynamics; the model of the manipulator is supposed to be exactly known. In the experimental case there will always be model inaccuracies, which decrease the performance or may even result in an unstable closed-loop behaviour. Some modifications and extensions of the ACTCR controller are also given and applied to the xy-table.

From the obtained results it appeared that the ACTCR controller can control the xy-table. Trajectory tracking is achieved while the elastic transmission torques/forces are bounded. If the system/load parameters are poorly known, the "learning" capability of the controller improves the performance.
Notation

$A, a$: scalar

$a$: column (small italic characters)

$a_i$: element on row $i$ of column $a$

$\hat{a}$: estimate of column $a$

$a_d$: column with desired values of column $a$

$\bar{a}$: estimation error $\hat{a} - a$

$\dot{a}$: first order derivative of column $a$

$\ddot{a}$: second order derivative of column $a$

$A$: matrix (capital italic characters)

$A_{ij}$: element on row $i$, column $j$ of matrix $A$

$A^{-1}$: inverse of matrix $A$

$\|A\|$: norm of matrix $A$

$a^T, A^T$: transpose of column or matrix
Chapter 1: Introduction

Manipulators (or robots) are often used in the industry because they increase the productivity, the quality of the products and can do unpleasant work.

In the last decade much research has been done to the control of manipulators. The robot dynamics are often highly nonlinear. The influence of the nonlinear dynamics are considerable if the operational speeds are high. For slow motions, the manipulator dynamics can be approximated well by a linear model. Then, the controller design can be based on the linear model. However, if the manipulator moves at high speeds, the nonlinearities cannot be linearized without a significant error. Using the controller based on a linear model leads to degradation of the performance or may even cause instability. Hence, for the increased demands on manipulators, advanced non-linear controllers are needed.

Nonlinear control is an important element for high speed robots. Another aspect that frequently may not be neglected, is the flexibility of the manipulators. To obtain fast responses and low energy consumptions, often a lightweight construction is used. Combined with high operational accelerations, this leads to systems that cannot be considered to be rigid. A control law designed for a rigid manipulator, applied to the flexible system may result in considerable tracking errors or even an unstable closed loop dynamics. So, a special strategy has to be used which can handle the flexibilities.

Further, the system and load parameters, like mass and friction, are usually poorly known or totally unknown. Often, these parameters are constant or vary slowly in time. An adaptive control law has to be used to cope with the parametric uncertainties.

In this report, the three aspects described above will be combined for a special class of flexible manipulators.

It is assumed that a manipulator can be modeled as an open chain of links interconnected by joints, with one degree of freedom per joint. One end of the chain is connected to the ground and the other end (end-effector) has to follow a specific trajectory. Desired trajectories of the links are assumed to be computable from the desired trajectory of the end-effector.

The actuators are positioned at the joints and drive the links by a transmission. Some of these transmissions are supposed to be flexible. Therefore, the positions of the actuators no longer agree with the position of the end-effector. This so-called joint flexibility is assumed to introduce an extra degree of freedom, describing the rotor of the motor, for every elastic transmission.

Schematically, the manipulators can be pictured as follows:
The control objective of this class of flexible manipulators is described as: given a desired trajectory of the end-effector and some or all the parameters being unknown, derive a control law for the actuator torques/forces and an estimation law for the unknown parameters, such that the end-effector tracks the desired trajectory after an initial adaptation process, taking into account flexible transmissions between motors and links.

The Adaptive Computed Torque Computed Reference controller (Lammerts [13]) is an adaptive control law which can handle the flexible transmissions. In this report, the ACTCR control strategy will be discussed and applied to the xy-table.

In chapter 2, the model of the (flexible-joint) manipulators and its properties will be presented. This is the starting point of the controller design.

In chapter 3, the control scheme of flexible-joint manipulators will be explained and discussed. After the non-adaptive control law, the adaptive version (ACTCR) will be given.

In chapter 4, 5 and 6, some possible modifications/extensions of the ACTCR control strategy will be presented. Also, their theoretical backgrounds will be explained.

In chapter 7, the experimental set-up will be discussed. Chapter 8 includes the design of the control law for this system.

In chapter 9 and 10, the results of the non-adaptive and adaptive controller will be given. Both simulation and experimental results are used to investigate the properties of the ACTCR control scheme.

In chapter 11, the main conclusions and recommendations for further research will be given.
Chapter 2: Robot Dynamics

A control system is required for a manipulator to fulfil its task appropriately. Many control methods need some knowledge about the behaviour of the manipulator. This behaviour (or dynamics) can be described by differential equations and is called a model.

With good knowledge of the system, such a model-based controller can be designed, so the closed-loop dynamics answers to the requirements closely.

Hence, it is very important that a good model of the manipulator is available. The method of Lagrange can be used to derive a model. A set of second-order differential equations is obtained, which is usually coupled and nonlinear.

In this report, it is assumed that the differential equations of a stiff manipulator can be written as follows:

\[
M(q)\ddot{q} + C(q,\dot{q})\dot{q} + w(q,\dot{q}) + g(q) + n(q,\dot{q},t) = Hu
\]  

(2.1)

with:
- \(M(q)\): symmetric positive definite inertia matrix
- \(C(q,\dot{q})\): vector containing Coriolis and centrifugal torques/forces
- \(w(q,\dot{q})\): friction vector
- \(g(q)\): vector of gravitational torques/forces
- \(n(q,\dot{q},t)\): vector of dynamics not accounted for above
- \(H\): distribution matrix
- \(u\): input vector
- \(q \in \mathbb{R}^n\): vector of generalized coordinates that fixes the position of the system
- \(u \in \mathbb{R}^m\)
- \(H \in \mathbb{R}^{nxm}\)

For rigid manipulators, the number of inputs is equal to the number of generalized coordinates, so \(n=m\). In case the manipulator cannot be considered to be stiff, it is modeled as:

\[
M(q)\ddot{q} + C(q,\dot{q})\dot{q} + Kq + w(q,\dot{q}) + g(q) + n(q,\dot{q},t) = Hu
\]

(2.2)

where:
- \(K\): symmetric stiffness matrix

If the elastic transmissions do not behave linear, the nonlinearities can be put in the vector \(n(q,\dot{q},t)\).

Because of the flexibilities, there are less inputs than generalized coordinates, so \(n>m\). This is the main problem in the control of flexible manipulators and will be faced in Chapter 3.
It is assumed that each flexibility in the system causes one extra degree of freedom. In fact, this excludes link flexibilities described by partial differential equations, because they have to be modeled by an infinite number of degrees of freedom. In a first attempt, these flexible links can be approximated by elastic joints. However, a more accurate description can be obtained if the distributed link flexibilities are modeled by using the method of Rayleigh-Ritz [4]. This yields an inertia and stiffness matrix describing the lowest, dominant, eigenfrequencies. Then, the link flexibilities are modeled by a finite number of degrees of freedom.

A few remarks can be made concerning the differential equations (2.1) and (2.2):

- The Coriolis and centrifugal forces are nonlinear in positions and velocities. Because multiplications of different velocities occur, the matrix \( C(q,\dot{q}) \) is not unique. However, if \( C \) is chosen as (see Slotine & Li [24]):

\[
c_{ij} = \frac{1}{2} \sum_{k} \frac{\partial M_{ik}}{\partial q_j} \dot{q}_k + \frac{1}{2} \sum_{k} \left( \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{ij}}{\partial q_k} \right) q_k
\]

the next relation between the inertia matrix and the Coriolis/centrifugal matrix \( C \) holds:

\[
x^T(\dot{M} - 2C)x = 0
\]

for every \( x \in \mathbb{R}^6 \).

This skew-symmetry property of \((\dot{M}-2C)\) will be used in the stability analysis of the closed-loop dynamics.

- The differential equations (2.1) and (2.2) are linear parametrizable, i.e. the left-hand side is linear dependent on a suitable selected set of system and load parameters. This is a necessary condition for the controllers of this report.

The linear parametrizability is valid for the inertia matrix, the Coriolis/centrifugal matrix, the stiffness matrix and the gravitation vector. It is assumed that it is also valid for the vectors \( w \) and \( n \).

Thus, for a rigid manipulator:

\[
M(q,a)\ddot{q} + C(q,\dot{q},a)\dot{q} + w(q,\dot{q},a) + g(q,a) + n(q,\dot{q},t,a) = Y(q,\dot{q},\ddot{q},t) a
\]

with:

- \( a \) : vector of system and load parameters
- \( Y(q,\dot{q},\ddot{q},t) \) : regression matrix

A slightly different equation also holds:

\[
M(q,a)x + C(q,\dot{q},a)y + w(q,\dot{q},a) + g(q,a) + n(q,\dot{q},t,a) = Y(q,\dot{q},x,y,t) a
\]

The properties (2.4) and (2.5) are valid for the flexible manipulator model (2.2) too.
Chapter 3: Control of Flexible Manipulators

Slotine & Li [23] propose a control law for rigid manipulators which consists of a PD feedback part and a "computed torque"-like full dynamics compensator. It can be used with or without exact knowledge of the system parameters. In the latter, an on-line parameter estimator has to be used. The suggested control law is the basis of this report. The other control laws that will be discussed, are extensions or modifications of this control law.

In this chapter, the non-adaptive control law of Slotine & Li will be presented. It is extended to make it applicable for flexible manipulators. This is the first element in the discussion of the Adaptive Computed Torque Computed Reference Controller (Lammerts [13]), which will be completed by its adaptive version.

3.1 : Non-Adaptive Control of Flexible Manipulators

In case the system parameters are known, the control law for rigid manipulators, as proposed by Slotine & Li, is:

\[ Hu = M(q) \ddot{q}_r + C(q, \dot{q}) \dot{\dot{q}}_r + w(q, \dot{q}) + g(q) + n(q, \dot{q}, t) - K_s \dot{e}, \]  \hspace{1cm} (3.1)

where:

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

- \( \ddot{q}_d \) = desired acceleration
- \( \dot{q}_d \) = desired velocity
- \( q_d \) = desired trajectory

\[ u \in \mathbb{R}^n \]
\[ q \in \mathbb{R}^n \]

Here, \( \dot{e}_r \) is a combined error of the position error and the velocity error and \( \dot{e}_r = 0 \) is a sliding surface as will be seen later.

Furthermore, \( \Lambda \) and \( K_s \) are positive definite, usually diagonal, feedback matrices.

The control law (3.1) is based on a rigid manipulator model. It cannot be used directly to control flexible robots, because it can lead to tracking inaccuracy or instability, especially if the flexibilities play an important role. Therefore, an extension of the control law is
necessary, such that the end-effector reaches its desired trajectory and the elastic deformations are bounded.

The control law is slightly changed to make it applicable for flexible robots (see Lammerts [13]):

\[ Hu = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + Kq + w(q,\dot{q}) + g(q) + n(q,\dot{q},t) - K_d \dot{e}_r \]  

(3.2)

With the system dynamics of the flexible manipulator:

\[ M(q)\ddot{q} + C(q,\dot{q})\dot{q} + Kq + w(q,\dot{q}) + g(q) + n(q,\dot{q},t) = Hu \]  

(3.3)

the control law (3.2) results in the closed loop dynamics:

\[ M(q)\ddot{e}_r + C(q,\dot{q})\dot{e}_r + K_d \dot{e}_r + Ke_r = 0 \]  

(3.4)

Stability can be shown by using the Lyapunov function:

\[ V = \frac{1}{2} \dot{e}_r^T M \dot{e}_r + \frac{1}{2} e_r^T Ke_r \]  

(3.5)

(the argument has been dropped for economy)

The derivative of this function is:

\[ \dot{V} = -\dot{e}_r^T C \dot{e}_r + \frac{1}{2} \dot{e}_r^T M \dot{e}_r - \dot{e}_r^T K_d \dot{e}_r = -\dot{e}_r^T K_d \dot{e}_r \]  

(3.6)

where the skew-symmetry property of \((M-2C)\) and the closed-loop dynamics have been used.

To prove that \(\dot{e}_r \rightarrow 0\), Barbalat's lemma [24] can be used, which states that \(\dot{V}\) tends to zero if \(\dot{V}\) is bounded. Because:

\[ \dot{V} = -2\dot{e}_r^T K_d \dot{e}_r \]  

(3.7)

\(\dot{V}\) is bounded if \(\dot{e}_r\) and \(\ddot{e}_r\) are bounded. Since \(V > 0\) and \(\dot{V} \leq 0\), \(V\) remains bounded, and thus \(e_r\) and \(\dot{e}_r\) also. Consequently, \(q\) and \(\dot{q}\) are bounded, since \(q_d\) and \(\dot{q}_d\) are. Using (3.4), \(\dot{e}_r\) will be bounded, and therefore \(\dot{V}\) too. Thus, control law (3.2) makes the tracking errors converge to the sliding surface \(\dot{e}_r = \dot{e} + \Lambda e = 0\), which implies that both \(e\) and \(\dot{e}\) tend to 0 as \(t \rightarrow \infty\).

The feedback matrix \(K_d\) determines the speed of convergence of \(\dot{e}_r\) to zero. After \(\dot{e}_r\) has become zero, the matrix \(\Lambda\) prescribes how fast \(e\) and \(\dot{e}\) tend to zero.

Boundedness of \(q\) has been guaranteed by using Barbalat’s lemma. Consequently, the torques/forces \(z = Kq\) due to the elastic deformations are bounded. So, control law (3.2)
insures zero tracking errors and bounded deformations.

It has been proven that \( \dot{e}_r \to 0 \) as \( t \to \infty \). The closed-loop dynamics can then be written as:

\[
M \ddot{e}_r + Ke_r = 0
\]

The inertia matrix \( M \) is positive definite and thus invertible, so the vector \( \ddot{e}_r \) is only zero if \( Ke_r \) is zero. Therefore, the system is stabilized at \( Ke_r = 0 \). Besides, the convergence of \( \dot{e}_r \) to zero implies that \( e_r \) converges to a constant value. This means that there can remain a difference between the reference trajectory and the actual trajectory for all generalized coordinates.

In order to compute the input vector \( u \), the reference trajectories of all generalized coordinates have to be available. However, the desired trajectories of some degrees of freedom are unknown, because the desired trajectory of the end-effector does not prescribe desired trajectories for all generalised coordinates. It is assumed that the desired trajectories of only \( m \) generalized coordinates are computable from the desired trajectory of the end-effector. Thus the number of inputs is equal to the number of desired trajectories. This is true for robots with (some) elastic transmissions between the motors and the links as considered in this report, where the desired trajectories of some or all motor coordinates are unknown. Therefore, their reference trajectories can not be determined by:

\[
q_r = q_d - \Lambda \int^t_0 (q - q_d) d\tau
\]

Example:

Consider the next system with two degrees of freedom, where \( x_1 \) is the motor-coordinate and \( x_2 \) is the link-coordinate, and with one input torque/force, \( u \).

\[
\text{Figure 3.1: Example}
\]

Mass \( m_2 \) (= end-effector) has to follow the desired trajectory \( x_{2d} \), so \( x_{2d} \) can be determined using equation (3.9). Besides \( x_{2d} \), the reference trajectory \( x_{1r} \) has to be available to calculate the input \( u \), but the desired trajectory of \( x_1 \) is unknown.

Lammerts [13] solved this problem by splitting the control law (3.2):

\[
H^T H u = H^T [ M(q) \ddot{q}_r + C(q,\dot{q}) \dot{q}_r + K q_r + w(q,\dot{q}) + g(q) + n(q,\dot{q},t) - K_d \dot{e}_r ]
\]

\[
N^T H u = N^T [ M(q) \ddot{q}_r + C(q,\dot{q}) \dot{q}_r + K q_r + w(q,\dot{q}) + g(q) + n(q,\dot{q},t) - K_d \dot{e}_r ]
\]
with:
\[ H \in \mathbb{R}^{(n \times m)} \]
\[ N \in \mathbb{R}^{(n \times n-m)} \]  

and
\[ N^T H = 0 \]

Consequently:
\[ u = (H^T H)^{-1} H^T \left[ M(q) \ddot{q}_r + C(q, \dot{q}) \dot{q}_r + K q_r + w(q, \dot{q}) + g(q) + n(q, \dot{q}, t) - K_a \dot{e}_r \right] \]  
\[ 0 = N^T \left[ M(q) \ddot{q}_r + C(q, \dot{q}) \dot{q}_r + K q_r + w(q, \dot{q}) + g(q) + n(q, \dot{q}, t) - K_a \dot{e}_r \right] \]

Equation (3.15) includes \( n-m \) second-order differential equations in \( m \) known reference trajectories and \( n-m \) unknown reference trajectories. The \( n-m \) unknown reference trajectories can now be determined from these equations. For a further analysis of this so called "Computed Torque Computed Reference" (CTCR)-strategy see Lammerts [13]. From now on it is assumed that it is possible to compute the unknown reference trajectories from equation (3.15). These computed reference trajectories can then be used in equation (3.2) to determine the input vector \( u \).

In chapter 5, it will be seen that for a class of flexible joint manipulators, it is possible to use the term \( Kq \) in the control law, instead of \( Kq_r \). However, control law (3.2) is applicable for any flexible joint manipulator.

3.2 : Adaptive Control of Flexible Manipulators

In practice, the system and load parameters are unknown and only estimates of the real values are available. Therefore, control law (3.2) cannot be used directly. It has to be changed in order to cope with the parametric uncertainties.

Many adaptive control laws in literature have some unattractive aspects, like the one proposed by Craig et al. [7]. Their adaptation algorithm requires measurement or estimation of the acceleration and a time consuming inversion of the estimated inertia matrix. The strategy that will be discussed here, uses estimates of the system parameters in the control law and has an on-line estimator which updates these estimates. Two adaptation methods, the direct adaptation algorithm and the composite adaptation algorithm, will be introduced successively.

3.2.1 : Direct Adaptive Control of Flexible Manipulators

The adaptive control law, which will be described in this paragraph, is based on lowering the tracking errors. Therefore, it is called a direct adaptive controller. First, the control law and adaptation algorithm will be given, followed by the stability analysis, which prescribes
the chosen parameter adaptation algorithm.

Since only estimates of the system and load parameters are available, an adaptive version of control law (3.2) is required. Instead of the real parameters now their estimates are used:

$$Hu = \ddot{M}(q)\ddot{q}_r + \dddot{C}(q,\dot{q})\dot{q}_r + \dddot{K}q_r + \dot{w}(q,\dot{q}) + g(q) + \dot{n}(q,\dot{q},t) - K_a\dot{e}_r$$  \hspace{1cm} (3.16)

The adaptation algorithm for the estimated parameters is:

$$\dot{\hat{a}} = -\Gamma Y^T_r \dot{e}_r$$  \hspace{1cm} (3.17)

with:

$$Y_r\hat{a} = Y_r(q,\dot{q},\ddot{q},\dddot{q},t)\hat{a} = \ddot{M}(q)\ddot{q}_r + \dddot{C}(q,\dot{q})\dot{q}_r + \dddot{K}q_r + \dot{w}(q,\dot{q}) + g(q) + \dot{n}(q,\dot{q},t)$$

$$\Gamma$$ : positive definite adaptation matrix, usually diagonal.

$$\hat{a}$$ : estimate of parameter vector $$a$$.

Again, for a special class of flexible joint manipulators, the term $$\dot{K}q_r$$ can be replaced by $$\dddot{K}q_r$$, as will be seen in chapter 5.

The adaptation matrix $$\Gamma$$ in equation (3.17) determines the speed of the adaptation process. If $$\Gamma$$ is chosen too large, the parameter estimation is too oscillatory. If, instead, $$\Gamma$$ is too small, the adaptation algorithm is not activated enough to let the estimates converge to constant values within an acceptable time. Unfortunately, until now, a good guideline to choose this matrix is not available.

To prove the stability of the controlled system, the closed loop dynamics are written as:

$$M(q)\ddot{e}_r + C(q,\dot{q})\dot{e}_r + K_a\dot{e}_r + Ke_r = Y_r(q,\dot{q},\ddot{q},\dddot{q},t)\ddot{\hat{a}}$$  \hspace{1cm} (3.18)

where:

$$\ddot{\hat{a}} = \hat{\dot{a}} - a$$

Use the Lyapunov function:

$$V = \frac{1}{2}e^T J M e_r + \frac{1}{2}e^T J K e_r + \frac{1}{2}e^T \hat{a} \Gamma^{-1} \hat{a}$$  \hspace{1cm} (3.19)

Differentiating yields:

$$\dot{V} = -e^T J K \dot{e}_r + e^T J \dot{a} + \dot{e}^T \hat{a} \Gamma^{-1} \dot{\hat{a}}$$  \hspace{1cm} (3.20)

Using the adaptation algorithm (3.17) results in:

$$\dot{V} = -e^T J K \dot{e}_r$$  \hspace{1cm} (3.21)
Barbalat's lemma can, again, be used to demonstrate the convergence of $e$ and $\dot{e}$ to zero. Thus control law (3.16) with adaptation algorithm (3.17) insures zero steady state errors.

Because $\dot{e}$, tends to zero, the closed loop dynamics for large $t$ can be written as follows:

$$M \ddot{e} + Ke = Y \dot{a}$$  \hspace{1cm} (3.22)

The controlled system stabilizes at $Ke=Y \dot{a}$, since $M$ is invertible.

This adaptive control law for flexible manipulators is called "Adaptive Computed Torque Computed Reference Control" (ACTCRC).

The adaptation algorithm does not insure that the parameter estimates convergence to their real values. It only guarantees constant estimates after an initial adaptation process. It can be derived that the estimated parameters converge to their real values if the regression matrix $Y(q, q, q, \dot{q}, \dot{q}, \dot{q}, t)$ is "persistently exciting" (see, for example, Sadegh & Horowitz [18]). A matrix $X$ is persistently exciting if there are positive constants $\alpha_1$, $\alpha_2$ and $\delta$, such that:

$$\forall t \geq 0 \quad \alpha_1 \leq \int_{t-\delta}^{t} X^T(\tau)X(\tau) d\tau \leq \alpha_2$$  \hspace{1cm} (3.23)

This condition prescribes that matrix $X$ must vary sufficiently over the interval $\delta$, so the entire space is spanned. Applied to the matrix $Y(q, q, q, \dot{q}, \dot{q}, \dot{q}, t)$, this means that the desired trajectory must vary enough. Often, it is said that the desired trajectory should be "sufficient rich".

If the ACTCR control law is analyzed, it can be seen that two coupled sets of equations have to be solved simultaneously:

- In the adaptive version, the system parameters of equation (3.15) are replaced by their estimates. So, to determine the unknown reference trajectories, the estimated parameters must be used.

- The reference trajectories of all generalized coordinates should be available to obtain the estimate of the parameter vector by:

$$\dot{a}(t) = \dot{a}(t_0) - \int_{t_0}^{t} Y_r^T \dot{e}_r d\tau$$  \hspace{1cm} (3.24)

with : $Y_r = Y_r(q, q, q, \dot{q}, \dot{q}, \dot{q}, t)$

It should be noticed that computers are often used for control purposes. These computers operate in discrete time, which means that, for example, a continue time-dependent variable $x$ is represented in the computer by a number of discrete points $x_i$. The problem stated above is solved by predicting the parameter vector $\dot{a}_i$ at time $t_i$ by looking at the previous behaviour of $\dot{a}_i$. For example, the Euler forward-integration scheme
is suitable to predict the parameter vector at $t_i$. Now, the unknown reference trajectories at $t_i$ can be determined. Of course more exact results will be obtained if the two coupled equations are combined to eliminate the coupling, but this will not always be possible.

### 3.2.2: Composite Adaptive Control of Flexible Manipulators

The direct adaptive control law, described in the previous paragraph, only uses the tracking errors to adapt the estimated parameters. Besides this, it is also possible to adjust the estimated parameters by so-called prediction errors. Then, the adaptation is not primarily focused on decreasing the tracking errors, but on extracting information about the true parameter values. Adaptive control laws with adaptation algorithms which only use these prediction errors, are called indirect controllers. If both errors are used, a composite controller is obtained. In this paragraph, a composite adaptive controller for flexible joint manipulators will be presented.

Because of the linear parametrizability property, the robot model (2.2) can be written as:

$$Hu = Y(q,\dot{q},\ddot{q})a$$  \hspace{1cm} (3.25)

This robot model is the only model that has information about the true parameters and, therefore, it will be used to estimate the parameters.

In the first part of this paragraph, the prediction error will be discussed briefly, followed by the composite adaptation algorithm.

The acceleration in equation (3.23) is undesirable for control purposes, because it is usually not measured and computation by numerical differentiation yields values contaminated with noise. By filtering equation (3.23) through an exponentially stable and strictly proper filter, the acceleration is no longer required. After filtering, the next equation is obtained:

$$y = W(q,\dot{q})a$$  \hspace{1cm} (3.26)

where:

$y$ : vector of filtered torques/forces.
$W(q,\dot{q})$ : filtered version of matrix $Y(q,\dot{q},\ddot{q})$.

A prediction of the filtered torques/forces $y$ can now be generated by using the estimated parameters:

$$\hat{y} = W(q,\dot{q})\hat{a}$$  \hspace{1cm} (3.27)

where:

$\hat{y}$ : vector of predicted filtered torques/forces.

The difference between the predicted filtered torques/forces and the actual filtered torques/forces is called the prediction error $e$.

$$e = \hat{y} - y = W(q,\dot{q})\hat{a}$$  \hspace{1cm} (3.28)
The prediction error will be used in the composite adaptation algorithm for the unknown parameters.

Both the direct and the composite adaptive control use the same control law (3.16). However, the direct adaptive scheme only uses the tracking errors to adjust the parameter estimates while the composite method needs both the tracking errors and the prediction errors. The reason for this is that both errors contain information about the true parameters.

The composite adaptation algorithm is defined as follows (e.g. Slotine & Li [22]):

\[ \dot{a} = -P(t) [Y_t^T \dot{e} + W_t^T R(t) e] \] (3.29)

with:

- \( P(t) \): constant or time varying positive definite gain matrix.
- \( R(t) \): constant or time varying weighting matrix, indicating how much attention should be paid to the parameter information of the prediction error.

The matrix \( P(t) \) can be formed in many different ways. In this report, only the next choice is used:

\[ P(t) = P_0 = \text{constant} \] (3.30)

For this and some other strategies to choose \( P(t) \) and their properties, see Vijverstra [27] and Li & Slotine [15].

The stability proof for the control law (3.16) with the composite adaptation algorithm (3.27), is given in appendix A.

In Li & Slotine [15], an indirect adaptation control law is proposed. The indirect adaptation algorithm only uses the prediction error to adapt the parameters. However, the indirect adaptive control law is very different from the direct and composite adaptive control law. Therefore, a comparison of the adaptation performance of the indirect controller is not possible.

Vijverstra [27] claims that the indirect adaptation law can be used with the same control law as the direct and composite adaptive controllers, but the stability proof he gives is not correct.

Yet, it is possible to construct a kind of indirect adaptation algorithm combined with control law (3.16). This "indirect" controller is obtained from the composite adaptation algorithm (3.27). Take \( R(t) = rI \) with \( r \) large positive scalar and \( I \) the unity matrix. The prediction error is thus weighted by \( r \). If \( P(t) = 1/r^*P(t)_{\text{indirect}} \), the next adaptation algorithm is obtained:

\[ \dot{a} = -P(t)_{\text{indirect}} W_t^T e \] (3.31)
This is, of course, only valid if the term $rW^te$ is large compared to $Y^Te$, which does not have to be true throughout the entire control process. However, if the "prediction error"-part of (3.27) tends to zero while the "tracking error"-part is not zero, the gain $r$ can be increased, so the former remains dominant. If $e$ is zero, the "indirect" adaptation algorithm turns to a direct adaptation algorithm, but if $r$ is large, the adaptation matrix $P$ is small and thus there will be no adaptation on basis of the tracking error.

In this chapter a non-adaptive and an adaptive control law for flexible joint manipulators have been presented. Two adaptation algorithms, required by the adaptive controller have been introduced, namely direct and composite adaptation. The composite adaptation can include many different estimators. Only one of them is given, namely the gradient estimator. In the next two chapters, two aspects of the control laws will be introduced.
Chapter 4: Improving Robustness of (A)CTCR Controller

For many controller designs, it is necessary to have an appropriate model of the system. A model is suitable if it describes the most important aspects of the system. In practice, there will always be facets not accounted for:

- **parameters not included in the adaptation algorithm**
  Sometimes, it is better not to estimate a certain parameter, but to make the controller robust against it. This can be done, for example, if the parameter is of minor importance to the system dynamics. The extra computational effort needed for estimation is eliminated then.

- **unmodeled dynamics**
  In practice, there will always be unmodeled dynamics and (measurement) noise. The control law has to be robust against these phenomena, otherwise the closed-loop dynamics may become unstable.

The (adaptive) control laws for rigid and flexible manipulators can be extended to make them more robust against model imprecisions. The rest of this chapter is focused on the robust, adaptive control of flexible manipulators, but it can be easily applied to other control laws.

Let the model of a flexible manipulator be given by:

\[
M(q)\ddot{q} + C(q,\dot{q})\dot{q} + Kq + \dot{w}(q,\dot{q}) + g(q) + n(q,\dot{q},t) + d = Hu
\]  

(4.1)

where \(d\) is the influence of the unmodeled dynamics and the unestimated parameters. The maximum value of \(d\) is supposed to be known; this is necessary to guarantee stability.

Now, the next robust, adaptive control law is chosen:

\[
Hu = \dot{M}(q)\dot{q}_r + \dot{C}(q,\dot{q})\dot{q}_r + \dot{K}q_r + \dot{\psi}(q) + \dot{g}(q) + \dot{n}(q,\dot{q},t) - K_d\dot{\epsilon}_r - K_s\text{sign}(\dot{\epsilon}_r)
\]  

(4.2)

with \(K_s\): positive definite diagonal matrix.

Despite of the unmodeled dynamics and unestimated parameters, stability can be proven with Lyapunov function:

\[
V = \frac{1}{2}\dot{\epsilon}_r^T\dot{\epsilon}_r + \frac{1}{2}\dot{\epsilon}_r^TKe_r + \frac{1}{2}\dot{a}^T\Gamma^{-1}\dot{a}
\]  

(4.3)

So,

\[
\dot{V} = -\dot{\epsilon}_r^TK_s\dot{\epsilon}_r + \dot{\epsilon}_r^T(-d-K_s\text{sign}(\dot{\epsilon}_r))
\]  

(4.4)
If $K_{ii} \geq \max |d_i| + \eta_i$ for $i=1..n$, where $\max |d_i|$ is the maximum value of $|d_i|$ and $\eta_i$ is a positive constant, the expression for the derivative of the Lyapunov function becomes:

$$\dot{V} \leq -\dot{e}_r^T K_d \dot{e}_r - \sum_{i=1}^{n} \eta_i |\dot{e}_{ir}|$$  (4.5)

Because of the second right-hand term, the feedback matrix $K_d$ is no longer necessary, but it can be used to accelerate the convergence.

To avoid chattering, $\text{sat}(\dot{e}_r, \Phi)$ should be used instead of $\text{sign}(\dot{e}_r)$ (e.g. Asada & Slotine [2]):

$$\text{sat}(\dot{e}_r, \Phi) = \begin{cases} 
-1 & \text{if } \dot{e}_r \leq -\Phi \\
\frac{\dot{e}_r}{\Phi} & \text{if } -\Phi \leq \dot{e}_r \leq \Phi \\
1 & \text{if } \dot{e}_r \geq \Phi 
\end{cases}$$  (4.6)

$\Phi$ is called the boundary layer.

No information for parameter estimation can be extracted from the errors $\dot{e}_r$, if they are in the boundary layer. The vector $d$ can drive $\dot{e}_r$ anywhere in the boundary layer, without containing information about the estimated parameters; the adaptation should be stopped then.
Chapter 5: (A)CTCRC with Exact Flexibility Compensation

The ACTCR control law, as discussed in chapter 3, can be used to control flexible joint manipulators, for which the number of inputs equals the number of desired trajectories. The coupling between the differential equations describing the dynamics of the motor coordinate, and the differential equations describing the link coordinate, is always realized by the stiffness matrix $K$ (and eventual an extra dampings matrix if damping is present in the flexible joint). However, for some flexible joint manipulators, it is also established by the inertia matrix $M$ and Coriolis/centrifugal matrix $C$. In this chapter, a modified version of the ACTCR control law will be given for this class of manipulators. The non-adaptive control law is accordingly.

If, besides $K$, the matrices $M$ and $C$ provide a coupling between the differential equations describing the motor coordinate and those describing the link coordinate, the next altered control law can be used:

$$ Hu = \dot{M}(q)\ddot{q}_r + \dot{C}(\dot{q},\dot{q})\dot{q}_r + \ddot{K}q + \ddot{w}(q,\dot{q}) + \ddot{g}(\dot{q}) + \ddot{a}(q,\dot{q},t) - K_d\ddot{e}_r \quad (5.1) $$

Now, the closed-loop dynamics does no longer contain the "stiffness" term $K\ddot{e}_r$:

$$ M\ddot{e}_r + C\dot{e}_r + K_d\dot{e}_r = Y_\ddot{a} \quad (5.2) $$

with:

$$ Y_\ddot{a} = Y_\ddot{a}(q,\dot{q},\ddot{q},\dot{q},t) $$

This closed-loop dynamics can be compared to a mass-damper system. If such a system is excited by a puls-shape force, it will reach its steady-state faster than when the system is extended with a spring (i.e. with $K\ddot{e}_r$). Consequently, it is expected that the ACTCR controller with $\dot{K}q$ yields better results than the ACTCR controller with $\dot{K}q_r$.

Now, assume that the stiffness matrix $K$ is the only coupling between the differential equations describing the dynamics of the motor coordinate, and the differential equations describing the link coordinate. In figure 3.1, a manipulator consisting of one elastic transmission is pictured schematically. The desired trajectories of the link coordinates ($x_l$ in figure 3.1) are assumed to be computable from the desired trajectory of the end-effector. Therefore, the reference trajectories of the motor coordinates ($x_l$ in figure 3.1) have to be calculated. Since equation (3.15) is used to determine these unknown reference trajectories, the only coupling between the known and unknown reference trajectories is established by $\dot{K}q$. Replacing $\dot{K}q$ by $\dot{K}q$ eliminates this coupling. The unknown reference trajectories of the motor coordinates are no longer present in equation (3.15) and thus can not be computed. Even if the feedback matrix $K_d$ is not chosen diagonal, the coupling is not restored.
Chapter 6: Variable Feedback Matrix $K_d$

Consider the next system:

![Figure 6.1: Example](image)

A controllable torque/force $u$ is acting on a mass-spring system. If mass $m$ has to follow a specific trajectory $x_d$, it has to be controlled by, for example, a PD-controller: $u = -K_d\dot{e} - K_pe$ with $e = x - x_d$. Suppose an initial difference between the actual state and the desired state. The PD-controller will generate an input $u$ so the initial errors disappear. Now, assume the same magnitude of the initial errors but with a bigger mass $m$. The controller has to supply a larger input $u$ to get the same performance as with lower mass. So, the gains of the PD-controller have to be increased with bigger mass.

The reasoning above applied to the CTCR controller leads to the choice $K_d = \alpha M$, with $\alpha$ a positive constant. The inertia matrix $M$ is positive definite and, therefore, $K_d$ is positive definite too (which is the stability condition as seen in chapter 3). Thus, the CTCR controller with $K_d = \alpha M$ yields stable closed-loop dynamics.

If the system parameters are unknown a priori, the ACTCR has to be used, which produces an estimated inertia. But with $K_d = \alpha \dot{M}$, the stability can no longer be guaranteed because the Lyapunov function (3.19) then leads to:

$$\dot{V} = -\alpha \dot{e}^T \dot{M} \dot{e}$$

(6.1)

if adaptation algorithm (3.17) is used.

The estimated inertia matrix is not necessarily positive definite and thus stability cannot be ensured by Lyapunov function (3.19).

To guarantee stability for the choice $K_d = \alpha \dot{M}$, the regression matrix $Y_r(q, \dot{q}, q_r, \dot{q}_r, t)$ is modified into:

$$\dot{M} \ddot{q}_r - \alpha M \dot{e}_r + \dot{C} \dot{q}_r + \ddot{K} q_r + \ddot{\ddot{w}} + g + \ddot{n} = Y_m \ddot{a}$$

(6.2)

(arguments have been dropped)
With adaptation algorithm:

\[ \dot{\alpha} = -\Gamma \gamma_m \dot{\hat{e}}_r \]  \hspace{1cm} (6.3)

the Lyapunov function (3.19) yields:

\[ \dot{V} = -\alpha \dot{e}_r^T M \dot{e}_r \]  \hspace{1cm} (6.4)

Because the inertia matrix \( M \) is positive definite and \( \alpha \) is positive, the new adaptation algorithm leads to a stable closed-loop behaviour.

This control law can be changed to make it suitable to control rigid robots with known parameters. Then the error \( \dot{e} \), will converge to zero exponentially with rate \( \alpha \) which benefits the robustness, as pointed out in Anderson and Johnson [1]. By choosing \( \alpha = \lambda \) and \( \Lambda = \lambda I \), the next control law is obtained:

\[ H u = M (\ddot{q}_d - 2\lambda \ddot{q} + \lambda^2 \ddot{q}) + C \dot{q}_r + w + g + n \]  \hspace{1cm} (6.5)

This is nearly equal to the "computed torque" strategy. The resulting error dynamics are critical damped.
Chapter 7 : The XY-table

The xy-table is a test-apparatus at the WFW laboratory at Eindhoven University of Technology and is mainly being used to investigate control laws on their practical applicability. The discussed (non)adaptive control laws for flexible manipulators have been applied to the xy-table. In this chapter, a brief discussion of this test "robot" will be given.

The xy-table can be pictured schematically as follows:

Figure 7.1 : The xy-table

The end-effector of the system is a slide with mass $m_e$ which can move in the horizontal xy-plane by means of three slides, namely two slides in x-direction and one slide in y-direction. The xy-table has three degrees of freedom, $x_1$, $x_2$ and $y$, and two inputs, $T_1$ and $T_3$. The input torques $T_1$ and $T_3$ act directly on the beltwheels of slideway 1 respectively 3. The degree of freedom $x_2$ is driven by the flexible spindle, situated between $x_1$ and $x_3$. Because of the presence of the flexible spindle, the system can be considered as a flexible manipulator. It is possible to use spindles with different spring stiffnesses, so the magnitude of flexibility can be changed. The translational degrees of freedom $x_1$, $x_2$ and $y$ can directly be converted to rotational degrees of freedom $\varphi_1$, $\varphi_2$ and $\varphi_3$ by using the radii of the beltwheels. These angles are measured during control by incremental encoders. Furthermore, a Coulomb friction acts on each degree of freedom.

Since the xy-table has three degrees of freedom, the model that describes its behaviour contains three differential equations. These equations are given in appendix B; see v.d. Molengraft [16]. They can be written in the same structure as formula (2.2):

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + Kq + w(\dot{q}) = Hu$$

with:

$$q = [\varphi_1 \ \varphi_2 \ \varphi_3]^T \ \ \ \ \ \ \ \ \ \ \ \text{and} \ : \ u = [T_1 \ T_3]^T$$
The matrices $M, C, K$ and $H$ and vectors $w$ and $u$ are also given in appendix B.

It should be noted that other research projects have been executed, using the xy-table, including identification. Knowledge about the system parameters, obtained by these studies, has been used during this research project.

For practical implementation, the model can be used to design control laws, for example, the (A)CTCR controller. However, the model and associated control law can also be implemented in a computer program to simulate the closed-loop dynamics. Performing simulations is especially recommendable if there is no experience with the designed control law. With a good model, the performance of the real system can be predicted quite well. Both methods, experiments and simulations, have been used and their results will be presented in the next chapters.

7.1: Analysis of Model XY-table

Before the model, as given in appendix B, is used it is checked for validity. This is done by comparing the pulsresponse$^1$ of the xy-table with the pulsresponse of the model. The knowledge of the system parameters provided by previous research is used in the model. The Fast Fourier Transformations of the different pulsresponses are compared. Here, only the results of generalized coordinate $\varphi_i$ will be given because they are most clear. The FFT’s for two different spring stiffnesses, $k=0.50$ Nm/rad and $k=3.16$ Nm/rad, are shown below. The end-effector was positioned in the middle of the xy-table.

![FFT's of pulsresponse](image)

**Figure 7.2**: FFT’s of pulsresponses

Because the spring stiffness $k=3.16$ Nm/rad is quite large, $\varphi_2$ is almost equal to $\varphi_1$, and the xy-table can practically be considered stiff. If $k=0.50$ Nm/rad, the FFT of the experimental pulsresponse shows two eigenfrequencies. The first eigenfrequency of about 6 Hz moves to higher frequencies with increasing spring stiffness $k$, while the second eigenfrequency of 12 Hz is stationary. If $k=3.16$ Nm/rad, only one eigenfrequency is visible. This stationary eigenfrequency is due to springs connecting the slides to the belts, which are present to

---

$^1$ In practice, it is impossible to supply an impuls, therefore a puls is applied which provides a maximum torque of 2.56 Nm, during 0.03 seconds.
smooth the forces on the belts. Simulations show only one eigenfrequency because the model of the xy-table does not contain these springs.

It is possible to eliminate the influence of the springs at the belts by fastening the sides to the belts. Then, the FFT are:

![Figure 7.3: FFT's pulsresponses with fastened springs](image)

The second eigenfrequency has nearly disappeared. The eigenfrequency of the spring with spring constant $k=3.16$ Nm/rad is about 12 Hz, thus identical to the eigenfrequency of the springs at the belt. In figure 7.2, they cannot be distinguished because the two eigenfrequencies overlap. This could be expected from figure 7.2 because the eigenfrequency due to spring $k$ cannot disappear.

An important aspect that should be noted is that for spring constant $k=0.50$ Nm/rad, the influence of the spring between $\varphi_1$ and $\varphi_2$ is bigger than the influence of the springs at the belts. With the (A)CTCR control, it is in principle possible to account for the beltsprings as well. But they are not dominant and for sake of simplicity they are omitted.

In the figures 7.2 and 7.3, the upper, dashed lines for the FFT-simulations do not match the lower, solid lines for the FFT-experiments. This is due to uncertainty about the real system parameters. Actually, the (im)pulsrespons of the model does not have to fit the reality because an adaptive control law will be used. However, it is necessary to make realistic simulations. Besides, the model with the best known parameters is used as a basis for the filter design, which will be discussed in chapter 8.
Chapter 8: Design (Adaptive) Computed Torque Computed Reference Controller for the XY-table

The xy-table has been used to investigate the ACTCR controller by means of simulations and experiments. Before the results are given, first the design of the controller and the observer will be discussed.

If the xy-table is to be controlled by the ACTCR strategy, the next control law has to be used:

\[ Hu = \dot{M}(q)\ddot{q} + \dot{C}(q,\dot{q})\dot{q} + \dot{K}q + \dot{\psi}(\dot{q}) - K_q\dot{q} \]  

with: \( \dot{M}, \dot{C}, \dot{K}, H, \dot{\psi}, u \) as in appendix B, where the system parameters are replaced by their estimates.

In order to reduce the number of computations, some constants are measured. These parameters, \( b, l, r_x, r_y \), will not occur in the vector of estimated parameters (see appendix B for their values).

If the terms of (8.1) are rewritten, this gives:

\[ u_1 = \dot{M}_1\ddot{\varphi}_1 + \dot{M}_2\ddot{\varphi}_2 + \dot{C}_1\dot{\varphi}_1 + \dot{C}_2\dot{\varphi}_2 + \dot{K}_1\varphi_1 + \dot{K}_2\varphi_2 + \dot{\psi}_1\text{sign}(\varphi_1) - Kd_1\dot{\varphi}_1 - Kd_2\dot{\varphi}_2 - Kd_3\dot{\varphi}_3 \]  
\[ u_2 = \dot{M}_2\ddot{\varphi}_2 + \dot{M}_3\ddot{\varphi}_3 + \dot{C}_2\dot{\varphi}_2 + \dot{C}_3\dot{\varphi}_3 + \dot{K}_2\varphi_2 + \dot{K}_3\varphi_3 + \dot{\psi}_2\text{sign}(\varphi_2) - Kd_2\dot{\varphi}_2 - Kd_3\dot{\varphi}_3 - Kd_4\dot{\varphi}_4 \]  
\[ u_3 = \dot{M}_3\ddot{\varphi}_3 + \dot{M}_4\ddot{\varphi}_4 + \dot{C}_3\dot{\varphi}_3 + \dot{C}_4\dot{\varphi}_4 + \dot{K}_3\varphi_3 + \dot{K}_4\varphi_4 + \dot{\psi}_3\text{sign}(\varphi_3) - Kd_3\dot{\varphi}_3 - Kd_4\dot{\varphi}_4 - Kd_5\dot{\varphi}_5 \]

The ACTCR control law, as discussed in chapter 3, can be used if the number of desired trajectories is equal to the number of inputs. In case of the xy-table, this means that the desired trajectory of the end-effector has to be converted to desired trajectories of two generalized coordinates. However, all three generalized coordinates are necessary to fix the position of the end effector. Consequently, the discussed (A)CTCR control is not an output control strategy for the xy-table, but rather a tracking control law for two generalized coordinates. A general output control theory for the (A)CTCR control law is presented in Lammerts [13]. The main goal of this research is to investigate the ACTCR controller and not to control the end-effector, so the above problem is not relevant.

Now, there are two possibilities to convert the desired trajectory of the end effector \( x_d(t) \) and \( y_d(t) \) into desired trajectories of two generalized coordinates, under the assumption that the deformations are negligible small:

1. \( \varphi_{1d} = x_d/r_x, \varphi_{3d} = y_d/r_y \), with \( r_x \) and \( r_y \) radii of the beltwheels. This means that the desired trajectory \( \varphi_{2d} \) is a priori unknown.

   A reference trajectory for \( \varphi_2 \) will be computed during control, by solving the second-order differential equation (8.3) for the unknown variable \( \varphi_2 \). Simulations showed that this method yields a stable closed loop behaviour.

2. \( \varphi_{2d} = x_d/r_x, \varphi_{3d} = y_d/r_y \), so \( \varphi_{1d} \) is a priori unknown.

   Now, reference trajectory \( \varphi_1 \) should be solved from (8.3). Contrary to the choice
above, this will lead to instability. Obviously, the differential equation for solving \( \varphi_y \) is responsible for the instability. Simulations have been executed for the two mass-spring system of figure 3.1, which yielded a stable behaviour if \( x_y \) and is computed, as well as if \( x_x \) is computed. So, the instability is due to the specific structure of the xy-table (see further, Lammerts [13]).

In the following, only the former method will be used.

The desired trajectory of the end-effector is a circle, which has its centre in the middle of the xy-table. The radius of the circle is \( R=0.25 \) m and the angular frequency is \( \omega=\pi \) rad/s. Hence,

\[
\begin{align*}
x_d(t) &= \mathbf{c}_p_x - R \cos(\omega t) \\
y_d(t) &= \mathbf{c}_p_y + R \sin(\omega t)
\end{align*}
\]  

with \( \mathbf{c}_p_x=0.679 \) m and \( \mathbf{c}_p_y=0.425 \) m.

At \( t=0 \), the end-effector is positioned at \((x_d(t=0), y_d(t=0))\), thus \( e(t=0)=0 \).

8.1 : Discrete Implementation

The (A)CTCR control law is discretized to control the xy-table. The differential equations that have to be solved are discretized by using:

\[
\begin{align*}
\dot{x}_i &= \dot{x}_{i-1} + T_s \ddot{x}_{i-1} \\
\dot{y}_i &= \dot{y}_{i-1} + T_s \ddot{y}_{i-1} + \frac{T_s^2}{2} \dddot{x}_{i-1}
\end{align*}
\]  

(8.6)

where \( x \) is a time dependent vector and \( T_s \) is the sample time. The sample frequency \( f_s=1/T_s \) is 200 Hz for all experiments.

All differential equations can be solved using data of the last discrete time point. So, to start the control process, initial values are required, which are obtained by the initial position of the end-effector and by the desired trajectory at \( t=0 \). However, the reference trajectory \( \varphi_y \) has to be calculated from (8.3). Because its behaviour is unknown, the next starting values are used: \( \varphi_y(t=0)=0 \) rad and \( \dot{\varphi}_y(t=0)=0 \) rad/s. The reference acceleration \( \ddot{\varphi}_y \) at \( t=0 \) can now be calculated using (8.3).

8.2 : Kalman Observer XY-table

The position as well as the velocity of all generalized coordinates must be available if the (A)CTCR controller is used. In the case of the xy-table, only positions are measured by incremental encoders. To obtain estimates of the velocities, an observer is used. It requires the measurements of the positions at \( t_i \), the input vector at \( t_i \) and knowledge of the dynamics in order to predict the position and the velocity of all generalized coordinates at \( t_{i+1}=t_i+T_s \). The input vector at \( t_{i+1} \) is calculated with these estimated positions and velocities and is applied to the system at \( t_{i+1} \). In appendix C, a complete description of the designed Kalman observer is given. The basis of the observer design is the model described in appendix B and the knowledge of the system parameters. Because this research is focused on a control law and not on an observer, the assumption of parameter knowledge is justified. The only concern is to obtain good estimates of the positions and velocities.
Chapter 9: Computed Torque Computed Reference Control of the XY-table

Simulations with the non-adaptive control laws have been performed first to get an idea of the correct control parameter setting. Unfortunately, this is a trial-and-error method because, so far, no good procedure is available to tune the control parameters appropriately. The setting obtained by simulations is used as a guideline for the experiments. These values have to be decreased when they are applied in a practical situation where unmodeled dynamics are present.

Firstly, the results of the CTCR controller are presented. This is done because it is an introduction to the adaptive controller and some of its general aspects will be explained here. These considerations will also be valid for the adaptive control law, for which the results will be discussed in chapter 10. The results of the non-adaptive CTCR control law will be used also to comparatively evaluate the performance of the adaptive CTCR controller.

9.1: Results of Computed Torque Computed Reference Control

A suitable choice of the feedback matrix $K_d$ and the matrix $\Lambda$, containing the different break-frequencies of the first order filter $\dot{e}+\Lambda e = \dot{\gamma}$, is:

$$K_d = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} \quad \Lambda = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

The matrix $\Lambda$ is of size $\mathbb{R}^{3 \times 2}$, because only the generalized coordinates $\varphi_1$ and $\varphi_3$ have an a priori known desired trajectory.

If the CTCR control law (8.1) is discretized and used to control the flexible xy-table, with $k=0.50$ Nm/rad for the stiffness of the spindle, the next tracking errors are obtained for both experiments and simulations:

![Figure 9.1: Tracking errors](image)

It is assumed that the system is not exactly known. The parameter vector used in the
The control law is 70% of the "actual" parameter values, supplied by earlier research projects.

The results of the experiment and the simulation are quite similar for the tracking error $e_1$, i.e. the behaviour of the tracking error is similar. Inaccuracy of the "actual" parameter vector is probably the cause of the better results of the experiment. The difference for tracking error $e_3$ is larger, but the tendency is similar. The deviations between experiment and simulation is due to parametric uncertainties, unmodeled dynamics, discretisation and the use of an observer.

From the results of the experiment, it is clearly seen that the tracking error $e_3$ has an offset; it does not fluctuate around zero (what would be expected intuitively and what is seen with simulation). The same is true for tracking error $e_1$, but it can hardly be noticed. The reason for this will be given at the end of this chapter. Now, the consequences of these offsets will be discussed.

The reference trajectory $q_r$ is defined as:

$$q_r = q_d - \Lambda \int_0^t (q - q_d) dt$$

In theory, it can be proven that, with exact system knowledge, $e_r = q - q_r$ tends to a constant as $t \to \infty$, which finally establishes a constant difference between $q$ and $q_r$. Because of the parameter error, the combined error $\dot{e}_r$ will not stabilize and $e_r$ is expected to fluctuate around a certain value as $t \to \infty$.

In the experiment with the xy-table, the tracking error $e_1$ does not become zero (or its average is not zero). Hence, the reference trajectory $\phi_{1r}$ keeps growing because the integral of the tracking error increases/decreases in time, see figure 9.2. The same appears in $\phi_3$-direction. Consequently, the computed reference trajectory $\phi_{3r}$ does not stabilize either. The reference trajectory $\phi_{3r}$ drifts away faster than the reference trajectory $\phi_{1r}$. Because $\lambda_1 = \lambda_3$, this indicates that the offset of the tracking error $e_3$ is larger than the offset of $e_1$. This can also be seen in figure 9.1.

![Figure 9.2: Reference trajectories experiment](#)

The results of the simulation with parameter inaccuracy are as expected. The error $e_r = q - q_r$
does not become constant. Therefore, the difference between the reference trajectories and their generalized coordinates is not constant but varies periodically (\( \varphi_3 \) and \( \varphi_{Sr} \) are given in figure 9.3). If, instead of 70% of the system parameters, full knowledge of the system is used, simulations show that \( e_r \) becomes constant and \( \dot{e}_r \) converges to zero.

![Figure 9.3: Reference trajectory simulation](image1)

The spring deformation is seen in figure 9.4. The simulated spring deformation \( \varphi_1 - \varphi_2 \) is much more oscillatory than the experimental deformation, while the frequencies of the deformations agree.

![Figure 9.4: Spring deformation](image2)

The inputs \( u_1 \) and \( u_3 \) are quite different for simulation and experiment; see figure 9.5.

![Figure 9.5: Inputs \( u_1 \) and \( u_3 \)](image3)

During simulation, the input \( u_1 \) stabilizes the deformation of the spring, while input \( u_3 \) shows a smooth behaviour. However, in the experimental case, the situation is inverse. There, the input \( u_1 \) is not as oscillatory, which is due to the less dynamic behaviour of the spring. Input \( u_3 \) contains an extra vibrating component that is not shown on simulation. In order to find the cause of this oscillation, two extra experiments have been executed with different desired trajectories. The new desired trajectories have angular frequencies of \( 1/3*\pi \) and \( 2/3*\pi \) rad/s, while this used to be \( \pi \) rad/s. In both cases, the input \( u_1 \) showed an oscillation. The frequency of the vibration decreased with lower angular frequencies. Therefore the relation between the input \( u_1 \) and the position \( \varphi_3 \) has been investigated; see figure 9.6.
For clarity, only one period has been plotted, but after the initial transient has damped, the revolutions are identical. For all three experiments, the peaks of the oscillating component occur at approximately the same position, thus a position dependent phenomena exists. It appeared that the shaft of the bearing belt is bent, which introduced a harmonic friction component. The controller tries to compensate for this friction, in which it does not succeeds totally, because a small, position-dependent oscillation is seen if the tracking error $e_3$ is plotted as function of the position (see figure 9.7 for angular frequency of $\pi$ rad/s).

If the angular frequency of the desired trajectory is large, the inertia forces are large and thus the input is larger. This explains the difference in steepness in figure 9.6.

To investigate whether the slight vibration of input $u_i$ is to stabilize the spring or is due to a similar harmonic friction, the same plots are made for input $u_i$ (figure 9.8 and 9.9).

As shown in figure 9.8, an harmonic, position-dependent component is present. Besides, there is an extra stabilization of the spring deformation, which is larger for higher angular frequencies. This is seen when the end-effector changes direction (in the lower right and upper left corner of figure 9.8).

9.2: PID-Feedback

The drift of the reference trajectories is undesirable because the controller does not settle,
which gives an incorrect picture of the properties of the CTCR controller.

If an extra term is added to the control law, containing the integral of the tracking errors, the offsets will disappear. The new control law is:

\[ H u = M(q) \ddot{q}_r + C(q, \dot{q}) \dot{q}_r + K \dot{q}_r + w(q, \dot{q}) + g(q) + n(q, \dot{q}, t) - K_d \dot{e}_r - K_p e_r \]  

(9.1)

with: \( K_p \) : positive definite matrix

and: \( e_r = e + \Lambda \int e \, dt \)

Stability can be proven by the Lyapunov function:

\[ V = \frac{1}{2} e_r^T M \dot{e}_r + \frac{1}{2} e_r^T (K + K_p) e_r \]

which yields:

\[ \dot{V} = -e_r^T K_d \dot{e}_r \]

By adding the integral term, the closed-loop dynamics is:

\[ M \ddot{e}_r + C \dot{e}_r + K_d \dot{e}_r + (K + K_d) e_r = 0 \]  

(9.2)

This means, in fact, that the "spring constant" of the closed-loop dynamics is increased. By taking \( K_p \) diagonal, the whole vector \( e_r \) will become zero. Without the integral term, the vector \( K \dot{e}_r \) will tend to zero, as seen in chapter 3. For the xy-table, this results in convergence to zero of \( e_{r_x}, e_{r_y} \), which can be derived by considering the structure of the stiffness matrix \( K \), as given in appendix B.

If \( K_p \) is chosen as:

\[
K_p = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}
\]

the tracking errors shown in figure 9.10 are obtained.

The average of both tracking errors is zero after the initial transient has damped, which results in reference trajectories that do not drift away. However, the start of the controller with the integral term is more oscillatory, especially in \( \varphi_r \)-direction.

9.3 : Offset of Tracking Errors

A solution to the offset of the tracking errors has been discussed in the previous paragraph. The cause is not clear yet. In this paragraph, the aspect responsible for the
offsets will be given.

Firstly, the filter has been checked to determine the quality of the estimates of the position and the velocity.

After the experiment with the CTCR controller, the error of the estimated position can be calculated easily by subtracting the estimated position from the measured position. The velocity error of the filter can be found by using a central difference scheme on the measured positions. The obtained "real" velocities can be compared to the estimated values. Because the velocity passes zero during the experiment, a proportional error can not be found. Therefore the Root Mean Square (RMS) and the average of the filter errors and the "real" positions/velocities are analyzed.

In table 9.1 the RMS and average of the position errors and velocity errors are given:

<table>
<thead>
<tr>
<th>RMS [rad] (x10^5)</th>
<th>e_1</th>
<th>e_2</th>
<th>e_3</th>
<th>e_1</th>
<th>e_2</th>
<th>e_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.28</td>
<td>0.77</td>
<td>1.9</td>
<td>80</td>
<td>128</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>average [rad] (x10^5)</td>
<td>0.17</td>
<td>-1.9</td>
<td>1.3</td>
<td>49</td>
<td>-333</td>
<td>91</td>
</tr>
</tbody>
</table>

Table 9.1: Filter errors

Note: e and \( \dot{e} \) are vectors with filter errors.

In the ideal case, the filter would yield filter errors with average zero and with minimum variance. From table 9.1, it is seen that both the position and velocity filter errors have a non-zero average. The conclusion that the designed filter does not work appropriately, is not correct, because if the RMS of the filter errors are compared to RMS of the "actual" position and velocity, the next table can be made, where:

\[
f = \frac{\text{RMS(filter error)}}{\text{RMS("actual")}} \times 100\%
\]
The factor $f$ is a measure of the quality of the filter. As seen in Table 9.2 the positions are estimated very well. The velocity filter errors are larger, but still they are within acceptable values. Thus, the filter yields a good estimate of the state one sample-time ahead, but the non-zero average of the filter errors is undesirable, because it leads to the offset of the tracking errors.

It appeared that the non-zero average is not a property of the filter, but is due to initialization errors. Before an experiment can be done, the incremental encoders have to be initialized. The end-effector moves to its origin where the encoders are set to zero. The axis $\phi_2$ will however not reach its zero-point because of the presence of the spring and the friction, so its initialization is wrong. Besides this, the axes $\phi_1$ and $\phi_3$ are approximately positioned at the origin.

The position of the axes before initialization have been corrected by pushing the slides against the cushions. The cushions are safety precautions, so the xy-table is not damaged if instability occurs. The CTCR controller with a 70% parameter vector now gives the next tracking errors:

![Tracking errors with correct initialization](image)

The tracking error $e_1$ does not have an offset, while the offset of $e_2$ is decreased. Obviously, the initialization is responsible for the offset of the tracking errors.

Suppose the zero-point of the xy-table does not coincide with the zero-point of the model. Because the inertia matrix $M$ and Coriolis/centrifugal matrix $C$ are dependent of $q$, a wrong initialization yields wrong matrices $M$ and $C$. Since the inertia matrix is used for the Kalman observer ($C$ matrix has been neglected; see appendix C), this consequently leads to wrong estimates of the positions and velocities. If the xy-table is considered stiff, the model is independent of $q$ and an incorrect initialization is not relevant.

Brevoort [3] already did a research about the CTCR controller, but he did not use the

<table>
<thead>
<tr>
<th>pos. $\phi_1$</th>
<th>pos. $\phi_2$</th>
<th>pos. $\phi_2$</th>
<th>vel. $\phi_1$</th>
<th>vel. $\phi_2$</th>
<th>vel. $\phi_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>0.04</td>
<td>0.12</td>
<td>1.48</td>
<td>2.30</td>
<td>2.77</td>
</tr>
</tbody>
</table>

Table 9.2: Filter quality
correct origin (his origin lay in the middle of the xy-table).

Although the offset of the tracking error $e_3$ is decreased, it still is not zero. This may be due to improper conversion of the translational coordinates $x_1$, $x_2$ and $y$ to rotational coordinates $\varphi_1$, $\varphi_2$ and $\varphi_3$. The measurements of the encoders are internally converted to positions $x_1$, $x_2$ and $y$ with regard to the middle of the xy-table. The CTCR control program tries to restore the rotational coordinates, by using the radii of the beltwheels and the position of the middle of the xy-table. An inaccuracy in this conversion (measurement uncertainty) is possibly the reason that tracking error $e_3$ still has an offset.

This explanation of the offsets was found at the end of the research. There was no time left to do all the experiments again. The experimental results of the next chapters will therefore contain offsets or an extra integral term is used. However, different results can be compared because the experiments can be reproduced quite satisfactory, which means that the initialization errors are constant.
Chapter 10: Adaptive Computed Torque Computed Reference Control of the XY-table

In chapter 9, the CTCR control of the xy-table has been discussed. Most of the properties of the CTCR controller are also valid for the ACTCR controller, like offset of the tracking errors and the oscillatory component of the inputs.

The subject of this chapter is the adaptive control of the flexible xy-table. The results of the adaptive control law (8.1) will be given as well as results obtained by modifications and/or extensions of this control law.

10.1: Results of Adaptive Computed Torque Computed Reference Control

The ACTCR controller is used to control the flexible xy-table. The obtained tracking errors, the estimates and the input signals will be discussed. The interaction between these elements will also be considered.

The settings of the feedback matrix \( K_d \) and \( \Lambda \) are, again, chosen as:

\[
K_d = \begin{bmatrix}
0.1 & 0 & 0 \\
0 & 0.01 & 0 \\
0 & 0 & 0.01
\end{bmatrix}
\quad \Lambda = \begin{bmatrix}
5 & 0 \\
0 & 5
\end{bmatrix}
\]

The next parameters are adjusted on-line:

\[
a = \begin{bmatrix}
J_1 + m_1r_x^2 & m_2 & J_3 & m_y & m_e & k & w_1 & w_2 & w_3
\end{bmatrix}^T
\]

The corresponding adaptation matrix \( \Gamma \) has to be determined by trial-and-error. The values obtained by simulations were too large for the experimental case, and led to instability. After a lot of experiments, finally an acceptable setting was found:

\[
\Gamma = \text{diag}(1.5 \cdot 10^{-4}, 3 \cdot 10^{-1}, 2 \cdot 10^{-7}, 8 \cdot 10^{-1}, 1 \cdot 10^{-1}, 4 \cdot 10^{-4}, 2 \cdot 10^{-1}, 5 \cdot 10^{-4}, 5 \cdot 10^{-3})
\]

The tracking errors \( e_1 \) and \( e_3 \) of the ACTCR control law with the parameters setting as above are:

![Tracking errors](image1)

Figure 10.1: Tracking errors
Compared to the tracking errors of the CTCR controller, the results are better. The learning capability of the adaptive controller clearly improves the performance. In the next table, the RMS of the errors during the last cycle are given for the CTCR and the ACTCR controller.

<table>
<thead>
<tr>
<th></th>
<th>CTCR</th>
<th>ACTCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>e₁</td>
<td>4.39</td>
<td>0.53</td>
</tr>
<tr>
<td>e₃</td>
<td>5.29</td>
<td>1.91</td>
</tr>
<tr>
<td>RMS [rad]</td>
<td>(x 10⁻³)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.58</td>
<td>5.58</td>
</tr>
<tr>
<td>RMS [mm]</td>
<td></td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.02</td>
</tr>
</tbody>
</table>

Table 10.1: RMS tracking errors

The tracking errors are improved by 88% respectively 64%. This is due to the parameter adaptation which estimates nine parameters on-line. Some of the estimated parameters achieve an approximately constant value, while other estimates are only stabilized but keep oscillating. In figure 10.2, the estimated parameters $\hat{a}_2$, $\hat{a}_4$ and $\hat{a}_5$ ($m_2$, $m_3$, and $m_4$) are shown.

The estimates are adjusted much in the beginning and then stay approximately constant. The computed reference acceleration $\phi_{2r}$ is responsible for this behaviour. In figure 10.3, it can be seen that $\phi_{2r}$ has a large pulse immediately after $t=0$. The reason for this large acceleration is the wrong initial condition of the unknown reference trajectory $\phi_{2r}$ and reference velocity $\phi_{2r}$. The estimated parameters $\hat{a}_2$, $\hat{a}_4$ and $\hat{a}_5$ are mass-parameters, and their adaptation algorithm contains $\hat{a}_{2r}$. Therefore they are changed very much in the beginning.

The estimates drift slightly during control, partly due to unmodeled dynamics and partly due to the initialization errors.

The estimated parameters $\hat{a}_1 (J_1+m_1r_x^2)$ and $\hat{a}_3 (J_3)$ do not converge to a "constant" value, as can be seen in figure 10.4.
These estimates keep oscillating, just like the estimated friction parameters $\hat{a}_2 (=w_1)$ and $\hat{a}_3 (=w_2)$ in $\varphi_1$ and $\varphi_3$-direction (figure 10.5).

A non-constant parameter leads to this behaviour. After some tests, it appeared that the friction parameter is not constant over the desired trajectory. The adaptation algorithm tries to follow the time-varying friction parameter, which results in oscillating parameter estimates. A further discussion of this aspect will be given in paragraph 10.3.

Now, only the estimate $\hat{a}_6 (=k)$ and $\hat{a}_8 (=w_2)$ remain; these can be seen below (figure 10.6).

The estimate of the friction in $\varphi_2$-direction does approximately become constant, after a slight overshoot.

A strange phenomenon occurs with the estimate of the spring constant $k$, namely it is estimated at zero. By using the regression matrix as given in appendix D, the convergence of $\hat{a}_6$ to zero implies:

$$d_6(t=t_0) - \Gamma_{66} \int_0^\infty (\varphi_{1r}-\varphi_{2r})(\dot{e}_{1r}-\dot{e}_{2r}) \, d\tau = 0 \quad (10.1)$$

In principle, it might be possible that the adaptation factor $\Gamma_{66}$ is accidently chosen such that formula (10.1) holds. However, if $\Gamma_{66}$ is increased from $4 \times 10^{-4}$ to $6 \times 10^{-4}$, the estimated spring constant still tends to zero, although its behaviour is more oscillatory.
Because the estimate of the spring constant tends to zero, it can be concluded that no parameter convergence occurs, i.e. the parameter estimates do not match the real values. The desired trajectory is obviously not rich enough, through which $Y_d$ is not persistently exciting.

The inputs $u_1$ and $u_3$ differ slightly from the inputs of the CTCR controller, see figure 10.7.

During operation, the controller tries to compensate for the Coulomb friction. This means that, if the velocity changes sign, the input $u_1$ shows a step of size $2*\dot{\omega}_1$ and input $u_3$ of size $2*\dot{\omega}_3$. Because the CTCR control is executed with 70% parameter inaccuracy, the friction compensation will not be complete. In the adaptive case, the values of the friction are estimated, but as seen in figure 10.5 they are not constant. The estimated frictions $\dot{\omega}_1$ and $\dot{\omega}_3$ depend on the direction of movement and are, in fact, larger if the end-effector moves in negative direction. So, if the direction changes from negative to positive, the inputs are too large. This especially results in peaks for input $u_1$, because the friction is dominant in $\varphi_3$-direction, as will be seen later. The relative size of the friction in $\varphi_1$-direction is not so large. Therefore, the influence of the direction dependent friction is small for input $u_1$.

Another aspect influencing the step size at each change of direction, is the incorrect friction parameter used in the filter. The estimates of the positions and velocities is dependent of these values. Peaks in the estimated positions and velocities also result in peaks in the inputs.

As in the previous chapter for non-adaptive CTCR control, here the offset of the tracking errors can be removed also by changing the PD-feedback into a PID-feedback. If $K_p$ is chosen as:

$$
K_p = \begin{bmatrix}
0.3 & 0 & 0 \\
0 & 0.1 & 0 \\
0 & 0 & 0.01 
\end{bmatrix}
$$

and the other settings are the same, the tracking errors do not have an offset. The RMS of tracking error $e_t$ during the last cycle is increased from 0.053 rad for the PD-feedback to 0.0677 rad for the PID-feedback. The tracking error $e_t$ only shifts to zero, and its RMS stays at 0.20 rad. The decrease in performance in $\varphi_1$-direction is probably due to a less
suitable adaptation matrix $\Gamma$.

If the initialization of the xy-table is corrected, the results of the ACTCR are better. Just like in the non-adaptive case, the average of the tracking error $e_1$ is zero while the offset of $e_3$ is decreased (figure 10.8). Because the dynamic behaviour of the closed-loop system is different, the adaptation matrix $\Gamma$ has to be changed for better results.

![Figure 10.8: Tracking errors](image)

10.2: Input Vector $u$

In this paragraph, the input vector $u$ will be investigated further by considering the different components, which form the inputs.

If the model fits the reality in a close way and no unmodeled dynamics or unestimated parameters are present, the feedback component $K_d\dot{e}_r$ of the input $u$ will tend to zero. Then the input will only consist of the computed torque part $Y_\eta$. However, in practice there are always model imprecisions, which cause the feedback part not to become zero. The magnitude of $K_d\dot{e}_r$, after the initial transient has damped, is a measure of the model inaccuracy.

If the inputs $u_1$ and $u_3$ are split into a computed torque part and a feedback part, the next plots can be made (figure 10.9):

![Figure 10.9: Inputs split](image)

The error $\dot{e}_i$ is zero at $t=0$ ($e_1=0$ and $\dot{e}_1=0$), so instantly there is no feedback $K_d\dot{e}_i$. In
figure 10.9, this can not be seen because the first points are not plotted. Immediately after 
$t=0$, the error is present to compensate for the model inaccuracies and the wrong estimates.
The computed torque term of $u_1$ also shows a puls at $t=0$, which will be considered later 
in this paragraph. The initial velocity in $\varphi_2$-direction is wrong, leading to an error $\dot{\varphi}_2$, and 
thus a puls in the feedback term at $t=0$. The computed torque part of $u_1$ starts without a 
puls.

The size of $K_d \dot{\varphi}_r$ becomes relatively small, which indicates that the estimates are adjusted 
well and the computed torque part mainly determines the input. The unmodeled position 
dependent friction can be seen in the PD-feedback part. Also the peaks, caused by 
unmodeled direction dependent friction are visible in figure 10.9.

The computed torque part exists of four different components. If the inertia, 
Coriolis/centrifugal, stiffness and friction terms are separated, figure 10.10 is obtained.

![Figure 10.10: Computed torque split](image)

The inertia matrix mainly determines the input $u_1$, while the friction is dominant for $u_3$. 
The computed reference acceleration $\varphi_{2r}$ starts with a peak at $t=0$ as seen in figure 9.3. 
This results in a rather violent behaviour of the inertia part of $u_1$ at $t=0$. The wrong initial 
value of $\varphi_2$, also results in a big influence of the stiffness term. This is decreased rapidly 
because $\varphi_{1r}-\varphi_2$, and the estimate of the spring constant are lowered quickly. These two 
aspects are responsible for the peak in the computed torque part of input $u_1$.
The influence of the Coriolis/centrifugal torques is negligible for both directions, as stated 
before. The increasing estimate of $a_1$ and the decreasing estimate of $a_3$ can be seen in 
figure 10.10 by the progress of the inertia terms.

10.3: Direction Dependent Friction

The friction model of the xy-table can be extended, so it corresponds better to the reality. 
In this paragraph, the Coulomb friction model is replaced by a direction dependent 
Coulomb friction model.

The estimates of the friction $w_1$ and $w_3$ do not become constant for the ACTCR controller, 
which implies that the friction varies over the trajectory. To get an idea of the correct 
friction model, some tests have been executed. Hereby, the spring has been replaced by a 
rigid bar, so that a model of two uncoupled differential equations can be derived for the
xy-table. The desired trajectory of the end-effector is a straight line from \((0,0)\) to \((0.5,0.5)\). If the velocity is constant, the required input torque is equal to the friction torque. The acceleration will never be exactly zero due to unmodeled dynamics, so afterwards the validity should be checked.

It appeared that the total friction contains three parts: a Coulomb friction, a viscous friction and a position dependent friction (see also De Jager [10]). The Coulomb friction parameter highly depends on the direction of movement, while the viscous friction and position dependent friction are less variable. For maximum friction compensation, all three components should be included in the model. Here, the friction model has been extended only with a direction dependent Coulomb friction.

The adaptation algorithm for the parameters of the Coulomb friction is changed now, so that for positive direction a different value can be estimated than for negative direction. The setting of the ACTCR is left unchanged:

\[
K_d = \begin{bmatrix}
0.1 & 0 & 0 \\
0 & 0.01 & 0 \\
0 & 0 & 0.01
\end{bmatrix} \quad \Lambda = \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix}
\]

\[
\Gamma = \text{diag}(1.5\cdot10^4, 3\cdot10^{-1}, 2\cdot10^{-7}, 8\cdot10^{-1}, 1.10^{-1}, 4\cdot10^{-4}, 2\cdot10^{-4}, 5\cdot10^{-4}, 5\cdot10^{-3})
\]

for:

\[
a = \begin{bmatrix} J_1 + m_1 r_x^2 & m_2 & J_3 & m_y & m_e & k & w_1 & w_2 & w_3 \end{bmatrix}^T
\]

The estimates of the friction parameters are:

![Figure 10.11: Estimates](image)

Now, the x-axis represents the sample points, instead of a time sequence. It is seen that for negative direction the estimates \(\hat{\omega}_1\) and \(\hat{\omega}_3\) are larger than for positive direction, while the friction estimate \(\hat{\omega}_2\) is larger for positive direction. The estimates are less oscillatory at the end, compared to figure 10.5. This indicates an improvement of the tracking errors. The remaining oscillation is due to the unmodeled viscous friction. The PD-feedback tries to compensate for the position dependent friction, as already seen.

The RMS of the tracking errors during the last cycle are:
The dynamic behaviour of the friction estimates directly depends on the error $e_r$, since:

$$\dot{a}_i = \Gamma_i \text{sign}(\dot{\phi}_i) e_r$$

for $i=7,8,9$.

A less oscillating behaviour of the estimates corresponds to a smaller error $e_r$. For $e_1$, this results in a lower RMS and average, while for $e_3$ only a decrease of the average is established. This is probably due to the large influence of the position dependent friction. The lowering of the offsets implies that the control law tries to compensate for the initialization errors.

### 10.4: Improving Robustness

The robustness of the ATCRC controller against unmodeled dynamics/unestimated parameters can be improved by adding a sliding term to the control law, as described in chapter 4. The results of the controller with PID-feedback shall be used as reference.

Stability is guaranteed if the elements of matrix $K_s$ are larger than the corresponding influence of the model uncertainties. To adequately choose matrix $K_s$, the results of the PID-feedback controller are analyzed. By using the "real" parameter vector $a$, obtained by earlier research projects, and the acceleration, obtained by a central difference scheme, the model inaccuracies can be determined with equation (4.1). This led to the choice:

$$K_s = \text{diag}(0.5 \ 0.2 \ 0.05)$$

First, the boundary layer $\Phi$ was set equal to the maximum values of $e_r$, obtained by the ACTCR with PID-feedback. This parameter setting did not improve the tracking errors. Therefore, the boundary layer was lowered to:

$$\Phi = [0.8 \ 4 \ 1.5]^T$$

The obtained tracking errors can be seen in figure 10.12.

The RMS of $e_1$ is decreased from 0.087 rad to 0.070 rad. Also tracking error $e_3$ is lowered, namely from 0.19 rad to 0.16 rad.
The errors $\hat{e}_{1r}$, $\hat{e}_{2r}$, and $\hat{e}_x$ are:

The errors $\hat{e}_{1r}$, $\hat{e}_{2r}$, and $\hat{e}_x$ do not stay in their boundary layers during the entire control process, which is due to unmodeled dynamics (improper friction model). The filter estimates of the positions and velocities also contribute to this phenomenon.

The estimates $\hat{\delta}_i \left( I_1 + m_i r_i \right)$ and $\hat{\delta}_f \left( \hat{\omega}_1 \right)$ are shown below:

Because the adaptation process is stopped when the errors are in their boundary layer, the estimates are not so oscillatory, compared to the ACTCR controller with PID-feedback.

10.5 : Exact Flexibility Compensation

In chapter 5, an altered control law is given for a special class of flexible joint
manipulators. For these manipulators, the coupling between the known and the unknown reference trajectories is not only established by the stiffness matrix $K$. In this paragraph, the control law (5.1) will be applied to the xy-table.

The xy-table does not belong to the class of flexible joint robots, but can still be controlled, although it is not an output control. The structure of its matrices $M$, $C$ and $K$ (see appendix B), show besides the coupling by stiffness matrix $K$ also a coupling by the matrices $M$ and $C$. The altered control law, as described in chapter 5, can now be applied, which compensates the flexibilities.

If the same parameter setting is used as before:

$$K_d = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

$$\Gamma = \text{diag}(1.5 \cdot 10^{-6}, 3 \cdot 10^{-7}, 8 \cdot 10^{-1}, 1 \cdot 10^{-1}, 4 \cdot 10^{-4}, 2 \cdot 10^{-1}, 5 \cdot 10^{-4}, 5 \cdot 10^{-3})$$

the results are:

Figure 10.15: Tracking errors

The tracking errors show a less violent behaviour, which is due to the smoother start of the errors $\dot{e}_{1r}$ and especially $\dot{e}_{2r}$.

The stiffness of spring $k$ does not influence the generalized coordinate $\varphi_3$ much. Therefore, the behaviour of $\varphi_3$ is not very different. Only at the start a small improvement is seen. This is due to the less dynamic response of $\varphi_{2r}$, which, by the structure of $M$ and $C$, slightly influences $\varphi_3$. The tracking error $e_3$ converges to the result obtained by the ACTCR controller with $K_q$, because the error $\dot{e}_{3r}$ which activates the most important estimates in $\varphi_3$ direction, $\dot{a}_3 (=J_3)$ and $\dot{a}_3 (=w_3)$, is not very different (see figure 10.16).
The RMS of \( e_r \) during the last cycle is increased due to worse parameter estimates, because estimates dependent of \( e_r \) and, more important \( e_{\alpha} \), are not adjusted enough (see figure 10.17). So, they are less suitable which leads to a larger tracking error \( e_r \).

Because of the offset of the tracking errors, the reference trajectories drift away, like with the ACTCR controller with \( Kq_r \).

The control law with exact flexibility compensation has also been used to execute simulations. To eliminate the influence of the adaptation algorithm, the CTCR strategy with PID-feedback has been used. The simulations have been done with exact knowledge of the system parameter vector. The errors \( e_r \) and \( e_{\alpha} \) are given in figure 10.18.

The tracking error \( e_r \) is better for the CTCR with exact flexibility compensation, while there is hardly a difference for tracking error \( e_{\alpha} \).

10.6: Variable Feedback Matrix

In chapter 6, a possibility to chose the feedback matrix \( K_d \) has been given. It has been applied to the \( xy \)-table. To avoid the influence of model inaccuracies, only simulations have been executed. First, the results of the CTCR controller with \( K_d = \alpha M \) will be given, followed by the adaptive version, where \( K_d = \alpha \dot{M} \).

The inertia matrix \( M \), and therefore, \( K_d \), are not constant over the desired trajectory. To
determine the performance of the controller with $K_d = \alpha M$, it is compared to the CTCR controller with a fixed feedback controller. The used constant matrix $K_d$ is chosen as the average of the variable feedback matrix. Because the behaviour of the controller highly depends on the factor $\alpha$, this has been done for different $\alpha$. The parameter vector is again chosen to be 70% of the actual parameter vector. The RMS of the tracking errors are shown in figure 10.19.

The results of the variable feedback matrix $K_d$ do not differ much from the results of the average, constant, feedback matrix. So, for the xy-table, the choice $K_d = \alpha M$ does not improve the performance of the CTCR controller. For small $\alpha$ the PD-feedback will hardly be present and thus the tracking errors do not change for different $\alpha$. With increasing $\alpha$ the RMS of the errors decrease exponentially with respect to $\alpha$.

Now, the results of the ACTCR controller with $K_d = \alpha \hat{M}$ and adjusted regression matrix, are given. Because of the linear parametrizability, the estimated inertia matrix at $t=0$ is 70% of the actual inertia matrix. It is adapted on line, with large parameter changes. Therefore the results of the variable and fixed feedback matrix do not differ much.

The tracking errors tend to zero with increasing $t$, so the next measure is compared:

$$\text{Sum Of Squares (SOS)} = \int_{t_0}^{\infty} e_i^2 \, dt \quad \text{for } i=1,3$$

In figure 10.20, the SOS is given for different $\alpha$ (the matrices $\Lambda$ and $\Gamma$ are left unchanged).

$$\Lambda = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

$$\Gamma = \text{diag}(1.5\times10^{-6} \ 3\times10^{-1} \ 2\times10^{-7} \ 8\times10^{-1} \ 1\times10^{-1} \ 4\times10^{-4} \ 2\times10^{-1} \ 5\times10^{-4} \ 5\times10^{-3})$$

The difference increases such that for small $\alpha$ the choice $K_d = \alpha \hat{M}$ yields better results and for large $\alpha$ a constant feedback matrix is preferred.

The tracking errors do not converge within an acceptable time if $\alpha$ is chosen lower than 10. For $\alpha=10$, the eigenvalues of the average matrix $K_d$ approximately equals the
systems may yield better performance.

Therefore, the variable feedback matrix is not investigated further. It is possible to use a

\[
\begin{bmatrix}
10^0 & 0 & 0 \\
0 & 10^0 & 0 \\
0 & 0 & 10
\end{bmatrix}
\]

\( \mathbf{K} \)

Eigenvalues of:

\( \text{Figure 10.20: SOS of tracking errors} \)

\( \text{ACTOR Control of the XY-table} \)
Chapter 11: Conclusions and Recommendations

Conclusions

The (Adaptive) Computed Torque Computed Reference control law can be used successfully to control the xy-table, although it does not belong to the class of flexible-joint manipulators. As a consequence, the (A)CTCR control is not an output control, but a tracking control for two of the generalized coordinates. The desired trajectories of these coordinates are reached within an acceptable time, while the deformation of the flexibility remains bounded.

The adaptation algorithm needs some time to adjust the parameters appropriately. Therefore, the CTCR controller settles faster than the adaptive version. If the system parameters are poorly known, the learning capability improves the performance. Still, the difficulty is the choice of the controller parameters, and especially the setting of the adaptation matrix $\Gamma$. The parameter setting that yields the best results is, in fact, unknown. A comparison between adaptive controllers should be done very carefully.

A good model of the manipulator is required for a proper controller design. If all dynamics are included in the model, the tracking errors tend to zero and the parameters estimates will converge to a constant value. In the presence of unmodeled dynamics and/or unestimated parameters, the tracking errors and the estimates do not converge, but will show some fluctuation or drift. The robustness against these aspects can be improved by adding an extra sliding term to the control law.

For the xy-table, the coupling between the known and unknown reference trajectories is not only established by $K_q$. A modified control law can then be used, which includes exact compensation of the flexibilities. It yields better results in $\varphi_1$-direction than the (A)CTCR controller with $K_q$, while the performance in $\varphi_3$-direction is hardly different.

If the feedback matrix is chosen as $K_f=\alpha \dot{M}$ and an adjusted regression matrix is used, stable closed loop dynamics are obtained for the xy-table. For small $\alpha$ this choice yields better results than the corresponding fixed feedback matrix. However, the convergence of the tracking errors to zero is very slow. The factor $\alpha$ must be increased to lead the system settle in an acceptable time interval. With a larger $\alpha$ it is better to apply a fixed feedback matrix.

Aspects not accounted for in the control law may lead to tracking errors which do not fluctuate around zero. This can be resolved by changing the PD-feedback into a PID-feedback. The added integral term forces the offsets of the tracking errors to zero.

The friction compensation might be of major importance in the control of mechanical manipulators. For non-adaptive controllers, the structure and the magnitude of the friction has to be known. In case of the ACTCR, only the model of the friction has to be known. The values of the different components can be estimated on-line. Often, the friction elements are not easy to control e.g. slip-stick friction. Then it is better
Flexible manipulators need to control a flexible manipulator.

Secondly, exponential stable systems possess desirable properties, such as exponential convergence.

The on-line computation of the regression matrix, Y, is a time consuming operation.

Matrices usually determine the closed loop behaviour.

Cautions will be required when the adaptive controller are compared, since the adaptation influences the performance of the composite controller. If the composite controller and composite controller properties are not improved, the indirect control is not competitive. The "indirect" adaptation algorithm is about a real implementation, can be easily integrated into the current control algorithm.

Besides the control law combined with the direct adaptation algorithm it is also possible to...
References


Appendix A: Stability Analysis of the Composite Adaptive Control Law

The stability of the composite adaptive controller with the gradient estimator is guaranteed by a Lyapunov stability analysis and will be given below.

Consider the Lyapunov function:
\[
V = \frac{1}{2} \dot{e}_e^T M \dot{e}_r + \frac{1}{2} e_r^T K_r e_r + \frac{1}{2} \ddot{\alpha}^T P_0^{-1} \ddot{\alpha}
\]  
(A.1)

Differentiating this function and using equations (3.18) and (3.29) yields:
\[
\dot{V} = -\dot{e}_e^T K_d \ddot{e}_r + \dot{\alpha}^T \left[ P_0^{-1} \dot{\alpha} + Y_r \dot{e}_r \right]
\]
\[
= -\dot{e}_e^T K_d \ddot{e}_r - e^T Re
\]  
(A.2)

Barbalat’s lemma insures that \( \dot{V} \to 0 \) as \( t \to \infty \) if \( \dot{V} \) is bounded, thus if \( \dot{e}_r, \dot{e}_r, e \) and \( \ddot{e} \) are bounded. Since \( V > 0 \) and \( \dot{V} < 0 \), \( \dot{e}_r, e \), and \( \dot{\alpha} \) are bounded, so \( q \) and \( \dot{q} \) are bounded too because \( q_d \) and \( \dot{q}_d \) are. The closed loop dynamics (3.18) insures boundedness of \( \ddot{e}_r \).

\( e \) is bounded because:
\[
e = W(q, \dot{q}) \dot{\alpha}
\]

Differentiating yields:
\[
\dot{e} = W(q, \dot{q}) \dot{\alpha} + W(q, \dot{q}) \ddot{\alpha}
\]

\( \dot{q} \) is bounded since \( \ddot{e} \), is. \( \dot{\alpha} \) is bounded too because all the right-hand side terms of (3.27) are bounded. Consequently, \( \dot{e} \) is bounded, and thus \( \dot{V} \) tends to zero if \( t \to \infty \).

Besides the gradient estimator, there are some other strategies to adjust the matrix \( P \). Their stability analysis is similar.
Appendix B : Description of the XY-table

The 3 differential equations of the xy-table are:

\[
\begin{align*}
\left( J_z + m_{x_1} r_1^2 + \frac{1}{3} m \left( \frac{r_1^2}{b} \right) \right) \frac{\partial^2 \phi_1}{\partial x^2} + \left( \frac{1}{2} m_{x_1} + \frac{1}{3} m \left( \frac{r_1^2}{b} \right) \right) \frac{\partial^2 \phi_1}{\partial x \partial y} + \frac{1}{2} m_{x_1} \frac{\partial^2 \phi_1}{\partial y^2} & = k_1 \phi_1 + w \text{sign}(\phi_1) = u_1 \\
\left( \frac{1}{2} m_{x_1} + \frac{1}{3} m \left( \frac{r_2^2}{b} \right) \right) \frac{\partial^2 \phi_2}{\partial x^2} + \frac{1}{2} m_{x_1} \frac{\partial^2 \phi_2}{\partial x \partial y} + \frac{1}{2} m_{x_1} \frac{\partial^2 \phi_2}{\partial y^2} & = k_2 \phi_2 + k_3 \phi_3 + w \text{sign}(\phi_2) = 0 \\
\left( \frac{1}{2} m_{x_1} + \frac{1}{3} m \left( \frac{r_3^2}{b} \right) \right) \frac{\partial^2 \phi_3}{\partial x^2} + \frac{1}{2} m_{x_1} \frac{\partial^2 \phi_3}{\partial x \partial y} + \frac{1}{2} m_{x_1} \frac{\partial^2 \phi_3}{\partial y^2} & = k_4 \phi_3 + w \text{sign}(\phi_3) = u_3
\end{align*}
\]

(B.1)

From the differential equations of the flexible xy-table the next matrices and vectors can be derived:

\[
M(q) = \begin{bmatrix} M_{11} & M_{12} & 0 \\ M_{12} & M_{22} & M_{23} \\ 0 & M_{23} & M_{33} \end{bmatrix} ; \quad C(q,\dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & 0 \end{bmatrix} ; \quad K = \begin{bmatrix} K_{11} & K_{12} & 0 \\ K_{12} & K_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} ; \quad w(q) = \begin{bmatrix} w_1 \text{sign}(\phi_1) \\ w_2 \text{sign}(\phi_2) \\ w_3 \text{sign}(\phi_3) \end{bmatrix} ; \quad H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

with:

\[
M_{11} = J_z + m_{x_1} \frac{1}{3} m \left( \frac{r_1^2}{b} \right) r_1^2
\]

\[
M_{12} = \left( \frac{1}{2} m_{x_1} + \frac{1}{3} m \left( \frac{r_2^2}{b} \right) \right) \frac{\partial^2 \phi_2}{\partial x^2}
\]

\[
M_{13} = \left( \frac{1}{2} m_{x_1} + \frac{1}{3} m \left( \frac{r_3^2}{b} \right) \right) \frac{\partial^2 \phi_3}{\partial x^2}
\]

\[
M_{22} = \left( \frac{1}{2} m_{x_1} + \frac{1}{3} m \left( \frac{r_2^2}{b} \right) \right) \frac{\partial^2 \phi_2}{\partial y^2}
\]

\[
M_{23} = \left( \frac{1}{2} m_{x_1} + \frac{1}{3} m \left( \frac{r_3^2}{b} \right) \right) \frac{\partial^2 \phi_3}{\partial y^2}
\]

\[
M_{33} = M_{23}
\]

\[
M_{12} = M_{21}
\]

\[
M_{13} = M_{31}
\]

\[
M_{23} = M_{32}
\]
**Appendix B: Description of the XY-table**

\[ c_{11} = m \left( \frac{r_1}{b} \right)^2 \psi_x \phi_y \]
\[ c_{13} = m \left( \frac{r_1}{b} \right)^2 \psi_x (\phi_1 - \phi_3) \]
\[ c_{21} = m \left( \frac{r_1}{b} \right)^2 \psi_x b (\frac{b}{r_2} - \phi_y) \]
\[ c_{23} = m \left( \frac{r_1}{b} \right)^2 \psi_x (\frac{b}{r_2} - \phi_y) (\phi_1 - \phi_3) \]
\[ c_{12} = -c_{11} \]
\[ c_{22} = -c_{21} \]
\[ c_{31} = -c_{13} \]
\[ c_{33} = c_{13} \]

\[ K_{11} = K_{12} = k \]
\[ K_{12} = K_{31} = -k \]

Parameters used for simulation:

- \( m_y = 8.5 \) [kg]
- \( m_s = 2.3 \) [kg]
- \( m_1 = 2.3 \) [kg]
- \( m_2 = 2.3 \) [kg]
- \( J_1 = 2.2e-3 \) [kgm²]
- \( J_3 = 1.58e-4 \) [kgm²]
- \( l = 1.23 \) [m]
- \( b = 1.004 \) [m]
- \( r_{x,1} = 10.45e-3 \) [m]
- \( r_{x,2} = 10.50e-3 \) [m]
- \( r_y = 10.55e-3 \) [m]
- \( w_1 = 0.447 \) [Nm]
- \( w_2 = 0.1 \) [Nm]
- \( w_3 = 0.12 \) [Nm]
- \( k = 0.50 \) [Nm/rad]
Appendix C: Kalman Observer XY-table

To estimate the positions and velocities one sample ahead, a Kalman observer is designed, based on a discrete time model (see Kok [11]). First the continuous time description of the system is required. As seen in chapter 7, the xy-table is modeled as:

\[ M(q)\ddot{q} + C(q,\dot{q})\dot{q} + Kq + w(q) = Hu \]  

(C.1)

In practice there will always be some process noise \( d(t) \) and measurement noise \( e(t) \). These are added to the model. Because the inertia matrix \( M(q) \) is positive definite, equation (C.1) can be rewritten as:

\[
\dot{x} = \begin{bmatrix} 0 & I \\ -M^{-1}K & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}(Hu - w - d)
\]

(C.2)

where:

\[ x = [\varphi_1, \varphi_2, \varphi_3, \phi_1, \phi_2, \phi_3]^T \]

The Coriolis and centrifugal forces have been neglected because simulations have shown that they are small compared to the other terms (maximum 2% of total torque \( Hu \)).

Only positions are measured, so the output equation is:

\[ y = \begin{bmatrix} I & 0 \end{bmatrix} x + e = Cx + e \]  

(C.3)

Note: the matrix \( C \) in the formula above is not the Coriolis/centrifugal matrix, but the output matrix of the state space description.

The matrices \( A \) and \( B \) are not constant because the inertia matrix is dependent of \( \varphi_3 \) and \( \varphi_3^2 \) which vary during control. For sake of simplicity it is desirable to have constant matrices \( A \) and \( B \). This can be achieved by using the average values of \( \varphi_{3d} \) and \( \varphi_{3d}^2 \), so the inertia matrix becomes constant (the system parameters are supposed to be known, as discussed in chapter 8). This is motivated by the consideration that a good control system will make \( q_3 \) converge to \( \varphi_{3d} \).

Now, the constant matrices \( A \), \( B \) and \( C \) are known a priori, which can be discretized to obtain the discrete time state space description:

\[ x(i+1) = A_d x(i) + B_d [Hu(i) - w(i) - d(i)] \]

\[ y(i) = C_d x(i) + e(i) \]  

(C.4)
The matrices $A_d, B_d,$ and $C_d$ can be found using the software package MATLAB (c2d($A,B,C,D$, sample time)).

To estimate the state one sample ahead, the observation update is chosen according to Kok [11]:

$$
\dot{x}(i+1) = A_d\dot{x}(i) + B_d[H_u(i) - w(i)] + K^o_d[y(i) - C_d\dot{x}(i)]
$$

When the measurement and process noise are assumed to be white noise sequences, an optimal gain matrix $K^o_d$ can be determined, which yields a minimum variance estimation. The covariance matrices of the measurement and process noise must be available to calculate this optimal gain matrix.

In a previous research by Van Gerwen [9] the covariance of the measurement noise has been estimated by determining the measurement inaccuracy due to the incremental encoders. However this is only done for the encoders of $\varphi_1$ and $\varphi_3$. The variance of the measurement noise of $\varphi_2$ is taken identical to that of $\varphi_3$, so:

$$
R = \text{covariance}(e) = \begin{bmatrix} 3.5 \times 10^{-8} & 0 & 0 \\ 0 & 8.4 \times 10^{-7} & 0 \\ 0 & 0 & 8.4 \times 10^{-7} \end{bmatrix}
$$

The measurement of $\varphi_1$ is more accurate than the measurement of $\varphi_2$ and $\varphi_3$ because of a transmission between the motor axis and the encoder.

The process noise is determined by executing experiments in which the ACTCR controller is used. The gain matrix $K^o_d$ is replaced by $K_d$ which is determined by pole placement. After an experiment the real velocities and accelerations can be calculated by the central difference scheme. If the measured positions and the calculated velocities and accelerations are employed in equation (C.1), an input vector is obtained which should be subtracted from the real inputs to get the process noise $d$. An average over the experiments of the process noise covariances is determined as:

$$
Q = \text{covariance}(B_d) = \begin{bmatrix} 0 & 0 \\ 0.1219 & -0.0441 & -0.0105 \\ 0 & -0.0441 & 0.3350 & 0.0381 \\ 0 & -0.0105 & 0.0381 & 0.1076 \end{bmatrix}
$$

Again, the software package MATLAB is used to calculate the optimal observer gain $K^o_d$ (dlqr($A_d^T, C_d^T, Q, R$)). This yields:

$$
K^o_d = \begin{bmatrix} 1.95 & 0 & 0 \\ -0.07 & 1.72 & 0.04 \\ -0.03 & 0.04 & 1.52 \\ 192.01 & 0.76 & -0.14 \\ -5.31 & 155.32 & 4.92 \\ -3.91 & 5.07 & 130.63 \end{bmatrix}
$$
Appendix D: Regression Matrix \( Y_r \)

The ACTCR control law (8.1) can be written as:

\[
Hu = \dot{\mathcal{M}}(q)\ddot{q}_r + \dot{\mathcal{C}}(q,\dot{q})\dot{q}_r + \dot{K}q_r + \dot{\omega}(q) - K_d\dot{\theta}_e,
\]

with regression matrix

\[
Y_r(q, q_r, \dot{q}_r, \ddot{q}_r) \in \mathbb{R}^{39}\text{.}
\]

\[
Y_{r,11} = \phi_{1r}
\]

\[
Y_{r,22} = T_5\phi_{2r}
\]

\[
Y_{r,33} = \phi_{3r}
\]

\[
Y_{r,44} = T_1\phi_{1r} + T_2\phi_{2r}
\]

\[
Y_{r,45} = T_3\phi_{1r} + (T_5 + T_6)\phi_{2r}
\]

\[
Y_{r,15} = T_2\phi_3\phi_{1r} + \phi_3(\phi_{1r} - \phi_{2r}) + \phi_4(\phi_{1r} - \phi_{2r})(T_4 - \phi_3 T_2)\phi_{2r}
\]

\[
Y_{r,25} = \phi_3(\phi_{1r} - \phi_{2r})(\phi_{1r} - \phi_{2r}) + T_6\phi_{2r} + T_4(\phi_{1r} - \phi_{2r})\phi_{3r} + (T_4 - \phi_3 T_2)[\phi_3(\phi_{1r} - \phi_{2r}) + \phi_3(\phi_{1r} - \phi_{2r})]
\]

\[
Y_{r,35} = T_6(\phi_{1r} - \phi_{2r})\phi_{3r} + T_7\phi_{3r} - T_4\phi_{3r}(\phi_{1r} - \phi_{2r})
\]

\[
Y_{r,16} = \phi_{1r} - \phi_{2r}
\]

\[
Y_{r,26} = \phi_{2r} - \phi_{1r}
\]

\[
Y_{r,37} = \text{sign}(\phi_3)
\]

\[
Y_{r,38} = \text{sign}(\phi_2)
\]

\[
Y_{r,39} = \text{sign}(\phi_3)
\]

(all the other elements of \( Y_r \) are zero)

with:

\[
T_1 = \frac{1}{3}\left( \frac{b}{l_x} \right)^2
\]

\[
T_2 = \left( \frac{b}{l_x} \right)^2
\]

\[
T_3 = \begin{bmatrix} \left( \frac{b}{l_x} \right)^2 & b & 1 \end{bmatrix}
\]

\[
T_4 = \frac{r_x^2}{b}
\]

\[
T_5 = r_x^2
\]

\[
T_6 = \begin{bmatrix} \left( \frac{b}{l_x} \right)^2 & b & 1 \end{bmatrix}
\]

\[
T_7 = r_x^2
\]

and:

\[
a = \begin{bmatrix} J_1 + m_1 l_x^2 & m_2 & J_3 & m_y & m_e & k & w_1 & w_2 & w_3 \end{bmatrix}^T
\]