Composite computed torque control of the XY table with an elastic motor transmission

intermediate report

COMPOSITE COMPUTED TORQUE CONTROL OF THE XY TABLE WITH AN ELASTIC MOTOR TRANSMISSION

theoretical analysis and simulations

by: Gert Brevoord

Professor: J.J. Kok
Coach: Ivonne Lammerts

WFW, Control and identification of Mechanical systems
Department of Mechanical Engineering, Eindhoven University of Technology.

Report no.: 92.030

ABSTRACT

This report gives a summary of the present-day state of my research. The research deals with the controlling of manipulators with flexible transmissions between the actuators and the stiff links. Ivonne Lammerts, member of the WFW group, has developed a control law for this kind of manipulators. This control law deals with tracking the desired trajectory as well as control of the elastic vibrations. The control law is an extended version of the Computed Torque Control strategy and is called the Composite Computed Torque Control strategy (C CTC strategy).
The goal of my research is to test the C CTC strategy and to apply the control law in a practical situation. The xy table, which is situated in the WFW lab, will be used as the test apparatus. By executing simulations with the model of the xy table and by executing experiments with the xy table I have to find the properties of the C CTC strategy.
The present day state of the research is that I have designed a C CTC law for the controlling of the xy table. Further, I have made some theoretical analysis and I have executed a lot of simulations.
During the research I have considered two different situations, a theoretical situation and a practical situation. For the theoretical situation appears that the stability of the system is guaranteed and that the simulation results are good. For the practical situation (the expected situation which will appear during the experiments) it is not possible to guarantee the stability. However, the simulation results are satisfactory, so there will be a reasonable change that the C CTC law will be applicable in practice.

CONTENTS

1. INTRODUCTION

Today industrial robots are used for various purposes. Because of hardware limitations in on-line applications, until now, robot control has been studied extensively under the assumption that the actuator transmissions are stiff and that the links can be modelled as rigid bodies. Therefore, most of today's robots have a very stiff construction in order to avoid deformations and vibrations.
For higher operating speeds, industrial robots should be light-weight constructions to reduce the driving force/torque requirements and to enable the robot arm to respond faster. However a lightweight manipulator may have flexibility in the link structure and elasticity in the transmissions between the actuators and links. For most manipulators, elasticity of the motor transmissions has a greater significance for the design of the controller than the deformation of the flexible links.
A well known approach to improve the behaviour of manipulators is the computed torque control method. In its original version this control method appears to be applicable only to rigid manipulators. If flexibility plays an important role, it often results in an instable system behaviour. Therefore, the control system must deal with control of the elastic vibrations as well as trajectory tracking.
However, it is not possible to find a control input for a flexible manipulator which will accomplish perfect tracking of any desired trajectory in space while totally damping the undesired elastic deflections. It is more realistic to search for a control strategy achieving both a reasonable trajectory tracking and a certain stabilization of acceptable vibrations.
Ivonne Lammerts, member of the WFW-group, has developed such a control strategy. This control strategy is an extended version of the familiar composite torque technique for rigid manipulators, and is called the Composite Computed Torque Control strategy (C CTC strategy). Ivonne Lammerts has proved, for a theoretical situation, that this control strategy will be applicable to systems with one or more flexible transmissions. The question is now: How will the C CTC strategy function in reality. During my research I have to answer this question.
This report gives the present-day state of my research. In this report I will show the C CTC strategy and I will show the C CTC law which I have designed for the controlling of a xy table with a flexible transmission. Further I will show and discuss some theretical analysis and some simulations results. At the end I will give some recommendations to continue the research in the future

2. THE COMPOSITE COMPUTED TORQUE CONTROL STRATEGY

A manipulator with elastic motor transmissions

We consider manipulators that can be modelled as an open chain of n rigid links interconnected bij joints with one degree of freedom per joint. One end of the chain is fixed to the ground and the other end has to follow a specified trajectory in space.
Since each joint allows one relative motion of the connected link, n generalized coordinates are necessary and sufficient to describe the kinematics of the links. These coordinates are the components of a vector $\underline{q}_l \in \mathbb{R}^n$. The desired path of $\underline{q}_l$ in time is denoted by $\underline{q}_{ld} = \underline{q}_{ld}(t)$.
Each joint has its own actuator and its own transmission between the actuator and the driven link. The motor torques (used in a generalized sense, i.e. denoting both torques and forces) acting on the transmissions are the robot control inputs. In this paper, we consider the case in which some or all transmissions are elastically deformable. Then, for each elastic deformation it is necessary to introduce an extra coordinate to desribe the rotation of the rotor of the motor. These extra coordinates are the components of a vector $\underline{q}_m \in \mathbb{R}^e$ with $e \leq n$.
For the sequel it is advantageous to regroup the coordinates $\underline{q}_l$ of the links in two vectors $\underline{q}_s \in \mathbb{R}^{n-e}$ and $\underline{q}_e \in \mathbb{R}^e$, where $\underline{q}_s$ and $\underline{q}_e$ contain the coordinates of the direct driven links (i.e. by the stiff transmission), respectively the coordinates of the elastically driven links (i.e. driven by the elastic transmissions). See fig 2.1.

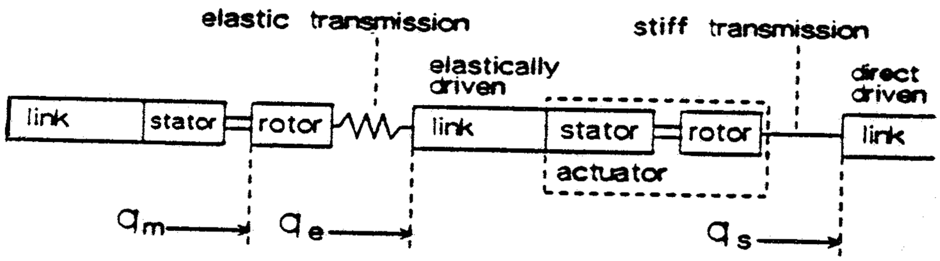


Fig 2.1 Elastically driven links - direct driven links.

This completes the introduction of the total vector of generalized coordinates $\underline{q} \in \mathbb{R}^{n+e}$.

$$\underline{q} = \begin{bmatrix} \underline{q}_s \\ \underline{q}_e \\ \underline{q}_m \end{bmatrix} \tag{2.1}$$

The components of the vector ε defined by

$$\underline{\varepsilon} = \underline{q}_m - \underline{q}_e \tag{2.2}$$

charaterize the deformations of the elastic motor transmissions. Hence, if these transmissions are modelled as massless linear springs, the elastic torques being the components of a vector $\underline{z}_e \in \mathbb{R}^e$ are related to ε by

$$\underline{z}_e = K\underline{\varepsilon} \tag{2.3}$$

where $K \in \mathbb{R}^{e*e}$ is the positive definite diagonal stiffness matrix.

Dynamic model of the flexible robot

Using a Langrangian approach, the dynamic model of the manipulator can be written as:

$$M(\underline{q})\ddot{\underline{q}} + C(\underline{q},\dot{\underline{q}})\dot{\underline{q}} + K_k\underline{q} + \underline{n}(\dot{\underline{q}}) + \underline{g}(\underline{q}) = \underline{u} \tag{2.4}$$

where:

$\underline{q} = [\ \underline{q}_s\ \underline{q}_e\ \underline{q}_m\ ]^T$
M = positive definite inertia matrix.
$C\dot{\underline{q}}$ = is a vector with torques due to te Coriolis and centrifugal effects.
$K_k\underline{q}$ = is the vector with elastic-transmission torques.
$\underline{n}$ = is the vector with torques due to friction.
$\underline{g}$ = is the vector with torques due to gravity.

The model can be given in a detailed form by:

$$m_{ss}\ddot{\underline{q}}_s + m_{se}\ddot{\underline{q}}_e + m_{sm}\ddot{\underline{q}}_m + c_{ss}\dot{\underline{q}}_s + c_{se}\dot{\underline{q}}_e + c_{sm}\dot{\underline{q}}_m + \underline{n}_s + \underline{g}_s = \underline{u}_s \qquad (2.5)$$

$$m_{es}\ddot{\underline{q}}_s + m_{ee}\ddot{\underline{q}}_e + m_{em}\ddot{\underline{q}}_m + c_{es}\dot{\underline{q}}_s + c_{ee}\dot{\underline{q}}_e + c_{em}\dot{\underline{q}}_m + \underline{n}_e + \underline{g}_e + K(\underline{q}_e - \underline{q}_m) = \underline{0} \qquad (2.6)$$

$$m_{ms}\ddot{\underline{q}}_s + m_{me}\ddot{\underline{q}}_e + m_{mm}\ddot{\underline{q}}_m + c_{ms}\dot{\underline{q}}_s + c_{me}\dot{\underline{q}}_e + c_{mm}\dot{\underline{q}}_m + \underline{n}_m + \underline{g}_m + K(\underline{q}_m - \underline{q}_e) = \underline{u}_e \qquad (2.7)$$

Composite computer torque control

The first problem in controlling a rigid-link manipulator with elastic motor transmissions, is that only the desired link coordinates $\underline{q}_{ld} = \underline{q}_{ld}(t)$ can be determined directly from the known desired gripper path, while there is no indication for a certain desired trajectory of of $\underline{z}_e(t)$ or $\underline{q}_m(t)$ (note: $\underline{z}_e = K(\underline{q}_m - \underline{q}_e)$.
To obtain a smooth robot performance in space, we define a reference trajectory $\underline{q}_{lr} = \underline{q}_{lr}(t)$ for the link variables, which will converge to $\underline{q}_{ld}$ after progression in time. Further, the idea is to formulate a 'reference manifold' $\underline{z}_{er}(t)$ (or $\underline{q}_{mr}$) on which the controller tries to keep the elastic-transmissions torques $\underline{z}_e(t)$, instead of trying to supress them totally.

The second problem is that there are more degrees of freedom than control inputs. The goal of the composite controller developed in this chapter is to track the reference trajectory of the links, while stabilizing the elastic vibrations around the specified reference manifold.

It is appealing to try to find the analogue for flexible manipulators of the socalled computed torque control method for rigid robots. However, an elastic-transmission robot does not allow a nonlineair feedback control as for rigid manipulators, since there are less control inputs than degrees of freedom. Here, we choose the next computed torque notation for the (n+e)-th order dynamic model of the flexible robot (Ivonne Lammerts, october 1991).

$$\underline{u} = M(\underline{q})\ddot{\underline{q}}_r + C(\underline{q},\dot{\underline{q}})\dot{\underline{q}}_r + K_k\underline{q}_r + \underline{n}(\dot{\underline{q}}) + \underline{g}(\underline{q}) + K_r\dot{\underline{e}}_r \qquad (2.8)$$

where,

- K_r is a diagonal positive gain matrix.

- $\underline{q}_r$ is a chosen reference trajectory of all system variables; in this case, the vector $\underline{s}=\dot{\underline{e}}_r$ is a sliding service for $\underline{q}$ according to Asada and Slotine (1986).
- $\dot{\underline{e}}_r=\dot{\underline{e}}+\Lambda\underline{e}$ $\quad\forall t\geq t_0$; $\underline{e}_r(t_0)=\underline{e}(t_0)$.
- $\underline{e}=\underline{q}_d-\underline{q}$ is the total error.
- $\underline{e}_r=\underline{q}_r-\underline{q}$ is the total reference error.

Note, that we do know the desired trajectory of the link variables, $\underline{q}_{ld}(t)$ and its time derivatives, but we haven't any indication of how to determine a certain desired trajectory $\underline{q}_{md}(t)$ for the elastic motor rotor variables, unless we make use of above computed torque expression in splitting it up again in a partitioned form according to the equations 2.5, 2.6 and 2.7:

$$\underline{u}_s = m_{ss}\ddot{\underline{q}}_{sr}+m_{se}\ddot{\underline{q}}_{er}+m_{sm}\ddot{\underline{q}}_{mr}+c_{ss}\dot{\underline{q}}_{sr}+c_{se}\dot{\underline{q}}_{er}+c_{sm}\dot{\underline{q}}+\underline{n}_s+\underline{g}_s+K_s\dot{\underline{e}}_{sr} \tag{2.9}$$

$$\underline{0} = m_{es}\ddot{\underline{q}}_{sr}+m_{ee}\ddot{\underline{q}}_{er}+m_{em}\ddot{\underline{q}}_{mr}+c_{es}\dot{\underline{q}}_{sr}+c_{ee}\dot{\underline{q}}_{er}+c_{em}\dot{\underline{q}}_{mr}+\underline{n}_e+\underline{g}_e+K(\underline{q}_{er}-\underline{q}_{mr})+K_e\dot{\underline{e}}_{er} \tag{2.10}$$

$$\underline{u}_e = m_{ms}\ddot{\underline{q}}_{sr}+m_{me}\ddot{\underline{q}}_{er}+m_{mm}\ddot{\underline{q}}_{mr}+c_{ms}\dot{\underline{q}}_{sr}+c_{me}\dot{\underline{q}}_{er}+c_{mm}\dot{\underline{q}}_{mr}+\underline{n}_m+\underline{g}_m+K(\underline{q}_{mr}-\underline{q}_{er})+K_m\dot{\underline{e}}_{mr} \tag{2.11}$$

where K_s, K_e and K_m are diagonal, positive definite gain matrices.

With equation (2.10) it is possible to find a reference trajectory for $\underline{q}_{mr}(t)$, $\dot{\underline{q}}_{mr}(t)$ and $\ddot{\underline{q}}_{mr}(t)$ (under the assumption that $\underline{q}(t)$ and $\dot{\underline{q}}(t)$ and the parameters are known). Now it is possible to formulate the reference manifold $\underline{z}_{er}(t)=K(\underline{q}_{mr}-\underline{q}_{er})$ and to determine the inputs $u_s(t)$ and $u_e(t)$ with (2.9) and (2.11).

3. CONTROLLING THE XY TABLE WITH A COMPOSITE COMPUTED TORQUE LAW

The goal of my research is to test the Composite Computed Torque Control strategy (C CTC strategy), and to find the properties of this control law. For the research I will use the xy table with one elastic transmission as test apparatus. The xy table is situated in the WFW lab.
In this chapter I will show the C CTC law which I have designed for the controlling of the xy table.

The xy table

A schematic representation of the xy table is given in fig. 3.1 (Heeren, 1989 and v.d. Molengraft, 1989).

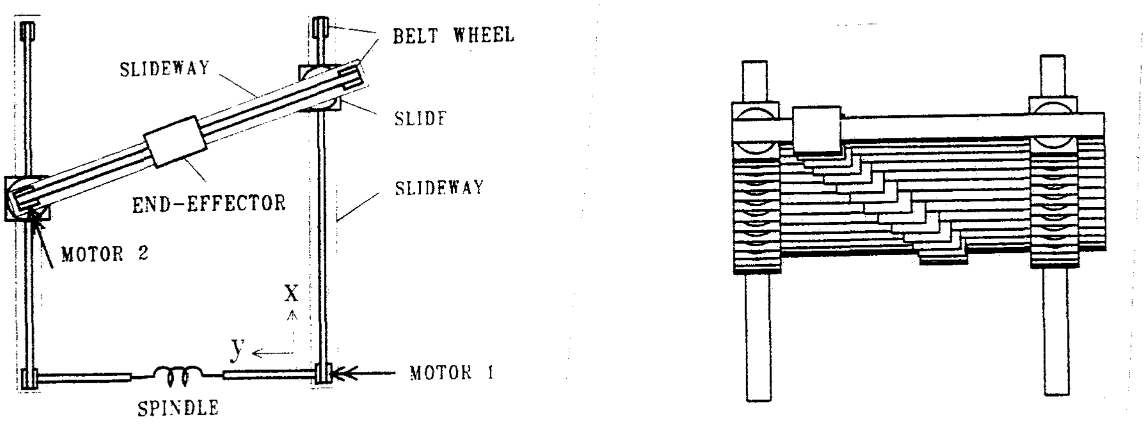


fig 3.1 Schematic representation of the xy table.

For control design we have to choose a suitable model of the xy table. I choose the next model (v.d. Molengraft, 1989).

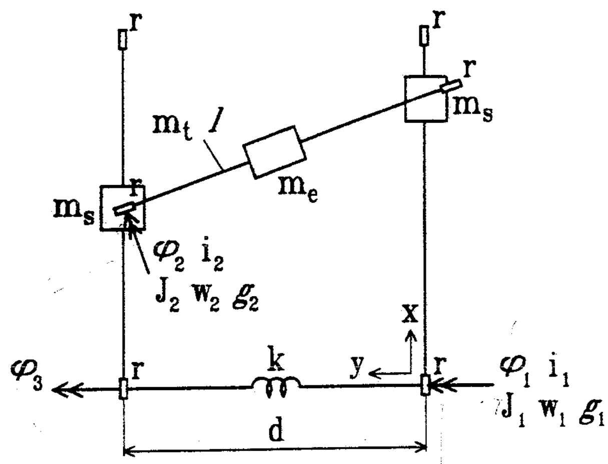


fig 3.2 model of the xy table

See appendix A for a more detailed description of the xy table.

The dynamic model of the xy table can be written as:

$$M(\underline{q})\underline{\ddot{q}}+C(\underline{q},\underline{\dot{q}})\underline{\dot{q}}+K_k\underline{q}+\underline{n}(\underline{\dot{q}}) = \underline{u} \tag{3.1}$$

If M,C and K_k are partitioned in accordance to the partitioning of $\underline{q}$, the dynamic model can be given in detailed form (see also appendix A).

$$m_{22}\ddot{\varphi}_2+m_{23}\ddot{\varphi}_3+c_{23}\dot{\varphi}_3+c_{21}\dot{\varphi}_1+n_{22} = u_s \tag{3.2}$$

$$m_{31}\ddot{\varphi}_1+m_{32}\ddot{\varphi}_2+m_{33}\ddot{\varphi}_3+c_{32}\dot{\varphi}_2+c_{33}\dot{\varphi}_3+c_{31}\dot{\varphi}_1-k(\varphi_1-\varphi_3) = 0 \tag{3.3}$$

$$m_{11}\ddot{\varphi}_1+m_{13}\ddot{\varphi}_3+c_{12}\dot{\varphi}_2+c_{13}\dot{\varphi}_3+c_{11}\dot{\varphi}_1+n_{11}+k(\varphi_1-\varphi_3) = u_e \tag{3.4}$$

The C CTC law

We choose, as shown in chapter 2, the next C CTC strategy:

$$\underline{u} = M(\underline{q})\underline{\ddot{q}}_r+C(\underline{q},\underline{\dot{q}})\underline{\dot{q}}_r+\underline{n}(\underline{\dot{q}})+K_k\underline{q}_r+K_r\underline{\dot{e}}_r \tag{3.5}$$

Splitting the above computed torque expression in a partitioned form according to the equations 3.2, 3.3 and 3.4 gives:

$$u_s = m_{22}\ddot{\varphi}_{2r}+m_{23}\ddot{\varphi}_{3r}+c_{23}\dot{\varphi}_{3r}+c_{21}\dot{\varphi}_{1r}+n_{22}+k_2\dot{e}_{2r} \tag{3.6}$$

$$0 = m_{31}\ddot{\varphi}_{1r}+m_{32}\ddot{\varphi}_{2r}+m_{33}\ddot{\varphi}_{3r}+c_{32}\dot{\varphi}_{2r}+c_{33}\dot{\varphi}_{3r}+c_{31}\dot{\varphi}_{1r}-k(\varphi_{1r}-\varphi_{3r})+k_3\dot{e}_{3r} \tag{3.7}$$

$$u_e = m_{11}\ddot{\varphi}_{1r}+m_{13}\ddot{\varphi}_{3r}+c_{12}\dot{\varphi}_{2r}+c_{13}\dot{\varphi}_{3r}+c_{11}\dot{\varphi}_{1r}+n_{11}+k(\varphi_{1r}-\varphi_{3r})+k_1\dot{e}_{1r} \tag{3.8}$$

where,

$$\begin{aligned} i &= 1,\ 2 \\ \dot{\varphi}_{ir} &= \dot{\varphi}_{id}+\lambda_i(\varphi_{id}-\varphi_i) \\ \varphi_{ir} &= \textit{reference trajectory} \\ \varphi_{id} &= \textit{desired trajectory} \end{aligned} \tag{3.9}$$

Now, we have to define a vector $\underline{q}_l$. In $\underline{q}_l$ we put those variables of which we define a desired trajectory. The number of variables of $\underline{q}_l$ is equal to the number of inputs. In the general case $\underline{q}_l$ exists of those variables which fix the position of the end effector of the robot.
In the case of the xy table we can put two variables in $\underline{q}_l$. The position of the end effector of the xy table is fixed by three variables (φ_1, φ_2, φ_3). So, we have to make a choice which variables we put in $\underline{q}_l$. We have two possibilities: we can choose $\underline{q}_l = [\varphi_2\ \varphi_3]^T$, or we can choose $\underline{q}_l = [\varphi_2\ \varphi_1]^T$.

If we choose $\underline{q}_l = [\varphi_2\ \varphi_3]^T$ and define the desired trajectories $\underline{q}_{ld}$, $\underline{q}_{ld}$ and $\underline{q}_{ld}$, it is possible with (3.7) to determine the reference trajectorie φ_{1r} and its time-derivatives:

$$\ddot{\varphi}_{1r} = -\frac{c_{31}}{m31}\dot{\varphi}_{1r} + \frac{k}{m_{31}}\varphi_{1r} - \frac{1}{m_{31}}(m_{32}\ddot{\varphi}_{2r} + m_{33}\ddot{\varphi}_{3r} + c_{32}\dot{\varphi}_{2r} + c_{33}\dot{\varphi}_{3r} + k\varphi_{3r} + k_3\dot{e}_{3r}) \qquad (3.10)$$

Equation (3.10) is in most cases an instable differential equation. Only if we choose a very small springconstant k (something like c_{31} or smaller), we will find a stable solution. This is not an useful result, so we have to try the other possibility $\underline{q}_l = [\varphi_2\ \varphi_1]^T$. After defining the desired trajectories and its time-derivatives we can determine the reference trajectory φ_{3r} and its time-derivatives:

$$\ddot{\varphi}_{3r} = -\left(\frac{c_{33}+k_3}{m_{33}}\right)\dot{\varphi}_{3r} - \frac{k}{m_{33}}\varphi_{3r} - \frac{1}{m_{33}}(m_{31}\ddot{\varphi}_{1r} + m_{32}\ddot{\varphi}_{2r} + c_{32}\dot{\varphi}_{2r} - k\varphi_{1r} - k_3\dot{\varphi}_3) \qquad (3.11)$$

Equation (3.11) is a stable differential equation. We can determine $\varphi_{3r}(t)$ and its time-derivatives, and with this variables it is posible to determine the inputs $u_s(t)$ and $u_e(t)$ with (3.6) and (3.8).

Now that we have found a way to use the C CTC strategy for controlling the xy table, we can test the control strategy and find the properties of the control strategy. In the next chapters I will describe how the testing is done and which results I have found.

4. STABILITY OF THE CLOSED LOOP SYSTEM

Stability is an extremely important factor for control design, especially for the kind of flexible robot systems considered in this report. Lyapunov's stability theorems make possible a method of synthesizing control laws which guarantee stability of the closed loop system (system and controller).
In the second stability approach of Lyapunov (Kok, 1991), the first step is the derivation of the equivalent error equations of the closed loop system. The equations which describes the closed loop error dynamics of the model of the xy table controlled by the C CTC law are: (3.6-3.2, 3.7-3.3, 3.8-3.4):

$$m_{22}\ddot{e}_{2r}+m_{23}\ddot{e}_{3r}+c_{23}\dot{e}_{3r}+c_{21}\dot{e}_{1r}+k_2\dot{e}_{2r} = 0 \tag{4.1}$$

$$m_{31}\ddot{e}_{1r}+m_{32}\ddot{e}_{2r}+m_{33}\ddot{e}_{3r}+c_{32}\dot{e}_{2r}+c_{33}\dot{e}_{3r}+c_{31}\dot{e}_{1r}-k(e_{1r}-e_{3r})+k_3\dot{e}_{3r} = 0 \tag{4.2}$$

$$m_{11}\ddot{e}_{1r}+m_{13}\ddot{e}_{3r}+c_{12}\dot{e}_{2r}+c_{13}\dot{e}_{3r}+c_{11}\dot{e}_{1r}+k(e_{1r}-e_{3r})+k_1\dot{e}_{1r} = 0 \tag{4.3}$$

$$i = 1,\ 2,\ 3$$
$$e_{ir} = \varphi_{ir}-\varphi_i$$

Then, we use the total reference error vector $\underline{e}_r$ in order to obtain a short notation of the equivalent error equations of the overall closed loop system:

$$M\underline{\ddot{e}}_r+C\underline{\dot{e}}_r+K_k\underline{e}_r+K_r\underline{\dot{e}}_r = 0 \tag{4.4}$$

In the second step, a positive definite Lyapunov function candidate V(t) of the total reference error vector $\underline{e}_r$ is chosen such that it represents the mechanical energy of the flexible system.

$$V = \frac{1}{2}\underline{\dot{e}}_r^{\,T}M\underline{\dot{e}}_r+\frac{1}{2}\underline{e}_r^{\,T}K_k\underline{e}_r \tag{4.5}$$

To guarantee the stability we have to come up the following 4 requirements:

1) $V(\underline{x},t) = 0$, with $\underline{x} = 0$

$\Rightarrow \underline{x} = [\underline{e}_r\ \underline{\dot{e}}_r] \rightarrow$ oké

2) $V(\underline{x},t) \geq \alpha \|\underline{x}\|$

$\Rightarrow$ The inertia matrix M is positive definite and K_k is not negative definite.
$\rightarrow$ oké

3) $V(\underline{x},t)$= continuous and differentiable.

$\rightarrow$ oké

4) $\dot{V}(\underline{x},t) < 0$

$\Rightarrow$ With (4.5) we can find:

$$\dot{V} = \underline{\dot{e}}_r^{\,T} M \underline{\ddot{e}}_r + \frac{1}{2}\underline{\dot{e}}_r^{\,T} \dot{M} \underline{\dot{e}}_r + \underline{e}_r^{\,T} K_k\, \underline{\dot{e}}_r \tag{4.6}$$

With (4.4) V can be given by:

$$\dot{V} = \underline{\dot{e}}_r^{\,T}\left(\frac{1}{2}\dot{M} - C\right)\underline{\dot{e}}_r - \underline{\dot{e}}_r^{\,T} K_r\, \underline{\dot{e}}_r \tag{4.7}$$

I have defined the matrix C (see appendix A) such that the matrix $[\frac{1}{2}\dot{M}\text{-}C]$ is skew-symmetric, i.e.:

$$\left[\frac{1}{2}\dot{M} - C\right] = -\left[\frac{1}{2}\dot{M} - C\right]^T \tag{4.8}$$

As a consequence, we can make use of the property of skew-symmetry of $[\frac{1}{2}\dot{M}\text{-}C]$ in that:

$$\underline{x}^T[\frac{1}{2}\dot{M}-C]\underline{x} = 0 \quad ,\textit{for any arbritrary } \underline{x} \tag{4.9}$$

So,

$$\dot{V} = -\underline{\dot{e}}_r{}^T K_r \, \underline{\dot{e}}_r < 0 \tag{4.10}$$

→ oké

The Lyapunov function comes up to the 4 properties. So we can conclude that the closed loop system is asymptotically stable.

Note:

Expression (4.10) shows that the total reference velocity error converges to the sliding surface $\underline{s} = \underline{\dot{e}}_r = \underline{\dot{e}} + \lambda \underline{e} = \underline{0}$, which implies that both the velocity and position tracking errors go to zero.

5. SIMULATIONS WITH THE THEORETICAL SITUATION

In this report I will make a distinction between a theoretical situation and a practical situation. In this chapter I will show and discuss the simulations I have executed with the theoretical situation.

Theoretical situation

The theoretical situation is the situation I was using in the previous chapters. This means that:

- all parameters are known.
- all variables + time derivatives are known (on every moment).
- the model of the xy table fits the reality.
- the inputs are continuous.

In chapter 4 I have proved that this theoretical situation leads to a asymptotically stable system.

Simulations

The mean goal of executing the simulations is to test the C CTC law and to determine the properties of the C CTC law. The simulations have been executed with the program MATLAB. The differential equations have been solved with a third order Runge Kutta integration algorithm. The integration accuracy can be chosen.
The goal of applying the C CTC law is to track the desired trajectory $\underline{q}_{ld} = [\varphi_{2d}\ \varphi_{1d}]^T$ and its time derivative. The C CTC law calculates φ_{3r} (+ time derivatives) and the inputs u_s and u_e so that $\underline{q}_l$, $\underline{q}_l$ will track $\underline{q}_{ld}$, $\underline{q}_{ld}$.
With the simulations I have to find the influence of:

- the control gain factors.
- the desired trajectories.
- the spring constant of the elastic transmission.
- the integration accuracy.

Further I have to find out:

- if the system is stable (if not something is wrong).
- how good and fast is the tracking of the desired trajectory.
- if the inputs are realistic (possible to create).

During the research I have executed a lot of simulations with different situations. In appendix B I have showed some results of these simulations.

Simulation results

In appendix B I have given an extensive review of the simulation results. The mean conclusion of that review is that the results are good and that the results confirm with the expectations.

6. SIMULATIONS WITH A PRACTICAL SITUATION

In this chapter I will show and discuss the simulations I have executed with some practical situations. I have executed this kind of simulations to form a picture of the properties of the system in reality (in practice).

Practical situation

In reality we have to do with the following situation:

- errors in the model of the xy table (unmodeled dynamics)
- wrong estimated parameters
- discontinuous inputs
- it will cost time to determine the inputs
- limited inputs
- measurement errors and measurement noise

The equations of motion in reality are (see also 3.2 to 3.4):

$$m_{22}\ddot{\varphi}_2+m_{23}\ddot{\varphi}_3+c_{23}\dot{\varphi}_3+c_{21}\dot{\varphi}_1+w_2 sign(\dot{\varphi}_2)+w_{22} = u_s \tag{6.1}$$

$$m_{31}\ddot{\varphi}_1+m_{32}\ddot{\varphi}_2+m_{33}\ddot{\varphi}_3+c_{32}\dot{\varphi}_2+c_{33}\dot{\varphi}_3+c_{31}\dot{\varphi}_1-k(\varphi_1-\varphi_3)+w_{33} = 0 \tag{6.2}$$

$$m_{11}\ddot{\varphi}_1+m_{13}\ddot{\varphi}_3+c_{12}\dot{\varphi}_2+c_{13}\dot{\varphi}_3+c_{11}\dot{\varphi}_1+w_1 sign(\dot{\varphi})_1+k(\varphi_1-\varphi_3)+w_{11} = u_e \tag{6.3}$$

w_{22}, w_{33}, w_{11} *represents the unmodeled dynamics*

In reality the C CTC law in partitioned form (see also 3.6 to 3.8) can be given by:

$$u_{sc} = m_{22e}\ddot{\varphi}_{2rm}+m_{23e}\ddot{\varphi}_{3rm}+c_{23e}\dot{\varphi}_{3rm}+c_{21e}\dot{\varphi}_{1rm}+w_{2e}sign(\dot{\varphi}_{2m})+k_2\dot{e}_{2rm} \tag{6.4}$$

$$0 = m_{31e}\ddot{\varphi}_{1rm}+m_{32e}\ddot{\varphi}_{2rm}+m_{33e}\ddot{\varphi}_{3rm}+c_{32e}\dot{\varphi}_{2rm}+c_{33e}\dot{\varphi}_{3rm}+c_{31e}\dot{\varphi}_{1rm}-k_e(\varphi_{1rm}-\varphi_{3rm})+k_3\dot{e}_{3rm} \tag{6.5}$$

$$u_{ec} = m_{11e}\ddot{\varphi}_{1rm}+m_{13e}\ddot{\varphi}_{3rm}+c_{12e}\dot{\varphi}_{2rm}+c_{13e}\dot{\varphi}_{3rm}+c_{11e}\dot{\varphi}_{1rm}+w_{1e}sign(\dot{\varphi}_{1m})+k_e(\varphi_{1rm}-\varphi_{3rm})+k_1\dot{e}_{1rm} \tag{6.6}$$

where,

$$i = 1,2,3 \quad j = 1,2,3 \quad k = 1,2$$

m_{ije}, c_{ije}, k_e , w_{ie} *are the estimated parameters* $(m_{ije} = m_{ij}+\Delta m_{ij}, ...)$

$$\dot{\varphi}_{krm} = \dot{\varphi}_{kd}+\lambda_k(\varphi_{kd}-\varphi_{km})$$

$$\dot{e}_{ir} = \dot{\varphi}_{irm}-\dot{\varphi}_{im}$$

$\varphi_{km} = \varphi_k+v_p$, $\dot{\varphi}_{im} = \dot{\varphi}_i+v_v$ *are the measurements*

v_p, v_v *are the measure errors*

u_{sc}, u_{ec} *are the discontinuous inputs*

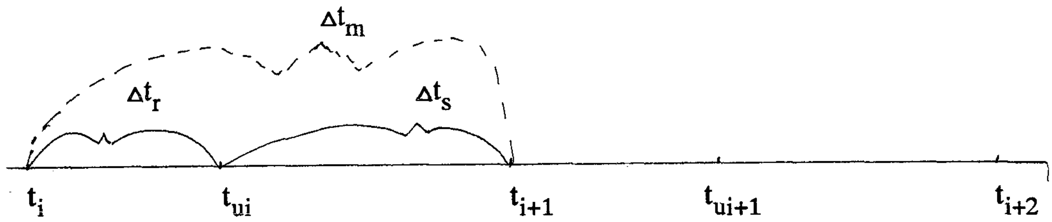
If we use the same proof of stability as in chapter 4, it is not possible to guarantee the stability of the (partial unknown) practical situation. So, the big question is : At which practical situations will the C CTC law be applicable. Or in other terms: What is the robustness of the C CTC controller.

By executing simulations with some practical situations, we have to form a picture of the robustness of the controller. With the simulation results we have to decide on which situations it is justified to apply the C CTC controller.

Simulations

We consider the next practical situation (= measure situation):

time axis:



Measure situation:

1) On t= t_i the measurements of φ_1 $\dot{\varphi}_1$, φ_2 $\dot{\varphi}_2$ and φ_3 $\dot{\varphi}_3$ are executed.
2) On t= t_{ui} the calculated inputs u_{sc} and u_{ec} are presented to the system. The calculation of the inputs costs Δt_r seconds.
3) On t= t_{i+1} the cyclus starts again.

Notes:
- the measure frequence = $1/(\Delta t_r+\Delta t_s) = 1/\Delta t_m$
- During an interval t_{ui+1}-t_{ui} the inputs have a constant value

Other aspects which plays an important role in the considered practical situation are:

- unmodeled dynamics
- wrong estimated parameters
- measure errors
- limited inputs

The C CTC law

The goal of applying the C CTC law in the practical situation is to track the desired trajectory $q_{ld} = [\varphi_{2d}, \varphi_{1d}]^T$ and its time derivative (the same as at the theoretical situation).

Simulation results

I have executed a lot of simulations with different situations. In appendix C I have showed some results of these simulations and I have given an extensive review of the simulation results.
The mean conclusion of that review is that the results are satisfying and that the results confirm with our expectations. There will be an reasonable change that the C CTC law applied on the xy table will answer to our desirements (stable system, small tracking errors).

7. FUTURE

In the future I will execute the following research:

1) Executing experiments

This is the most important part of the research. With the simulation results we have a good picture of the properties of the system in reality. With this knowledge we have to decide which situations we will consider during the experiments.
Before I can execute the experiments I have to make the designed C CTC law suitable for implementation in the program with which the controlling is executed.
An other problem which we have to solve is the handling of the inputs. It is possible to measure the angular position with an acceptable accuracy, but the measuring of the angular velocities have a too big inaccuracy. This problem can be solved by the implementation of an identification algorithm. With this identification algorithm (i.e. Kalman filter) it is possible to calculate the angular velocities out of the measurements of the angular positions and the known system behaviour.

2) Change the C CTC algorithm so that the x- and y position of the end effector of the xy table will track a desired trajectory.

With the designed C CTC algorithm it is possible to track $q_{ld} = [\varphi_{2d}\ \varphi_{1d}]^T$. The x- and y position of the end effector of the xy table are fixed by $\varphi_1, \varphi_2, \varphi_3$. So, with this C CTC algorithm the x- and y position of the end effector are not fixed.
The mean reason of controlling the xy table is that the end effector will track a certain desired trajectory. So, this have to be the final goal of applying the C CTC law. If I have time, I will try to design such a C CTC algorithm.

8. CONCLUSIONS

For the controlling of the xy table with one flexible transmission it is possible to design a control law ,which is based on the C CTC strategy. The goal of applying this C CTC controller is to track the desired trajectory $q_{ld}= [\varphi_{2d}\ \varphi_{1d}]^T$ and its time derivative.

It can be proved that applying of the C CTC law in a theoretical situation leads to a stable system behaviour.

The simulation results which are related with the theoretical situation confirm with the expectations, and give a good picture of the properties of the C CTC controller.

Applying of the C CTC law in a practical situation gives no guarantee of a stable system behaviour.

From the executed simulations it appears that there will be a reasonable change that the C CTC controller applied in a practical situation will answer to our desirements (stable system behaviour, small tracking errors).

Because of the satisfying simulation results, the next step of the research have to be the executing of some experiments.

APPENDICES

APPENDIX A: THE MODEL OF THE XY TABLE

In this appendix I will give a detailed desricption of the model of the xy table (v.d. Molengraft, 1989).

I have chosen for the next model of the xy table:

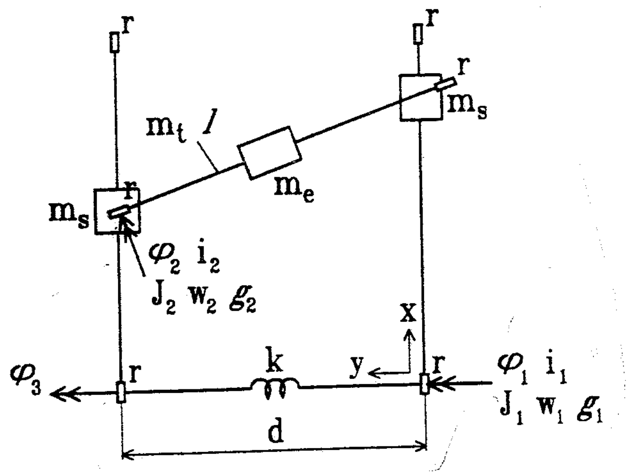


fig a1 model of the xy table

The equations of motion of the xy table can be written as:

$$M(\underline{q})\underline{\ddot{q}}+C(\underline{q},\underline{\dot{q}})\underline{\dot{q}}+K_k\underline{q}+\underline{n}(\underline{\dot{q}}) = \underline{u} \tag{A1.1}$$

where,

$$\underline{q} = \begin{bmatrix} \varphi_2 \\ \varphi_3 \\ \varphi_1 \end{bmatrix}; \qquad \underline{n} = \begin{bmatrix} n_{22} \\ 0 \\ n_{11} \end{bmatrix}; \qquad \underline{u} = \begin{bmatrix} u_s \\ 0 \\ u_e \end{bmatrix} \tag{A1.2}$$

$$M = \begin{bmatrix} m_{22} & m_{23} & 0 \\ m_{32} & m_{33} & m_{31} \\ 0 & m_{13} & m_{11} \end{bmatrix}; \qquad C = \begin{bmatrix} 0 & c_{23} & c_{21} \\ c_{32} & c_{33} & c_{31} \\ c_{12} & c_{13} & c_{11} \end{bmatrix}; \qquad K_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k & -k \\ 0 & -k & k \end{bmatrix} \tag{A1.3}$$

The dynamic model can be given in the following detailed form:

$$m_{22}\ddot{\varphi}_2+m_{23}\ddot{\varphi}_3+c_{23}\dot{\varphi}_3+c_{21}\dot{\varphi}_1+n_{22} = u_s \tag{A1.4}$$

$$m_{31}\ddot{\varphi}_1+m_{32}\ddot{\varphi}_2+m_{33}\ddot{\varphi}_3+c_{32}\dot{\varphi}_2+c_{33}\dot{\varphi}_3+c_{31}\dot{\varphi}_1-k(\varphi_1-\varphi_3) = 0 \tag{A1.5}$$

$$m_{11}\ddot{\varphi}_1+m_{13}\ddot{\varphi}_3+c_{12}\dot{\varphi}_2+c_{13}\dot{\varphi}_3+c_{11}\dot{\varphi}_1+n_{11}+k(\varphi_1-\varphi_3) = u_e \tag{A1.6}$$

where,

$$m_{11} = J_1+(m_s+\frac{1}{3}m_t(\frac{l}{d})^2+m_e(\frac{\varphi_2 r}{d})^2)r^2$$

$$m_{22} = J_2+m_e r^2$$

$$m_{33} = (m_s+m_t-m_t(\frac{l}{d})+\frac{1}{3}m_t(\frac{l}{d})^2+m_e-2m_e(\frac{\varphi_2 r}{d})+m_e(\frac{\varphi_2 r}{d})^2)r^2$$

$$m_{13} = m_{31} = (\frac{1}{2}m_t(\frac{l}{d})-\frac{1}{3}m_t(\frac{l}{d})^2+m_e(\frac{\varphi_2 r}{d})-m_e(\frac{\varphi_2 r}{d})^2)r^2$$

$$m_{23} = m_{32} = (m_e\frac{(\varphi_1-\varphi_3)r}{d})r^2$$

$$c_{11} = m_e\varphi_2\frac{r^4}{d^2}\dot{\varphi}_2$$

$$c_{33} = -m_e\frac{r^3}{d}\dot{\varphi}_2+m_e\varphi_2\frac{r^4}{d^2}\dot{\varphi}_2$$

$$c_{12} = m_e\varphi_2\frac{r^4}{d^2}(\dot{\varphi}_1-\dot{\varphi}_3)$$

$$c_{21} = -m_e\varphi_2\frac{r^4}{d^2}(\dot{\varphi}_1-\dot{\varphi}_3)$$

$$c_{13} = -m_e\frac{r^4}{d^2}\varphi_2\dot{\varphi}_2$$

$$c_{31} = m_e\frac{r^3}{d}\dot{\varphi}_2-m_e\varphi_2\frac{r^4}{d^2}\dot{\varphi}_2$$

$$c_{23} = m_e \varphi_2 \frac{r^4}{d^2}(\dot{\varphi}_1 - \dot{\varphi}_3)$$

$$c_{32} = m_e \frac{r^3}{d}(\dot{\varphi}_1 - \dot{\varphi}_3) - m_e \varphi_2 \frac{r^4}{d^2}(\dot{\varphi}_1 - \dot{\varphi}_3)$$

$$n_{11} = w_1 sign(\dot{\varphi}_1)$$

$$n_{22} = w_2 sign(\dot{\varphi}_2)$$

The following parameters have been determined by local identification.

m_s= 2.3 kg, m_t= 8.5 kg, m_e= 2.3 kg, l= 1.25 m, d= 1 m, r= 0.01 m

The following parameters have been determined by some identification algorithmen (v.d Molengraft, 1989).

J_1= 2.15 10^{-3} kgm^2, J_2= 1.45 10^{-4} kgm^2, w1= 0.47, w2= 0.15

APPENDIX B: SIMULATION RESULTS OF SOME THEORETICAL SITUATIONS

In this appendix I will show and discuss some simulation results. These results are the most important results of the simulations I have executed with some theoretical situations.

Theoretical situations

I have considered the next theoretical situations:

1) - desired trajectory: $\varphi_{2d}= \alpha_2+\alpha_2\cos(\omega_2 t)$; $\dot{\varphi}_{2d}= -\alpha_2\omega_2\sin(\omega_2 t)$; ..
$\varphi_{1d}= \alpha_1+\alpha_1\cos(\omega_1 t)$; $\dot{\varphi}_{1d}= -\alpha_1\omega_1\sin(\omega_1 t)$; ..
α_2= 25 rad, ω_2= 10 rad/s and α_1= 25 rad, ω_1= 10 rad/s
see 'plot 0' for these trajectories.

- spring constant: k= 1 Nm/rad

- control gain parameters: k_1= 0.1, k_2= 0.01, k_3= 0.01, λ_1= 5 and λ_2= 5

- integration accuracy: tol= 0.01

- tracking error t=0: $\Delta\varphi_2(0)= 0$, $\Delta\varphi_1(0)= 0$, $\Delta\varphi_3(0)= 0$
- velocity error t=0: $\Delta\dot{\varphi}_2(0)= 0$, $\Delta\dot{\varphi}1(0)= 0$, $\Delta\dot{\varphi}_3(0)= 0$

2). As situation 1 but now:

- integration accuracy: tol= 0.001

3). As situation 1 but now:

- tracking error t=0: $\Delta\varphi_2(0)$= -10 rad, $\Delta\varphi_1(0)$= -10 rad, $\Delta\varphi_3(0)$= -10 rad

4). As situation 3 but now:

- control gain parameters: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 , λ_2= 10

5). As situation 1 but now:

- velocity error t=0: $\Delta\dot{\varphi}_2(0)$= 100 rad/s, $\Delta\dot{\varphi}_1(0)$= 100 rad/s, $\Delta\dot{\varphi}_3(0)$= 100 rad/s

6). As situation 1 but now:

- spring constant: k= 0.1 Nm/rad.

The simulation results

Of every situation I have made the following plots:

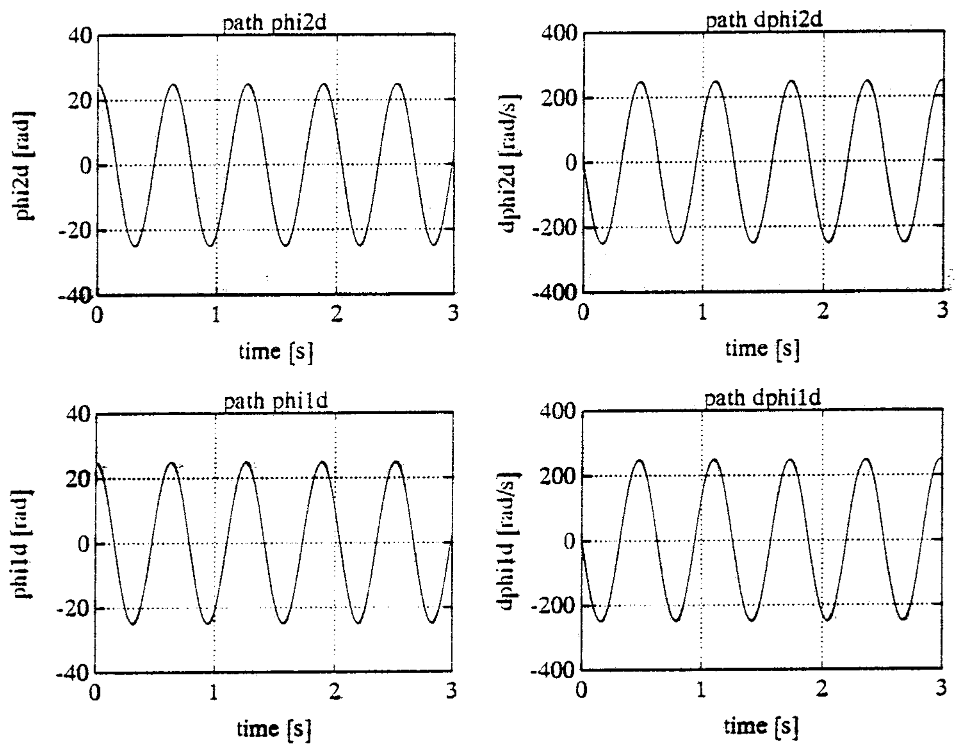
1	tracking error	$\varphi_{2d}-\varphi_2$
2	tracking error	$\varphi_{1d}-\varphi_1$
3	velocity error	$\dot{\varphi}_{2d}-\dot{\varphi}_2$
4	velocity error	$\dot{\varphi}_{1d}-\dot{\varphi}1$
5	$\varphi_{3r}-\alpha_1$ and $\varphi_3-\alpha_1$	
6	velocity error	$\dot{\varphi}3r-\dot{\varphi}_3$
7	elasticity	$\varphi_3-\varphi_1$
8	inputs	u_s and u_e

The structure of a page with plots is:

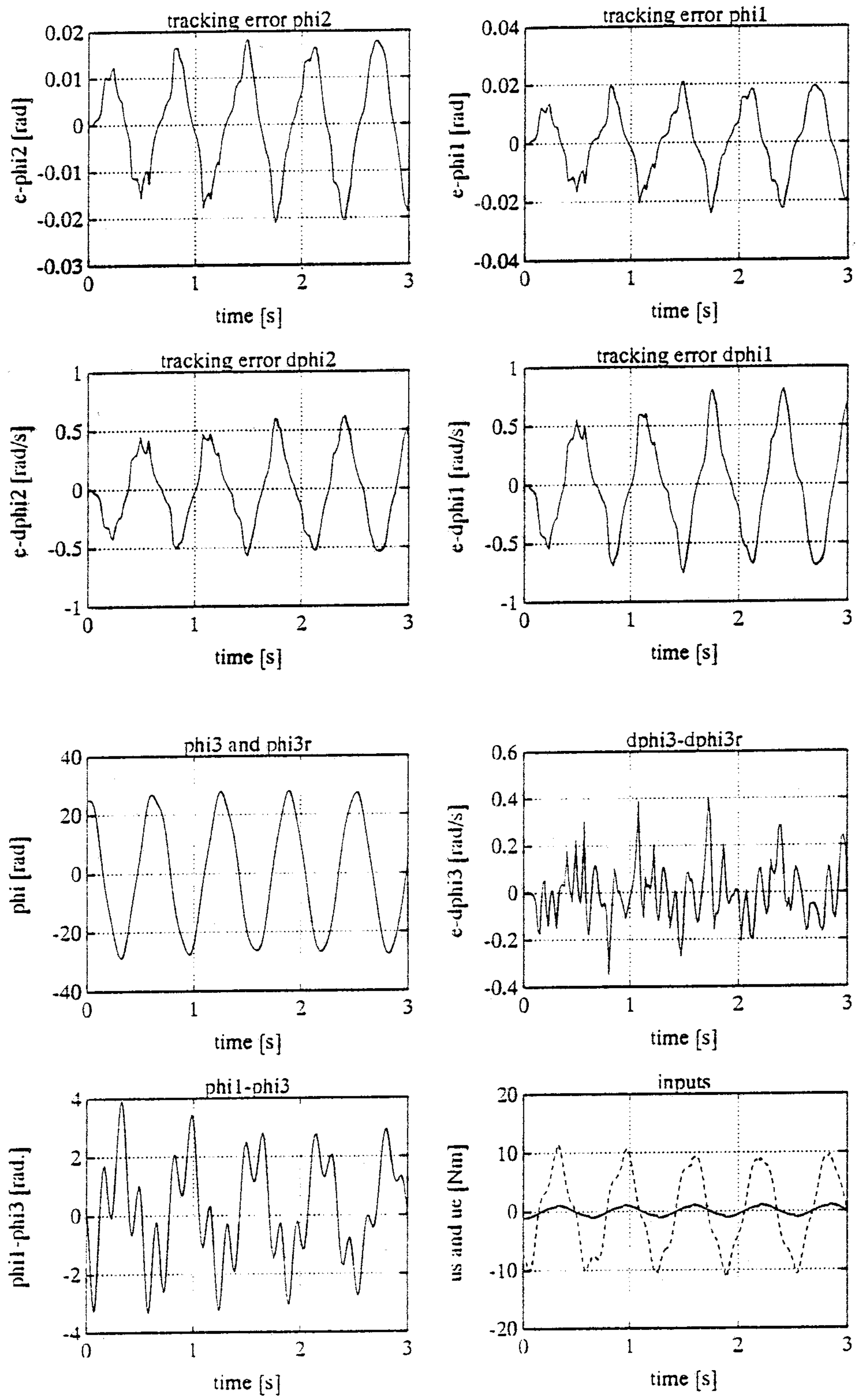
$$\begin{bmatrix} (1) & (2) \\ (3) & (4) \\ (5) & (6) \\ (7) & (8) \end{bmatrix}$$

Now I will give per situation the simulation results (see the following pages).

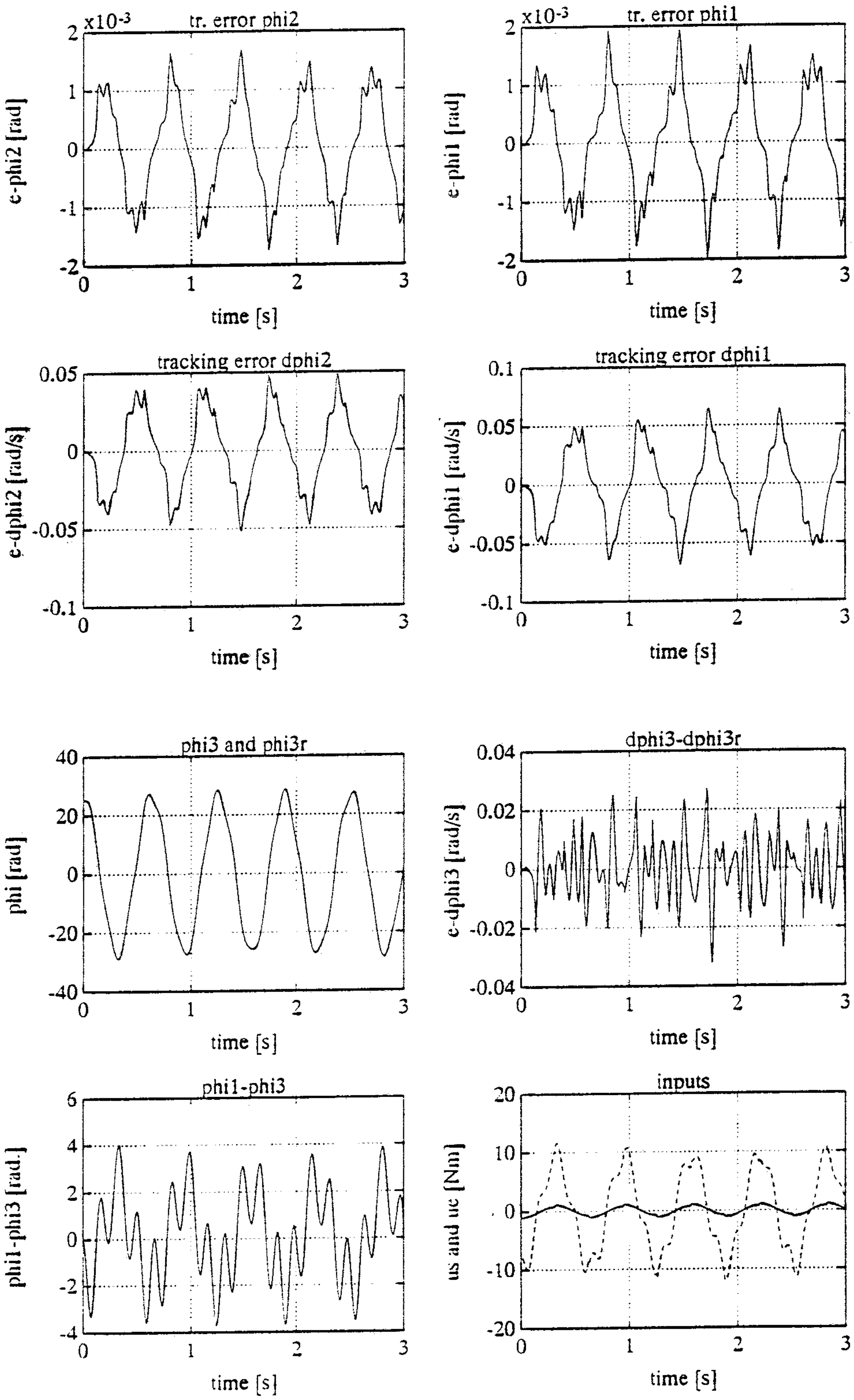
Desired trajectorie: plot 0



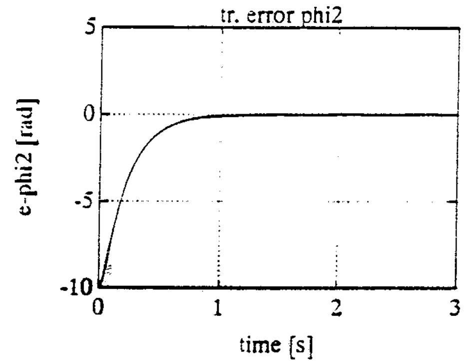
situation 1

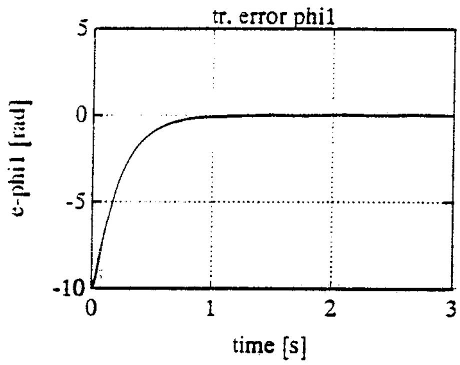

tracking error phi2
e-phi2 [rad]
time [s]
tracking error phi1
e-phi1 [rad]
time [s]
tracking error dphi2
e-dphi2 [rad/s]
time [s]
tracking error dphi1
e-dphi1 [rad/s]
time [s]
phi3 and phi3r
phi [rad]
time [s]
dphi3-dphi3r
e-dphi3 [rad/s]
time [s]
phi1-phi3
phi1-phi3 [rad.]
time [s]
inputs
us and uc [Nm]
time [s]

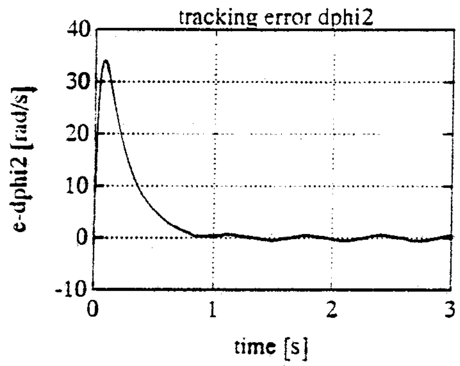
situation 2

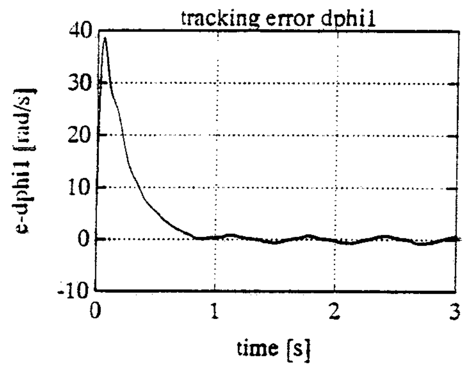

x10-3
tr. error phi2
e-phi2 [rad]
time [s]
x10-3
tr. error phi1
e-phi1 [rad]
time [s]
tracking error dphi2
e-dphi2 [rad/s]
time [s]
tracking error dphi1
e-dphi1 [rad/s]
time [s]
phi3 and phi3r
phi [rad]
time [s]
dphi3-dphi3r
e-dphi3 [rad/s]
time [s]
phi1-phi3
phi1-phi3 [rad]
time [s]
inputs
us and uc [Nm]
time [s]

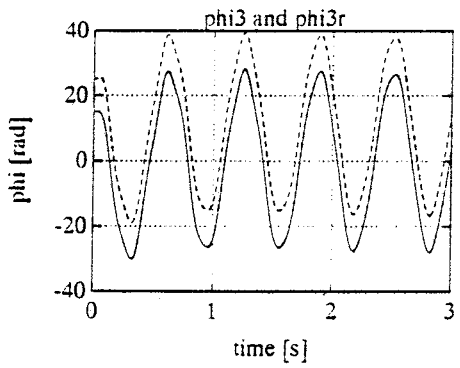
situation 3

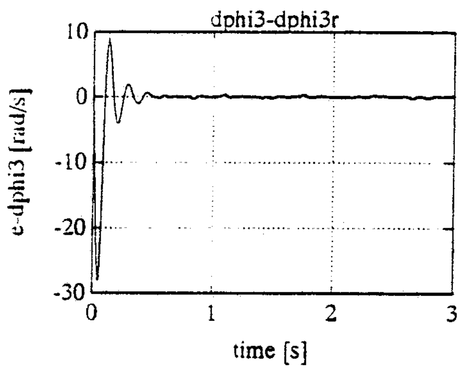

tr. error phi2
e-phi2 [rad]
time [s]

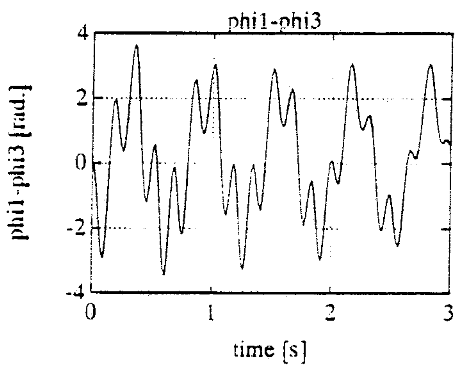

tr. error phi1
e-phi1 [rad]
time [s]

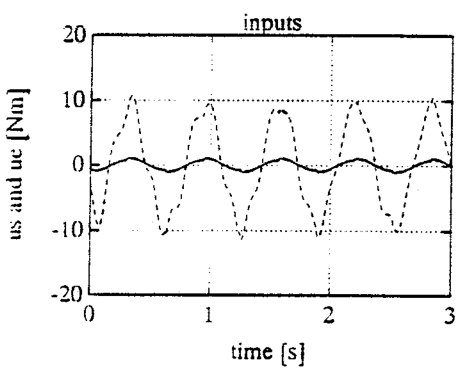

tracking error dphi2
e-dphi2 [rad/s]
time [s]


tracking error dphi1
e-dphi1 [rad/s]
time [s]

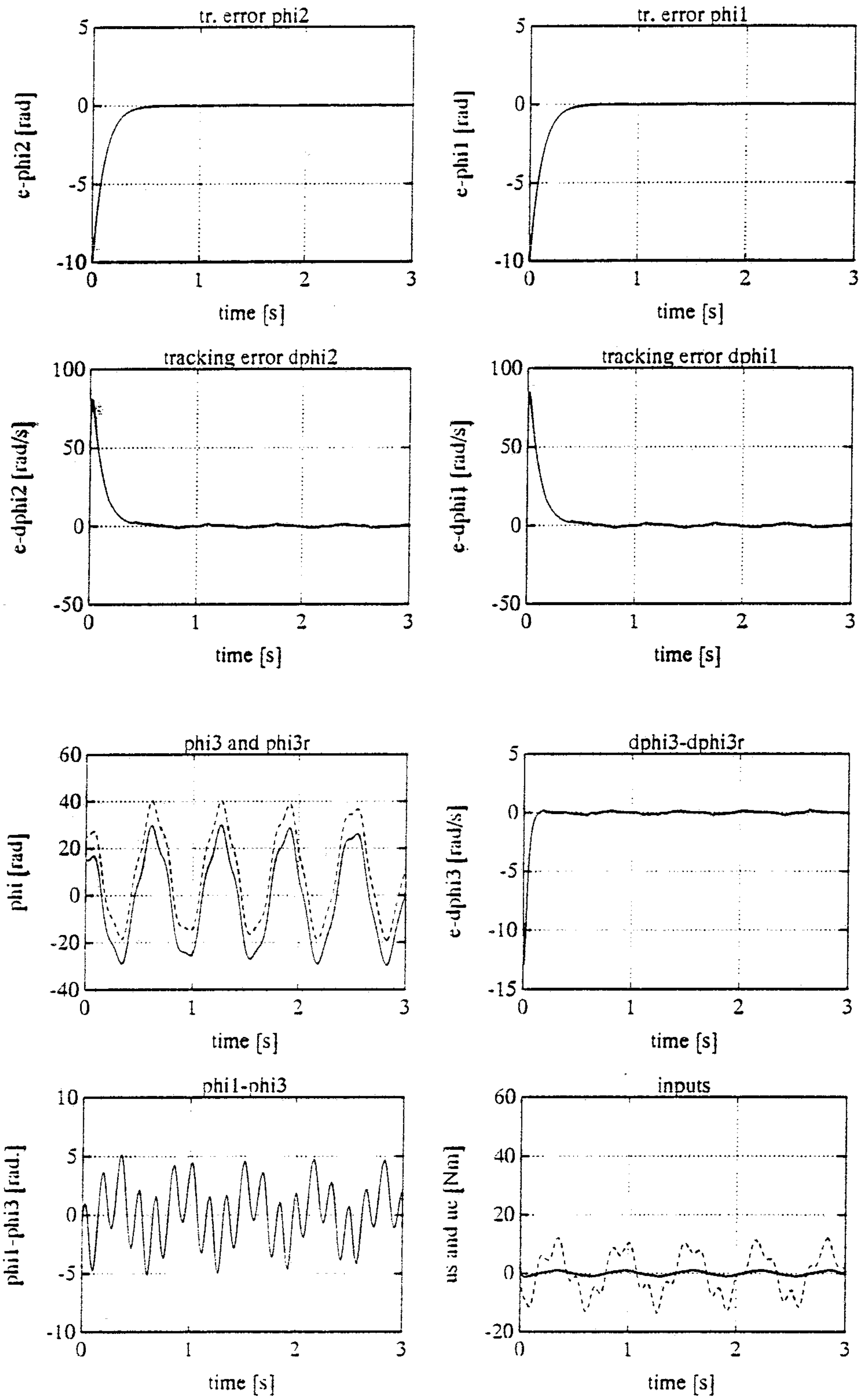

phi3 and phi3r
phi [rad]
time [s]


dphi3-dphi3r
e-dphi3 [rad/s]
time [s]

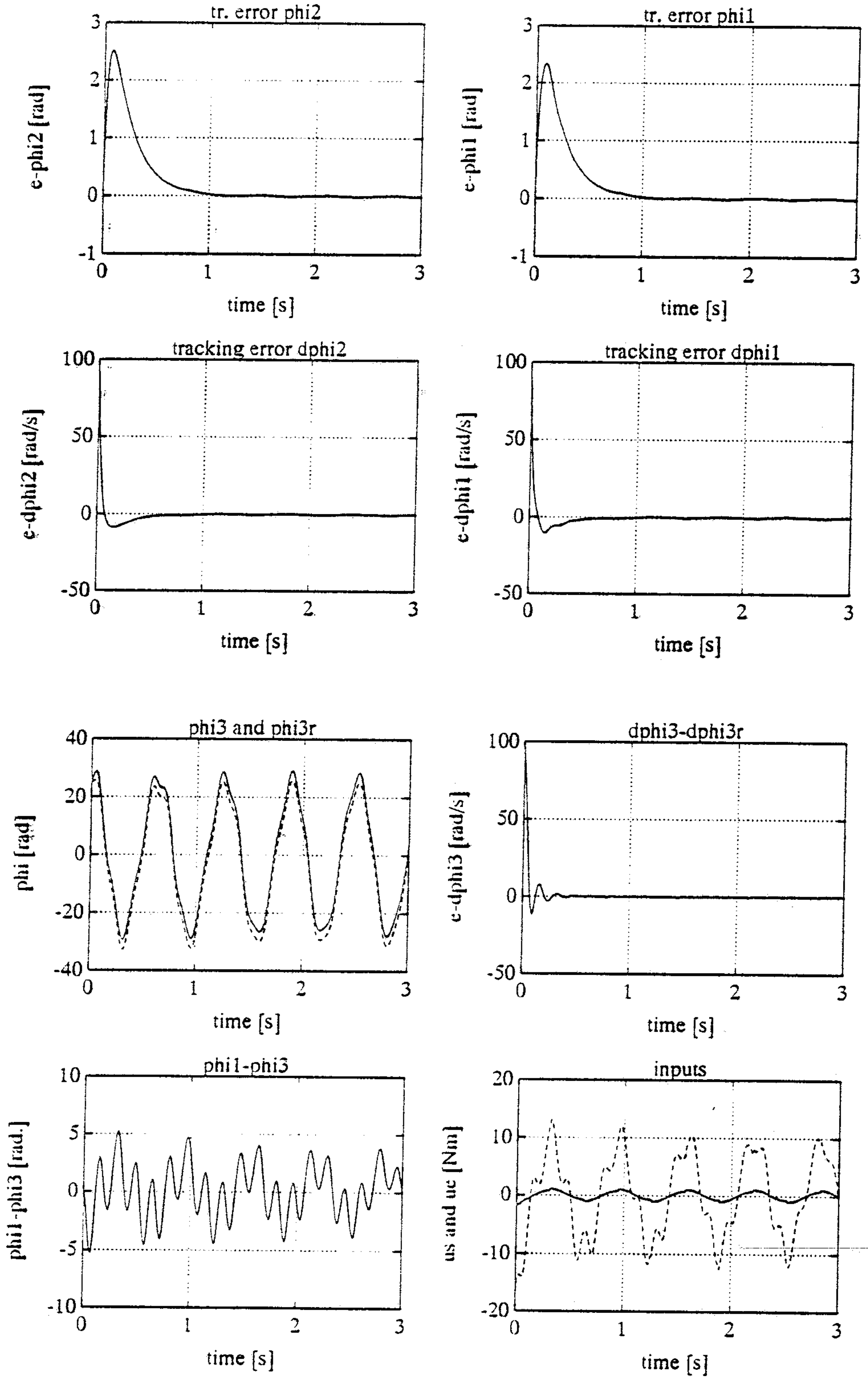

phi1-phi3
phi1-phi3 [rad.]
time [s]


inputs
us and ue [Nm]
time [s]

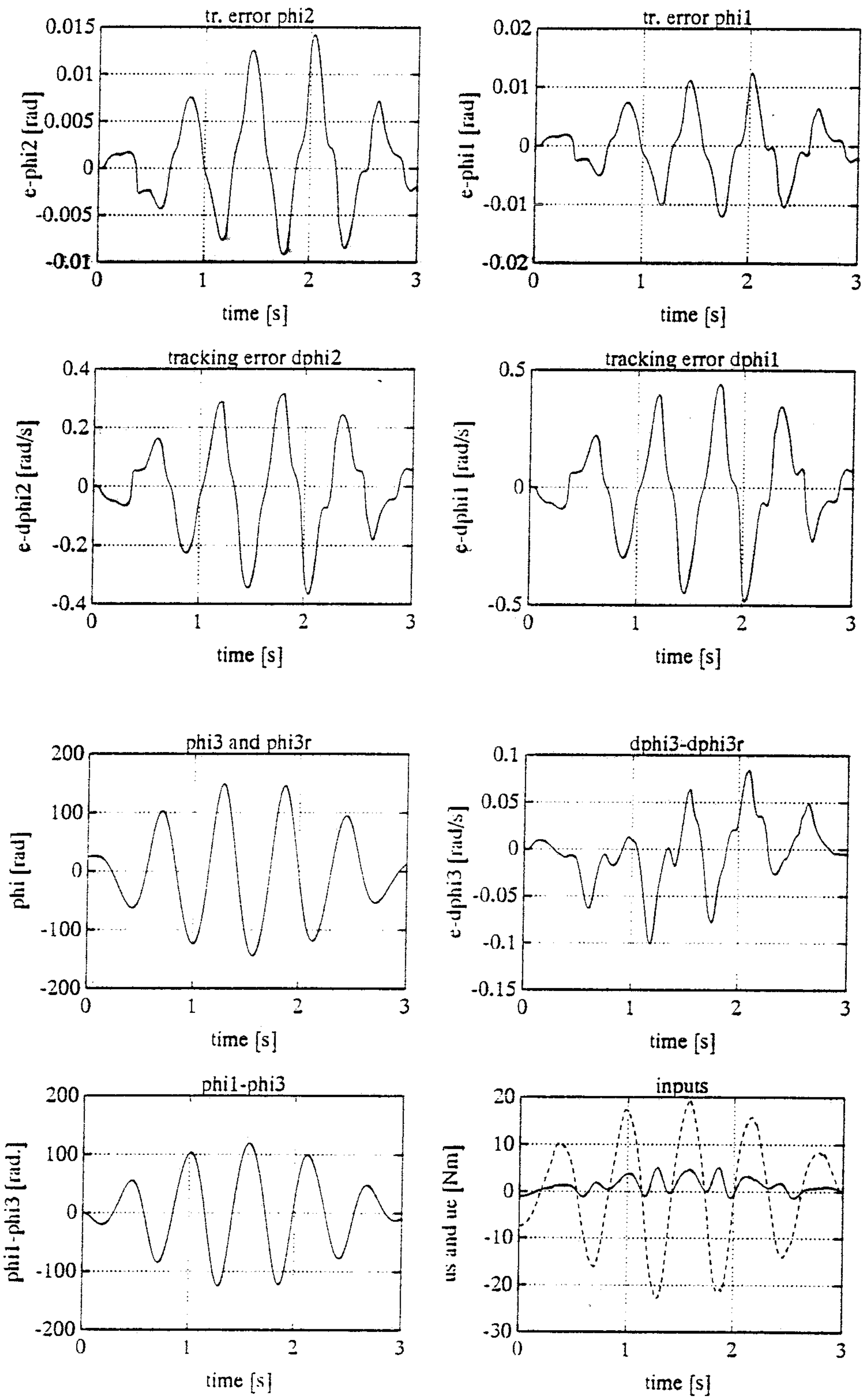
situation 4


tr. error phi2
e-phi2 [rad]
time [s]
tr. error phi1
e-phi1 [rad]
time [s]
tracking error dphi2
e-dphi2 [rad/s]
time [s]
tracking error dphi1
e-dphi1 [rad/s]
time [s]
phi3 and phi3r
phi [rad]
time [s]
dphi3-dphi3r
e-dphi3 [rad/s]
time [s]
phi1-phi3
phi1-phi3 [rad.]
time [s]
inputs
us and uc [Nm]
time [s]

situation 5


tr. error phi2
e-phi2 [rad]
time [s]
tr. error phi1
e-phi1 [rad]
time [s]
tracking error dphi2
e-dphi2 [rad/s]
time [s]
tracking error dphi1
e-dphi1 [rad/s]
time [s]
phi3 and phi3r
phi [rad]
time [s]
dphi3-dphi3r
e-dphi3 [rad/s]
time [s]
phi1-phi3
phi1-phi3 [rad.]
time [s]
inputs
us and uc [Nm]
time [s]

situation 6


tr. error phi2
e-phi2 [rad]
time [s]
tr. error phi1
e-phi1 [rad]
time [s]
tracking error dphi2
e-dphi2 [rad/s]
time [s]
tracking error dphi1
e-dphi1 [rad/s]
time [s]
phi3 and phi3r
phi [rad]
time [s]
dphi3-dphi3r
e-dphi3 [rad/s]
time [s]
phi1-phi3
phi1-phi3 [rad.]
time [s]
inputs
us and ue [Nm]
time [s]

Discussing the simulation results

situation 1

It appears that the tracking errors φ_{2d}-φ_2, φ_{1d}-φ_1 and the velocity errors $\dot{\varphi}_{2d}$-$\dot{\varphi}_2$, $\dot{\varphi}_{1d}$-$\dot{\varphi}_1$ are not equal to zero, and that these errors will not converge to zero.
According to the theory (proof of stability) the errors have to converge to zero. However, we may not forget that the simulations are an approach of reality. In the case of situation 1 the integration accuracy is equal to 0.01. This means that the integration algorithm will adapt the time step Δt so that the integration error will not be bigger as 0.01.
So, in the case of the simulations we have to do with a discontinuous situation. and we have to do with integration errors. That is the reason why the errors will not converge to zero (see also situation 2).

situation 2

By reducing the integration accuracy with a factor 10 its appears that the errors will also reduce with a factor 10. We can conclude that the integration algorithm with a certain accuracy will introduce some small tracking- and velocity errors. These errors will reduce by reducing the integration accuracy (practical disadvantage computing time will increase).

situation 3

It appears that the tracking errors are controlled to zero (in fact to the same value as in situation 1). It also appears that the tracking error φ_{3r}-φ_3 will not converge to zero, and that the velocity error $\dot{\varphi}_{3r}$-$\dot{\varphi}_3$ will converge to zero. It is possible to explain this aspect.

With the proof of stability I have proved that:

$$\lim_{t\to\infty} [\dot{e}_r] = 0 \Rightarrow$$

$$\begin{aligned} \dot{e}_{1r} &= (\dot{\varphi}_{1d}-\dot{\varphi}_1)+\lambda_1(\varphi_{1d}-\varphi_1) \\ \dot{e}_{2r} &= (\dot{\varphi}_{2d}-\dot{\varphi}_2)+\lambda_2(\varphi_{2d}-\varphi_2) \\ \dot{e}_{3r} &= \dot{\varphi}_{3r}-\dot{\varphi}_3 \end{aligned} \quad \text{(A2.2)}$$

It appears that:

$$t \to \infty \qquad \dot{\varphi}_{3r}-\dot{\varphi}_3 = 0 \qquad \text{(A2.3)}$$

In the case of φ_{3r} there is no relation between $\dot{\varphi}_{3r}$ and φ_3, so φ_{3r}-φ_3 will not converge to zero:

$$t \to \infty \qquad \varphi_{3r}-\varphi_3 \neq 0 \qquad \text{(A2.4)}$$

situation 4

It appears by increasing the control gain parameters, that the tracking and velocity errors will converge faster to zero. This is in accordance to our expectations.

situation 5

It appears that the velocity errors are controlled to zero.

situation 6

It appears that if flexibility plays an important role, that the tracking- and velocity errors will stay small. It also appears that φ_1-φ_3 gets an unrealistic big value.

Note:

I have executed some more simulations with other springconstants, desired trajectories, control gain parameters, integration accuracies and start errors. The results of these simulations are in accordance to the expectations, and the results don't lead to new insights.

APPENDIX C: SIMULATION RESULTS OF SOME PRACTICAL SITUATIONS

In this appendix I will show and discuss some simulation results. These results are the most important results of the simulations I have executed with some practical situations.

Practical situations

I have considered the next practical situations:

1) - desired trajectory: $\varphi_{2d}= \alpha_2+\alpha_2\sin(\omega_2 t)$, ...
$\varphi_{1d}= \alpha_1+\alpha_1\cos(\omega 3t)$, ...
α_2= 25 rad, ω_2= 1 rad/s and α_1= 25 rad, ω_1= 1 rad/s

- spring constant: k= 1 Nm/rad

- control gain parameters: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10

- integration accuracy: can not be chosen

- discretization: Δt_r= 0.001 s, Δt_s= 0.005 s (Δt_m= 0.006 s)

- particularies: none (no start errors, no wrong parameter estimations, no unmodeled dynamics, etc.)

2) As situation 1 but now:

- ω_2= 10 rad/s, ω_1= 10 rad/s.

3) As situation 2 but now:

- discretization: Δt_r= 0.002 s, Δt_s= 0.008 s (Δt_m= 0.01 s)

4) As situation 2 but now:

- discretization: Δt_r= 0.005 s, Δt_s= 0.005 s (Δt_m= 0.01 s)

5) As situation 2 but now:

- discretization: $\Delta t_r = 0.005$ s, $\Delta t_s = 0$ s ($\Delta t_m = 0.005$ s)

6) As situation 2 but now:

- discretization: $\Delta t_r = 0.005$ s, $\Delta t_s = 0.003$ s ($\Delta t_m = 0.008$ s)

7) As situation 2 but now:

- tracking errors t= 0: $\Delta\varphi_1 = -10$ rad, $\Delta\varphi_3 = -10$ rad

8) As situation 2 but now:

- tracking errors t= 0: $\Delta\varphi_1 = -25$ rad, $\Delta\varphi_3 = -25$ rad
- velocity error t= 0: $\Delta\dot{\varphi}_2 = -250$ rad/s

9) As situation 2 but now:

- wrong estimated parameters: reality: w1= 0.47 Nm, w2= 0.15 Nm; model: w1e= 0.3 Nm, w2e= 0.1 Nm

10) As situation 2 but now:

- unmodeled dynamics: reality: w1= 0.47 Nm, w2= 0.15 Nm; model: w1e= 0 Nm, w2e= 0 Nm

11) As situation 3 but now:

- spring constant: k= 0.1 Nm/rad

12) As situation 3 but now:

- measure errors:
  - angular position → white noise $v_p = a_{vp}\text{rand}(t)$ → $a_{vp} = 0.1$
  - angular velocity → white noise $v_v = a_{vv}\text{rand}(t)$ → $a_{vv} = 1$

  note: rand(t)= white noise, between -1 and 1.

Simulation results

I have made the same plots of every situation as of the theoretical situation (see appendix B). The structure of a page with plots is the same as given in apendix B.
Now I will give per situation the simulation results (see the following pages).

opm simulatie resultaten zie rap g2 052

Discussion of the simulation results

situation 1

If the angular velocities ω_1 and ω_2 have a small value, the controller will control the tracking- and velocity errors to zero.

situation 2

If we chose a bigger value of ω_1 and ω_2 the tracking- and velocity errors will increase (= logical). However, the system stays stable. It appears that the velocity error $\dot{\varphi}_{3r}$-$\dot{\varphi}_3$ have a big value. Further are the peaks in the velocity errors $\dot{\varphi}_{2r}$-$\dot{\varphi}_2$ and $\dot{\varphi}_{1r}$-$\dot{\varphi}_1$ strange. I don't have a clear explanation of this phenomenom. It could be possible that numerical aspects will cause this peaks.
Further, it appears that φ_1-φ_3 has a maximum of 4 rad. In reality the slide of the xy table will get stuck if this difference will grow too big. The question is at which difference this phenomenom will taken place. The answer of this question is not exactly known.

situation 3 to situation 6

By variation of Δt_r and Δt_s the tracking- and velocity errors will change in accordance to the expectations. If we chose Δt_r and/or Δt_s too big, we will get a instable system behaviour.

situation 7 and situation 8

It appears that the start errors are controlled to zero. The tracking error φ_{3r}-φ_3 will not converge to zero (the same as at the theoretical situation).
It also appears that at the beginning of the simulations, the inputs get big values. In practice this can give problems, because the inputs are limited.

situation 9 and situation 10

If the model with which we calculate the inputs (6.3 to 6.6) exists some wrong estimated parameters, it appears that the tracking- and velocity errors are bigger. But it appears also that (at the considered situations) the system behaviour stays stable !

situation 11

By decreasing the spring constant it appears that the tracking- and velocity errors are bigger but reasonable (instability !?). It also appears that φ_1-φ_3 is very big. In practice this is not possible. So, it is not recommended to do experiments with this situation.

situation 12

It appears that the presence of measure noise will cause a capricious expiration in time of the tracking- and velocity errors and of the inputs. In the case of situation 12 the tracking- and velocity errors don't have a bigger value as in the case without measurement noise. However, if we chose bigger values of a_{vp} and a_{vv} (see 6.6 to 6.8) the errors will increase.

Note:

I have executed some more simulations with other situations. The simulation results are in accordance to the expectations. The simulation results don't lead to new insights.

The mean conclusion of the discussion of the simulation results is that if we don't chose a too extreme situation, that the results are satisfying. So, there will be a reasonable change that the C CTC controller is suitable for the controlling of the xy table.