Composite computed torque control of the xy table with an elastic motor transmission theoretical analysis, simulations and experiments

Brevoord, Gert

Award date:
1992

part 1

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

theoretical analysis, simulations and experiments

by: Gert Brevoord

Professor: prof. dr. ir. J.J. Kok
Coach: ir. I.M.M. Lammerts

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

Report no.: 92.054

Recommendations

To continue the research in future the following aspects can be investigated.

1) Application of the C CTC strategy to other systems

The properties of the C CTC law that I have found during this research are related to the xy table. The question is: how will the properties of the C CTC strategy be if the control strategy is used for the controlling of other systems.
In future, the C CTC law has to be used for controlling other systems. The elasticity of those systems has to play a (very) important role. Then it is possible to make more general conclusions.

2) Implementation of an adaptation algorithm if system parameters are partially unknown.

One of the big disadvantages of the control strategy is that we have to know all the system parameters. In practice this can cause problems.
In future, it will be nice if an adaptation algorithm is implemented in the C CTC strategy. In practice this implementation can also lead to a reduction of the tracking - and velocity errors.
Note: Ricky Doelman (student of the WFW control group) is designing this mentioned adaptation algorithm for a flexible TR robot.

3) Comparison with other control strategies

During the research I have compared the results of the C CTC law with the general CTC law. It can be useful to make a comparison with other control strategies.

4) Improvement of some global investigated aspects:

- observer algorithm
- output control
- designing of a routine to chose the control - and observer parameters

ABSTRACT

This report gives a summary of the final 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 composed out of two Computed Torque Control routines, 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 strategy in a practical situation. The flexible xy table, which is situated in the WFW lab, will be used as the test apparatus. The xy table has 3 degrees of freedom (φ_1, φ_2, φ_3) and 2 control inputs. The flexibility can be desrcribed by φ_1-φ_3.

I have split up the research into two parts. At the first part of the research I focus my attention to get a picture of the properties of the C CTC strategy. I have designed a control law based on the C CTC strategy for controlling a (simplificated) model of the xy table. The goal of applying the designed C CTC law is to track the desired trajectories φ_{1d} and φ_{2d} and to control the flexibility φ_1-φ_3. This control law is a state control routine. By executing theoretical analysis and simulations with theoretical and practical situations (a situation where unmoddeled dynamics, wrong estimated parameters, measure noise, etc. plays an important role) I have formed a picture of the properties of the C CTC controller.

For theoretical situations it appears that the stability is guaranteed, and that the simulation results are good. For a practical situation it appears that the simulation results are satisfactory and that the robustness of the controller is reasonable. There will be a considerable chance that the C CTC controller will be applicable in practice.

At the second part of the research I focus my attention to the specific controlling of the experimental xy table. Now, we have to account for the limitations of the system. During this part of the research I have solved problems as: the redesigning of the C CTC law and the implementation of this control law in the control system, the handling of the measurements (observer design), the designing of a simulator to test designed control laws and the total organisation of executing experiments.

I have designed a set of programs with which it is possible to execute simulations and experiments. During the execution of simulations and experiments I have determined a suitable control system configuration and I have determined suitable control and observer parameters for different situations. Futher I have investigated the robustness of the control system, and I have made a comparison between the control results of the C CTC controller and a CTC controller.

It appears that the designed C CTC controller answers to our desirements. The system behaviour stays stable and the tracking and velocity errors converge to zero (to small values). The robustness of the system is reasonable and the control results of the C CTC controller are better than the control results of the considered CTC controller.

The final goal of applying a C CTC controller is that the end-effector will track a desired path. The end-effector of the xy table is fixed by φ_1, φ_2 and φ_3. So, with the designed C CTC state control routine it is not possible to track a desired end-effector path. To achieve this, the C CTC state control routine has to be changed in an output control routine. During the last part of the research I have designed such an output control routine. The control results of the by trial and error designed C CTC output control routine are reasonable.

CONTENTS

NOTATION

a,α,a_s	scalars
$\underline{a}$	vector
A	matrix
$\underline{a}^T$, A^T	transposition
A^{-1}	inversion
$A(i,j)$	element of matrix A on row i, column j
$f(t)$	argument of function f between round brackets
$\dot{a},\dot{\alpha},\dot{a}_s$	first order time derivative of a,α and a_s
$\ddot{a},\ddot{\alpha},\ddot{a}_s$	second order time derivative of a,α and a_s

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 Computed Torque Control 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 final 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 controlling the xy table with a flexible transmission (the test-apparatus of the research). Further I will show and discuss how the C CTC law is implemented in the control system of the xy table. By showing and discussing theoretical analysis, simulation results and experimental results I will give an extensive review of the properties of the C CTC controller. 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 report, 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.

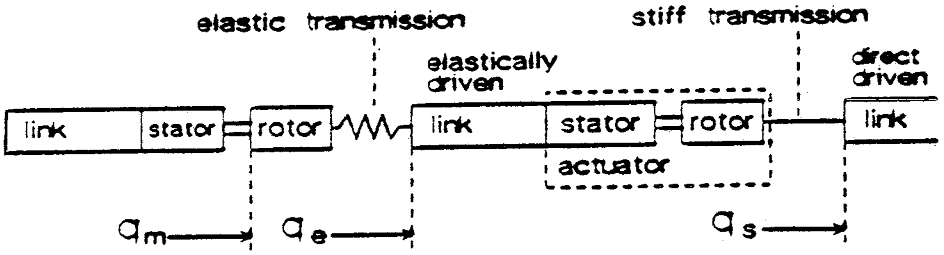


Fig 2.1 Elastically driven links - direct driven links.

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:

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

where:

$\underline{q} = [\ \underline{q}_s\ \underline{q}_e\ \underline{q}_m\ ]^T$
M = positive definite inertia matrix.
$C\underline{\dot{q}}$ = 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.
$H\underline{u}$ = is the vector with the inputs.

The model can be given in a detailed form by:

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

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

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

Composite computed 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 $\underline{q}_{mr(t)}$) 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).

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

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).
- $\underline{e}_r= \dot{\underline{e}}+\Lambda\underline{e} \quad \forall t\geq t_0; \quad \underline{e}_r(t_0)= \underline{e}(t_0)$.
- $\underline{e}= \underline{q}_d-\underline{q}$ 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:

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

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

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

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 $\ddot{\underline{q}}_{mr}(t)$. Now it is possible to determine the inputs $u_s(t)$ and $u_e(t)$ with (2.9) and (2.11).

One part of the controller deals with the tracking of a desired trajectory. This part of the controller is in fact a CTC controller. This CTC controller is described by (2.9) and (2.11). The question is: how can we find a reference trajectory of $q_m \rightarrow q_{mr}$.
The second part of the controller deals with the controlling of elastic vibrations. With (2.10) and q_{er} it is possible to calculate a reference manifold $z_{er}= K(q_{mr}-q_{er})$. The second part of the controller stabilizes the elastic vibrations around the specified reference manifold. This part of the controller is in fact also a CTC controller.
So, to control a system with flexible transmissions we use a control law which is composed out of two CTC laws. This is the reason why the control law is called the Composite Computed Torque Control law.

3. CONTROLLING THE XY TABLE WITH A COMPOSITE COMPUTED TORQUE CONTROL 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 controlling 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).

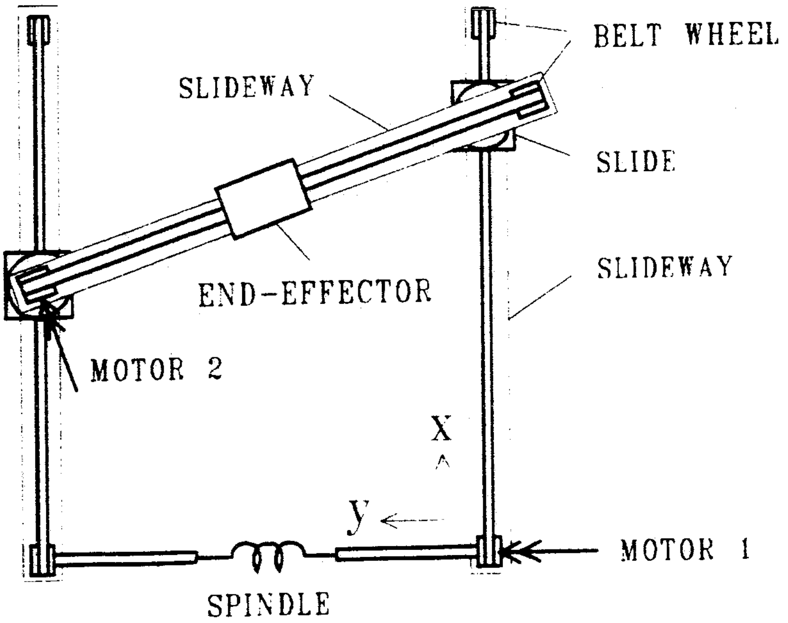


fig 3.1 Schematic representation of the xy table.

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

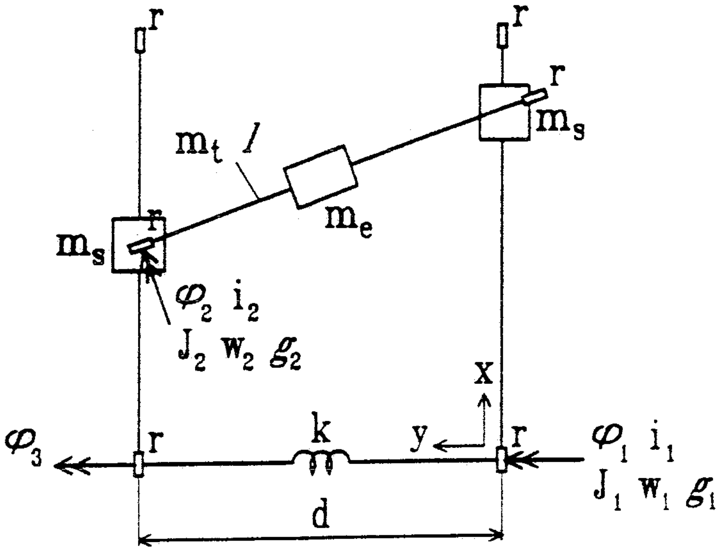


fig 3.2 model of the xy table

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

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

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

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:

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

Splitting the above computed torque expression into 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,

$$\begin{gathered} i = 1,\ 2 \\ \dot{\varphi}_{ir} = \dot{\varphi}_{id}+\lambda_i(\varphi_{id}-\varphi_i) \\ \varphi_{ir} = \textit{reference trajectory} \\ \varphi_{id} = \textit{desired trajectory} \\ j = 1,\ 2,\ 3 \\ e_{jr} = \varphi_{jr}-\varphi_j \end{gathered} \tag{3.9}$$

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 $\underline{q}_{ld}$, $\dot{\underline{q}}_{ld}$ and $\ddot{\underline{q}}_{ld}$, it is possible with (3.7) to determine the reference trajectorie φ_{1r} and its time-derivatives:

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

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:

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

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).

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):

$$m_{22}\ddot{e}_{2r}+m_{23}\ddot{e}_{3r}+c_{23}\dot{e}_{3r}+c_{21}\dot{e}_{1r}+k_{2}\dot{e}_{2r} = 0 \tag{3.12}$$

$$m_{31}\ddot{e}_{1r}+m_{32}\ddot{e}_{2r}+m_{33}\ddot{e}_{3r}+c_{32}\dot{e}_{2r}+c_{33}\dot{e}_{3r}+c_{31}\dot{e}_{1r}-k(e_{1r}-e_{3r})+k_{3}\dot{e}_{3r} = 0 \tag{3.13}$$

$$m_{11}\ddot{e}_{1r}+m_{13}\ddot{e}_{3r}+c_{12}\dot{e}_{2r}+c_{13}\dot{e}_{3r}+c_{11}\dot{e}_{1r}+k(e_{1r}-e_{3r})+k_{1}\dot{e}_{1r} = 0 \tag{3.14}$$

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

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:

$$M\underline{\ddot{e}}_r+C\underline{\dot{e}}_r+K_k\underline{e}_r+K_r\underline{\dot{e}}_r = 0 \tag{3.15}$$

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.

$$V = \frac{1}{2}\underline{\dot{e}}_r^{\,T}M\underline{\dot{e}}_r+\frac{1}{2}\underline{e}_r^{\,T}K_k\underline{e}_r \tag{3.16}$$

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

⇒ The inertia matrix M is positive definite and K_k is not negative definite.
→ oké

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

→ oké

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

⇒ With (3.16) we can find:

$$\dot{V} = \underline{\dot{e}}_r^T M \underline{\ddot{e}}_r + \frac{1}{2}\underline{\dot{e}}_r^T \dot{M} \underline{\dot{e}}_r + \underline{e}_r^T K_k \underline{\dot{e}}_r \tag{3.17}$$

With (3.15) $\dot{V}$ can be given by:

$$\dot{V} = \underline{\dot{e}}_r^T (\frac{1}{2}\dot{M} - C)\underline{\dot{e}}_r - \underline{\dot{e}}_r^T K_r \underline{\dot{e}}_r \tag{3.18}$$

I have defined the matrix C (see appendix A) such that the matrix [½$\dot{M}$-C] is skew-symmetric, i.e.:

$$[\frac{1}{2}\dot{M} - C] = -[\frac{1}{2}\dot{M} - C]^T \tag{3.19}$$

As a consequence, we can make use of the property of skew-symmetry of [½$\dot{M}$-C] in that:

$$\underline{x}^T[\frac{1}{2}\dot{M} - C]\underline{x} = 0 \quad , \textit{for any arbritrary } \underline{x} \tag{3.20}$$

So,

$$\dot{V} = -\underline{\dot{e}}_r^{\,T} K_r \underline{\dot{e}}_r < 0 \tag{3.21}$$

→ oké, if K_r is chosen positive definite

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

Note:

Expression (3.21) 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.

The research

Now we have found a suitable way to control the xy table with a flexible transmission. The designed C CTC law (3.6 3.8 3.11) guarantees stability of the closed loop system. It is possible to track $\underline{q}_{1d} = [\varphi_{2d}, \varphi_{1d}]^T$ and to stabilize the elastic vibrations. In fact this is a state control routine ($\underline{q}_1$ → $\underline{q}_{1d}$).
The x - and y position of the end-effector of the xy table are fixed by φ_1, φ_2 and φ_3 (see fig. 3.2). So, with the state control routine the x - and y position of the end-effector are not fixed. To control the end-effector, the C CTC law has to be changed in an output controller ([x,y] → $[x_d, y_d]$).
The mean goal of the research is to test the C CTC strategy and to find the properties of the C CTC strategy, and in first instance not to control the end-effector. For this research I use the designed state control routine. At the end of the research I have try to find a way of changing the C CTC law to control the end-effector (output control).
In the next chapters I will describe how the testing of the C CTC routines is executed, and which results I have found.

4. PROPERTIES OF THE C CTC STRATEGY

I have split the research into two parts. At the first part of the research I focus my attention to get a picture of the properties of the C CTC strategy. At the second part I focus my attention to the specific controlling of the experimental xy table. The first part is described in this chapter, and the second part is desrcribed in chapter 5 and chapter 6.
To get a picture of the properties of the C CTC strategy, I have executed a lot of simulations. I have made a distinction between theoretical - and practical situations.

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 3 is proven that this theoretical situation leads to an asymptotically stable system behaviour.

Simulations

The simulations have been executed with the program MATLAB. The differential equations have been solved with a third order Runge Kutta integration algorithm (variable stepsize). The integration accuracy can be chosen.
The goal of applying the C CTC law is to track the desired trajectory $\underline{q}_{1d}= [\varphi_{2d}\ \varphi_{1d}]^T$ and its time derivative. The C CTC law calculates φ_{3r} (+ time derivatives) and the inputs u_s and u_e so that $\underline{q}_1$, $\dot{\underline{q}}_1$ will track $\underline{q}_{1d}$, $\dot{\underline{q}}_{1d}$ (see chapter 3).
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).

Simulation results theoretical situations

In this paragraph I will show and discuss the most important simulation results.

I have chosen the following desired trajectory:

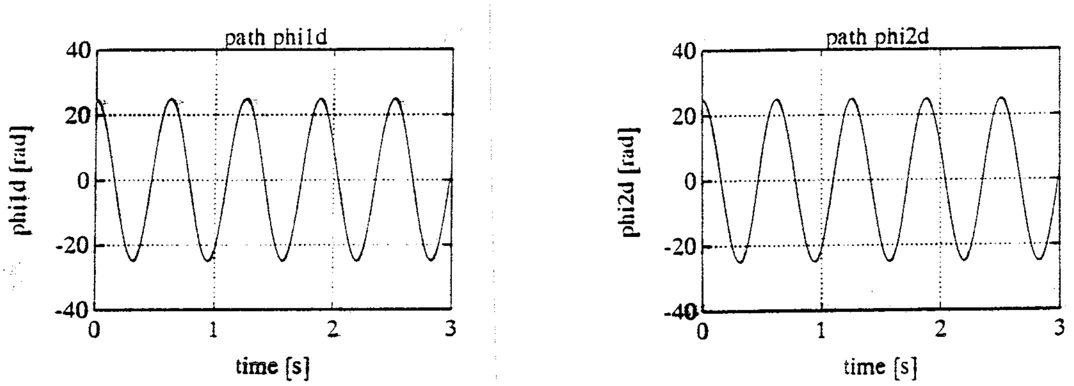
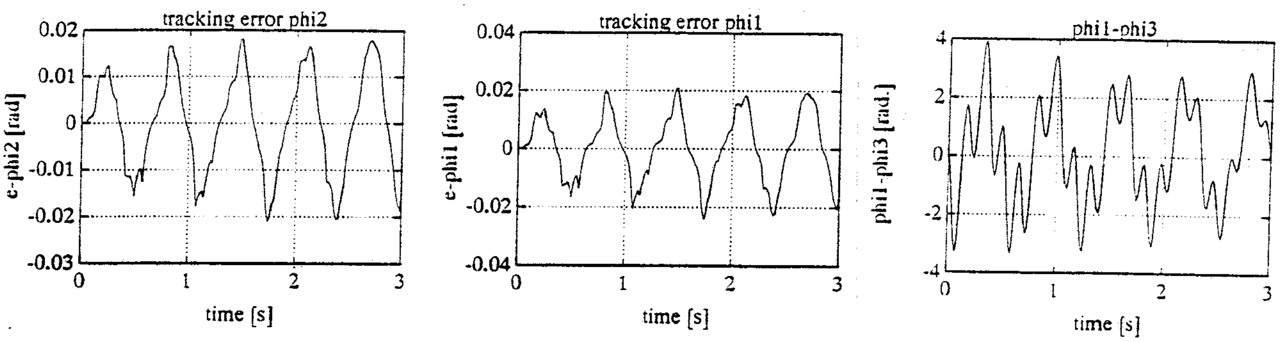


fig 4.1 desired trajectory

I define the following standard situation:

- desired trajectory: fig 4.1
- spring constant: k= 1 Nm/rad
- control gain parameters: k_1= 0.1, k_2= k_3= 0.01, λ_1= 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: $\Delta\varphi_2(0)= 0$, $\Delta\varphi 1(0)= 0$, $\Delta\varphi_3(0)= 0$

We get the following simulation results:



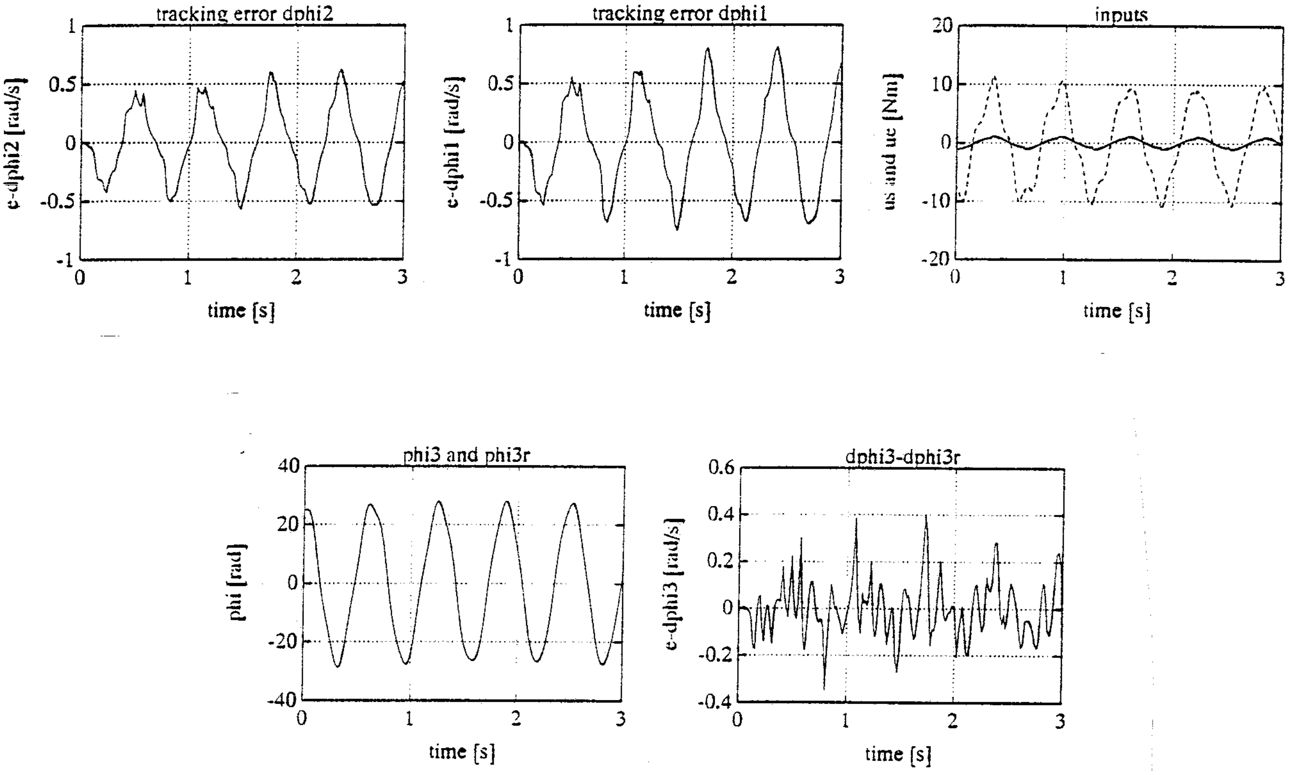


fig 4.2 simulation results standard situation

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 this situation 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 next situation).

Futher it appears that the elasticity φ_1-φ_3 is controlled, and that the inputs are realistic.

Now I will change one or some aspects of the presented standard situation. I will consider the influence of the changed aspects.

If we change the integration accuracy:

- integration accuracy: tol= 0.001

we get the following results:

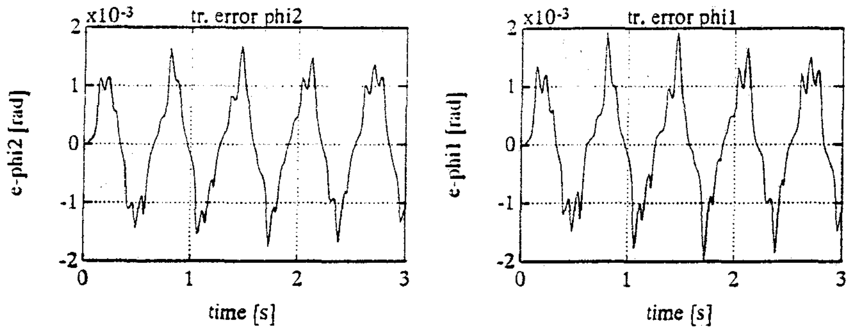
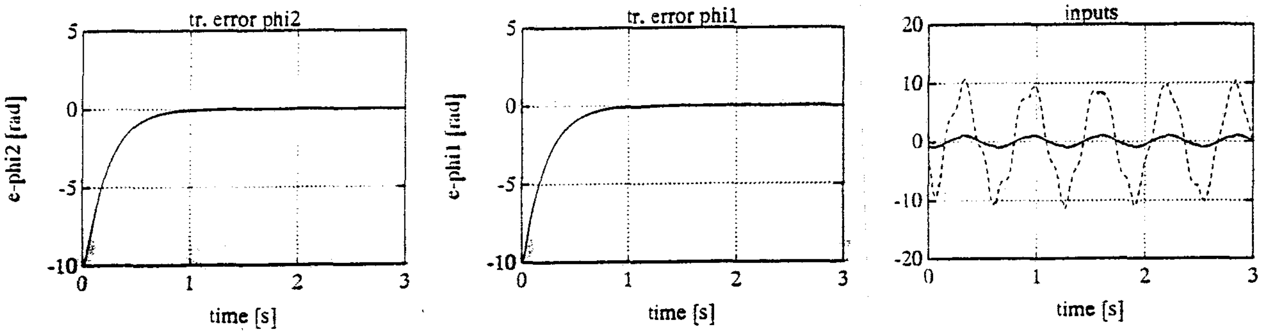


fig 4.3 influence of integration accuracy

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).

If we create a tracking error at t=0 we get:

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



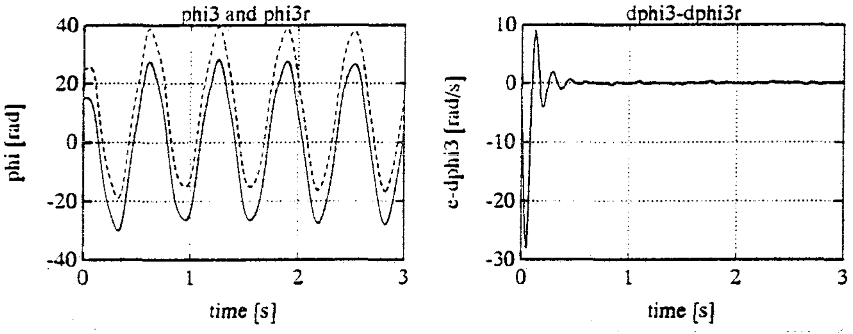


fig 4.4 influence of start errors

It appears that the tracking errors are controlled to zero (in fact to the same value as the standard situation). 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:

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

$$\begin{aligned} \dot{e}_{1r} &= (\dot{\varphi}_{1d}-\dot{\varphi}_1)+\lambda_1(\varphi_{1d}-\varphi_1) \\ \dot{e}_{2r} &= (\dot{\varphi}_{2d}-\dot{\varphi}_2)+\lambda_2(\varphi_{2d}-\varphi_2) \\ \dot{e}_{3r} &= \dot{\varphi}_{3r}-\dot{\varphi}_3 \end{aligned} \tag{4.1}$$

It appears that:

$$t \to \infty \qquad \dot{\varphi}_{3r}-\dot{\varphi}_3 = 0 \tag{4.2}$$

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

$$t \to \infty \qquad \varphi_{3r}-\varphi_3 \neq 0 \tag{4.3}$$

If we change the control gain parameters and create the start errors as mentioned before we get:

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

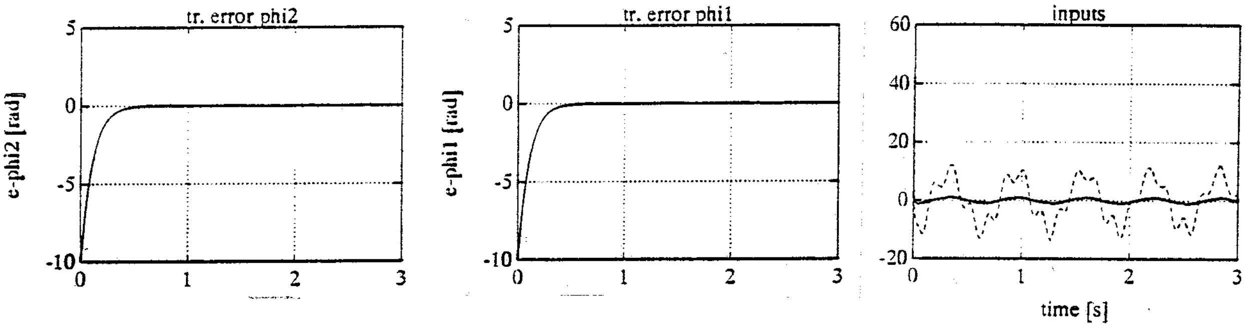


fig 4.5 influence of the control gain parameters

To guarantee the stability we have to chose k_1, k_2 and k_3 positive (see 3.21). The control gain parameters k_1, k_2, k_3 and λ_1, λ_2 determine the poles of $\dot{V}$. If we choose bigger values of k_1, k_2, k_3 and λ_1, λ_2 the negative real values of the poles of $\dot{V}$ will increase. This leads to a faster convergion of the tracking errors to zero. It also leads to bigger inputs at the beginning of the simulation (hardly to see).

If we change the springconstant we get:

- spring constant: k= 0.1 Nm/rad.

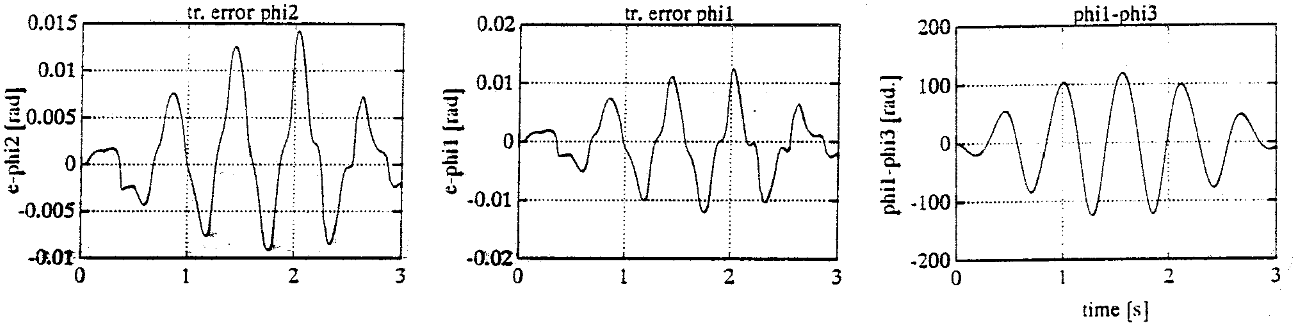


fig 4.6 influence of the spring constant k

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.

I have executed the same set of simulations with other desired trajectories. The results of the simulations confirm with the expectations → stable system behaviour and tracking errors converge to zero. These results don't lead to new insights.
In appendix B I have shown some more simulation results of theoretical situations.

Practical situation

In reality we have to do with the following situation:

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

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

$$m_{22}\ddot{\varphi}_2+m_{23}\ddot{\varphi}_3+c_{23}\dot{\varphi}_3+c_{21}\dot{\varphi}_1+w_2 sign(\dot{\varphi}_2)+w_{22} = u_s \tag{4.4}$$

$$m_{31}\ddot{\varphi}_1+m_{32}\ddot{\varphi}_2+m_{33}\ddot{\varphi}_3+c_{32}\dot{\varphi}_2+c_{33}\dot{\varphi}_3+c_{31}\dot{\varphi}_1-k(\varphi_1-\varphi_3)+w_{33} = 0 \tag{4.5}$$

$$m_{11}\ddot{\varphi}_1+m_{13}\ddot{\varphi}_3+c_{12}\dot{\varphi}_2+c_{13}\dot{\varphi}_3+c_{11}\dot{\varphi}_1+w_1 sign(\dot{\varphi}_1)+k(\varphi_1-\varphi_3)+w_{11} = u_e \tag{4.6}$$

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

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

$$u_{sc} = m_{22e}\ddot{\varphi}_{2rm}+m_{23e}\ddot{\varphi}_{3rm}+c_{23e}\dot{\varphi}_{3rm}+c_{21e}\dot{\varphi}_{1rm}+w_{2e}sign(\dot{\varphi}_{2m})+k_2\dot{e}_{2rm} \tag{4.7}$$

$$0 = m_{31e}\ddot{\varphi}_{1rm}+m_{32e}\ddot{\varphi}_{2rm}+m_{33e}\ddot{\varphi}_{3rm}+c_{32e}\dot{\varphi}_{2rm}+c_{33e}\dot{\varphi}_{3rm}+c_{31e}\dot{\varphi}_{1rm}-k_e(\varphi_{1rm}-\varphi_{3rm})+k_3\dot{e}_{3rm} \tag{4.8}$$

$$u_{ec} = m_{11e}\ddot{\varphi}_{1rm}+m_{13e}\ddot{\varphi}_{3rm}+c_{12e}\dot{\varphi}_{2rm}+c_{13e}\dot{\varphi}_{3rm}+c_{11e}\dot{\varphi}_{1rm}+w_{1e}sign(\dot{\varphi}_{1m})+k_e(\varphi_{1rm}-\varphi_{3rm})+k_1\dot{e}_{1rm} \tag{4.9}$$

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*

To guarantee the stability we have to come up to V < 0 (see 3.21). In practice it follows:

$$\dot{V} = -\underline{\dot{e}}_r K_r \underline{\dot{e}}_r + \eta \tag{4.10}$$

η → due to unmodelled dynamics, wrong estimated parameters, discontinuous inputs, time delay to calculate the inputs, measure noise and measure inaccuracy.

From equation (4.10) it appears that the system is stable if K_r is chosen in such a way that:

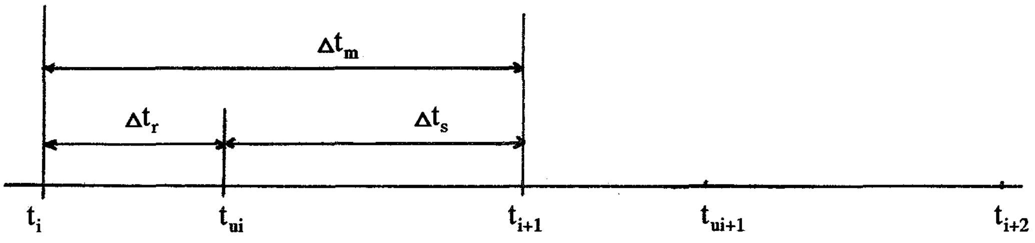
$$\underline{\dot{e}}_r K_r \underline{\dot{e}}_r > \eta$$

In practice η is not (exactly) known. So, we don't have an idea how we have to choose K_r. An other problem is that in practice we can't choose K_r too big because of the fact that the inputs u_s and u_e will grow to unrealistic values (in practice the inputs are bounded).

Measure situation

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

Simulation results of some practical situations

The goal of this part of the research is to get a global picture of the robustness against unmodelled dynamics, wrong estimated parameters, discontinuous inputs, time delay to calculate the inputs, measure noise and measure inaccuracy of the C CTC strategy.
In this paragraph I will show and discuss the most important simulation results.

Compared with the standard situation (page 14) I have changed the next aspects:

- 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

1- discretization: $\Delta t_r = 0.001$ s, $\Delta t_s = 0.005$ s ($\Delta t_m = 0.006$ s)
2- discretization: $\Delta t_r = 0.002$ s, $\Delta t_s = 0.008$ s ($\Delta t_m = 0.01$ s)

This leads to the following simulation results:

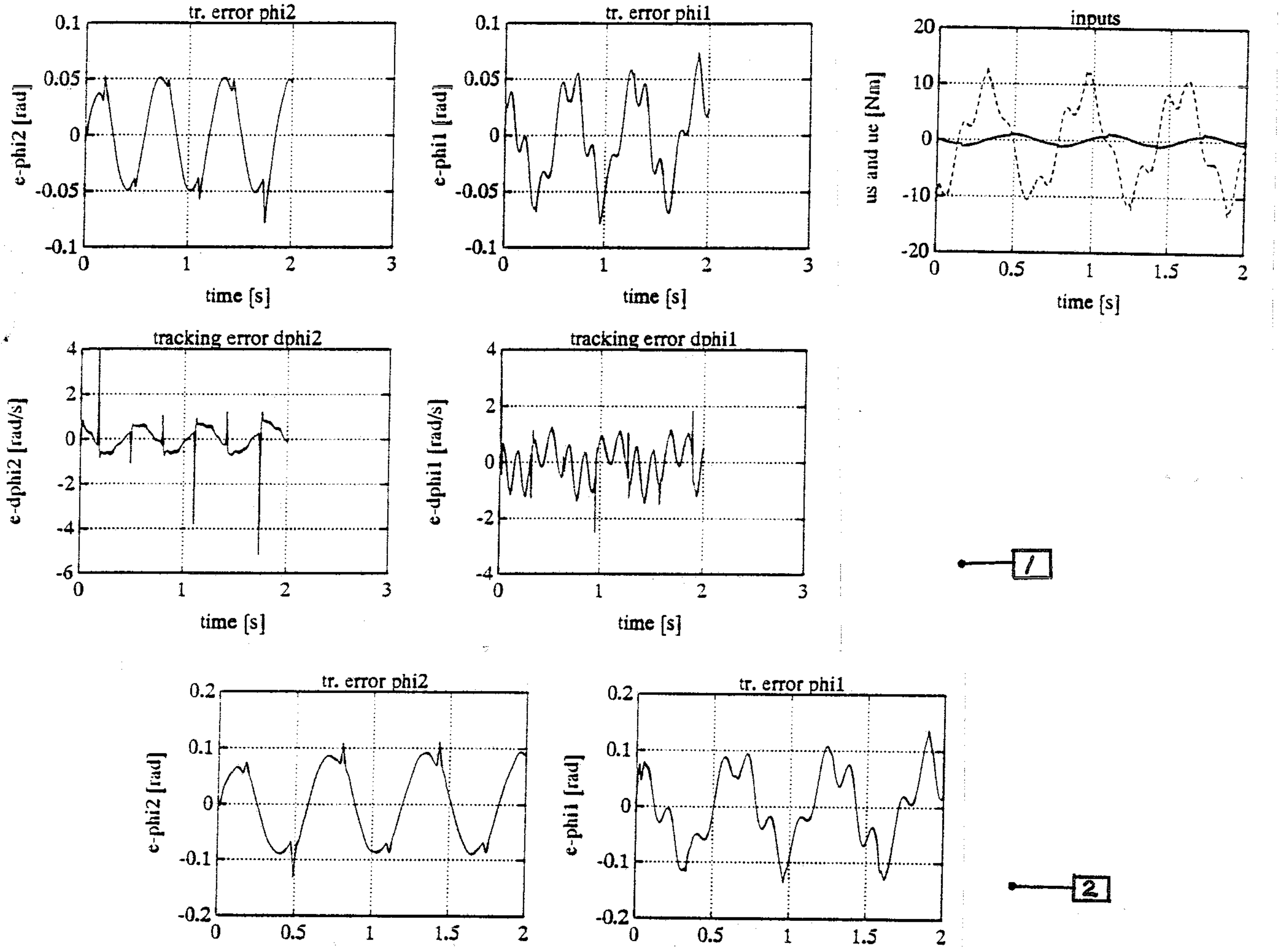


fig 4.7 Influence of measure frequence and discontinuous inputs

It appears that the system behaviour stays stable and that the tracking - and velocity errors will increase (compared with the standard situation). 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 the inputs stay realistic, and that the tracking errors will increase if we choose a lower measure frequence. Note: if we choose the measure frequence too low the system will get an instable system behaviour. See also appendix C where some more simulation results are shown.

Now I will consider the influence of wrong estimated parameters and unmodelled dynamics. Compared with the standard situation I have changed the next aspects:

- control gain parameters: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
- discretization: Δt_r= 0.001 s, Δt_s= 0.005 s (Δt_m= 0.006 s)

1- wrong estimated parameters: reality: w1= 0.47 Nm model: w1e= 0.3 Nm
w2= 0.15 Nm w2e= 0.1 Nm
2- unmodelled dynamics: reality: w1= 0.47 Nm model: w1e= 0 Nm
w2= 0.15 Nm w2e= 0 Nm

simulation results:

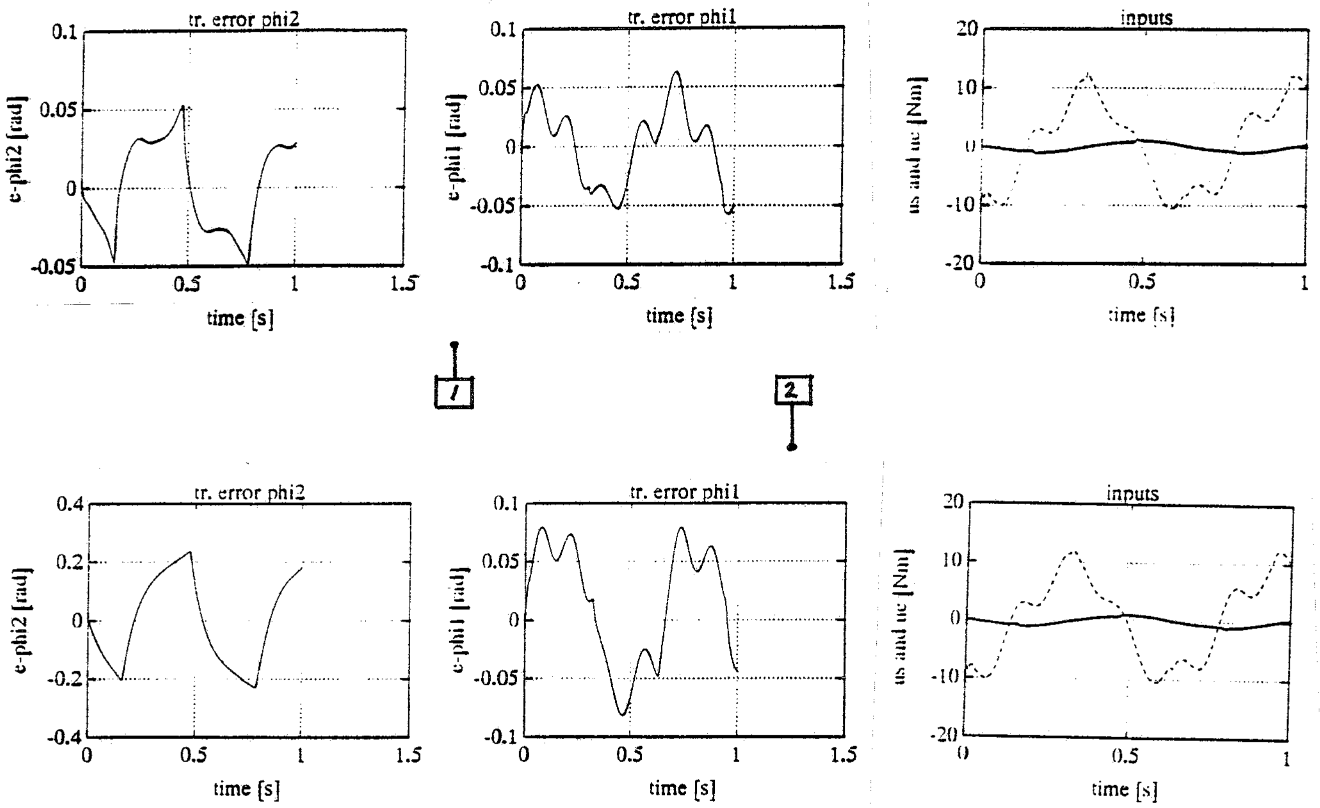


fig 4.8 influence of unmodelled dynamics and wrong estimated parameters

It appears that the system stays stable, and that the errors will increase (compared with the standard situation). The inputs are realistic.

I have executed some other simulations with wrong estimated parameters (see appendix C). It appears that the system stays stable if the estimated parameters are not chosen too worse.

Now I will consider the influence of measure noise. Compared with the standard situation I have changed the next aspects:

- control gain parameters: $k_1= 0.5$, $k_2= 0.05$, $k_3= 0.05$, $\lambda_1= 10$ and $\lambda_2= 10$
- discretization: $\Delta t_r= 0.002$ s, $\Delta t_s= 0.008$ s ($\Delta t_m= 0.01$ s)
- measure errors: - angular position → white noise $v_p= a_{vp}\text{rand}(t) \rightarrow a_{vp}= 0.1$
  - angular velocity → white noise $v_v= a_{vv}\text{rand}(t) \rightarrow a_{vv}= 1$

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

simulation results:

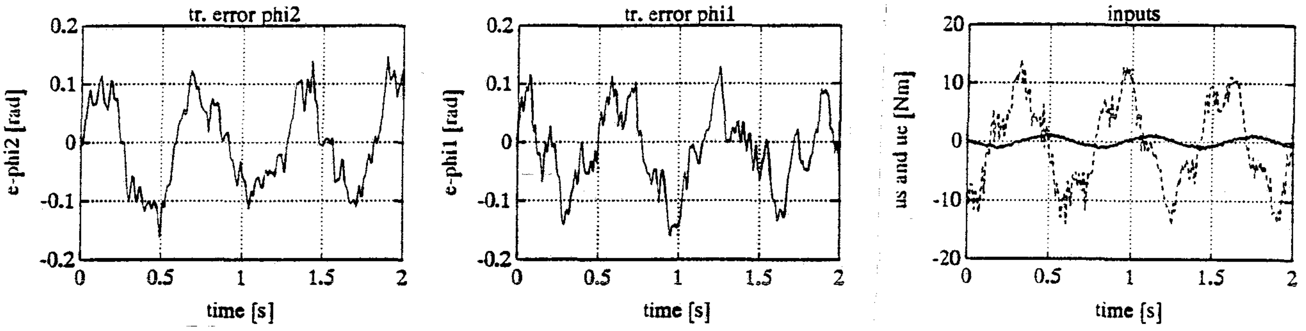


fig 4.9 influence of measure noise

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 this situation 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} the errors will increase.

In appendix C I have given some more simulation results of practical situations.

It appears from the practical simulation results that the robustness against unmodelled dynamics, wrong estimated parameters, discontinious inputs, time delay to calculate the inputs, measure noise and measure inaccuracy is reasonable.

The mean conclusion is that the theoretical - and practical simulation results are satisfying and confirm with the expectations. In practice there will be a reasonable chance that the C CTC law applied to the xy table will answer to our desirements (→ stable system behaviour, small tracking errors).

5. APPLYING OF THE CTC CONTROLLER TO THE XY TABLE

Now that we have a good picture of the properties of the controlled system we can focus our attention to the specific controlling of the xy table. This is the second part of the research.

The total system

In chapter 3 and appendix A I have shown a schematic representation of the xy table. In this paragraph I will give some more details of the measure- and control system of the xy table (see fig 5.1).

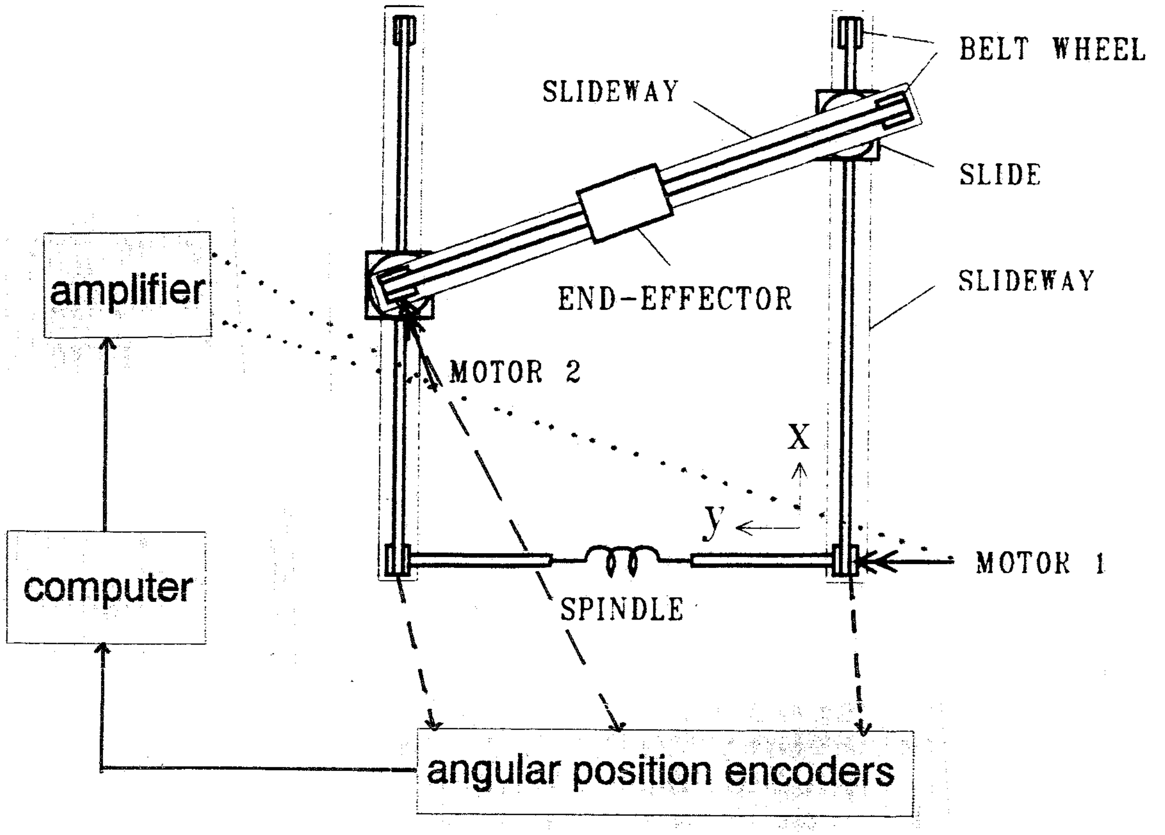


fig 5.1 Schematic representation of the total system.

The designed control routine has been put in the computer. The inputs of the control routine are the measurements of the angular positions φ_1, φ_2 and φ_3. The outputs are the by the control routine calculated small electrical currents. Before the currents are presented to the motors they are amplified. Depending on these electrical currents the motors deliver certain torques. These torques are the inputs of the system (u_s and u_e).
For the total organisation of executing experiments I have the avaliability of the new software designed by Jos Banens (member of the WFW group). This software is written in "C". The big advantage of using this software is that it is build up in standard routines and that the programs can communicate with Matlab programs.

The total system has its limitations. The most important ones are:

- The calculated inputs are clipped (are bounded) on a certain positive and negative value. This to protect the amplifier and the motors.
- Inaccuracy of the amplifier.
- The elasticity of the xy table is restricted. It appears that max $|\varphi_1 - \varphi_3| = \pm 3$ rad.
- The computing time. It will cost time to calculate the inputs. The computing time may not be too big (see chapter 4).
- Inaccuracy of the measurements. The measurements of the angular position do have a certain inaccuracy.
- Curved axis (y direction) of the xy table.

Implementation of the C CTC controller

For the controlling of the experimental xy table I use the C CTC law which is shown in chapter 3 (3.6 to 3.8). However, I have changed the friction term $\underline{n}$. It appears that in practice it is more realistic to model the friction as dry friction $\underline{n}_d$ (see appendix A).
This C CTC law has been implemented in one of the routines of the C-program designed by Jos Banens. Because of the fact that we have to do with a discret time system, we have to discretisize the C CTC law. The dicretization has been executed as follows:

general method: Euler scheme

$$\dot{x} = ax+bu \quad \rightarrow \quad x_{i+1} = x_i + \Delta t(ax_i + bu_i) \tag{5.1}$$

note: state x_{i+1} is Δt seconds after x_i.

The reference trajectories:

$$
\begin{aligned}
\ddot{\varphi}_{1r} &= \ddot{\varphi}_{1d}+\lambda_1(\dot{\varphi}_{1d}-\dot{\varphi}_1)\\
\dot{\varphi}_{1r} &= \dot{\varphi}_{1d}+\lambda_1(\varphi_{1d}-\varphi_1)\\
\varphi 1r_{i+1} &= \varphi 1r_i+\Delta t\dot{\varphi} 1r_i+\frac{1}{2}\Delta t^2\ddot{\varphi} 1r_i
\end{aligned}
\tag{5.2}
$$

$$
\begin{aligned}
\ddot{\varphi}_{2r} &= \ddot{\varphi}_{2d}+\lambda_2(\dot{\varphi}_{2d}-\dot{\varphi}_2)\\
\dot{\varphi}_{2r} &= \dot{\varphi}_{2d}+\lambda_2(\varphi_{2d}-\varphi_2)\\
\varphi 2r_{i+1} &= \varphi 2r_i+\Delta t\dot{\varphi} 2r_i+\frac{1}{2}\Delta t^2\ddot{\varphi} 2r_i
\end{aligned}
\tag{5.3}
$$

$$
\begin{aligned}
\ddot{\varphi}_{3r} &= -(\frac{c_{33}+k_3}{m_{33}})\dot{\varphi}_{3r}-\frac{k}{m_{33}}\varphi_{3r}-\frac{1}{m_{33}}(m_{31}\ddot{\varphi}_{1r}+m_{32}\ddot{\varphi}_{2r}+c_{31}\dot{\varphi}_{1r}+c_{32}\dot{\varphi}_{2r}-k\varphi_{1r}+n_{d33}-k_3\dot{\varphi}_3)\\
\dot{\varphi} 3r_{i+1} &= \dot{\varphi} 3r_i+\Delta t\ddot{\varphi} 3r_i\\
\varphi 3r_{i+1} &= \varphi 3r_i+\Delta t\dot{\varphi} 3r_i+\frac{1}{2}\Delta t^2\ddot{\varphi} 3r_i
\end{aligned}
\tag{5.4}
$$

For the calculation of the reference trajectories and the inputs, we need:

1. desired trajectories φ_{1d}, φ_{2d} (can be chosen)
2. estimated system parameters (see appendix A)
3. control gain parameters (can be chosen)
4. the angular position and angular velocities (φ_1, φ_2, φ_3, $\dot{\varphi}_1$, $\dot{\varphi}_2$ and $\dot{\varphi}_3$)

The first two points needs no further discussion. The third point about the control gain parameters needs a short explanation. As shown in chapter 3 (proof of stability) and chapter 4 the control gain parameters do partitial fix the poles of $\dot{V}$ (see 4.10):

$$
\dot{V} = -\underline{\dot{e}}_r K_r \underline{\dot{e}}_r+\eta \tag{5.5}
$$

If the poles of $\dot{V}$ have negative real values, the system is stable. Because of the fact that η is not known we have to find suitable control gain parameters by executing simulations and experiments.

Note: the control gain parameters can't be chosen too big, because of the fact that the inputs are clipped on certain values. This clipping of the inputs can lead to undesired system behaviour.
The fourth point deals with the handling of the measurements. As already written, the angular positions φ_1, φ_2 and φ_3 are measured. Out of these measurements and out of the (partial) known system behaviour we have to estimate the angular velocities $\dot{\varphi}_1$, $\dot{\varphi}_2$ and $\dot{\varphi}_3$. In other terms, we have to design an observer algorithm for the reconstruction of the angular velocities. In appendix D I have designed such an algorithm.

Now it is possible to calculate the inputs u_s and u_e (see 3.6 and 3.8):

$$u_s = m_{22}\ddot{\varphi}_{2r}+m_{23}\ddot{\varphi}_{3r}+c_{23}\dot{\varphi}_{3r}+c_{21}\dot{\varphi}_{1r}+n_{d22}+k_2\dot{e}_{2r} \tag{5.6}$$

$$u_e = m_{11}\ddot{\varphi}_{1r}+m_{13}\ddot{\varphi}_{3r}+c_{12}\dot{\varphi}_{2r}+c_{13}\dot{\varphi}_{3r}+c_{11}\dot{\varphi}_{1r}+n_{d11}+k(\varphi_{1r}-\varphi_{3r})+k_1\dot{e}_{1r} \tag{5.7}$$

The research

The goal of this part of the research is to test the C CTC law in practice and to find a suitable (or optimal) way for controlling the xy table with a flexible transmission. Further I will consider the robustness of the controller.
For the executing of a part of the research I have designed a simulator. The simulator fits the reality on a close way (model with dry friction, time delay for the calculation of the inputs, clipping of the inputs, measure noise and the possibility to implement unmodelled dynamics and wrong estimated parameters). The reason why I have designed a simulator is:

- Designed control laws and system configurations can be tested. This is to protect the xy table.

- Some parts of the research can be executed on a more suitable way (e.g. research robustness, determination of the observer parameters).

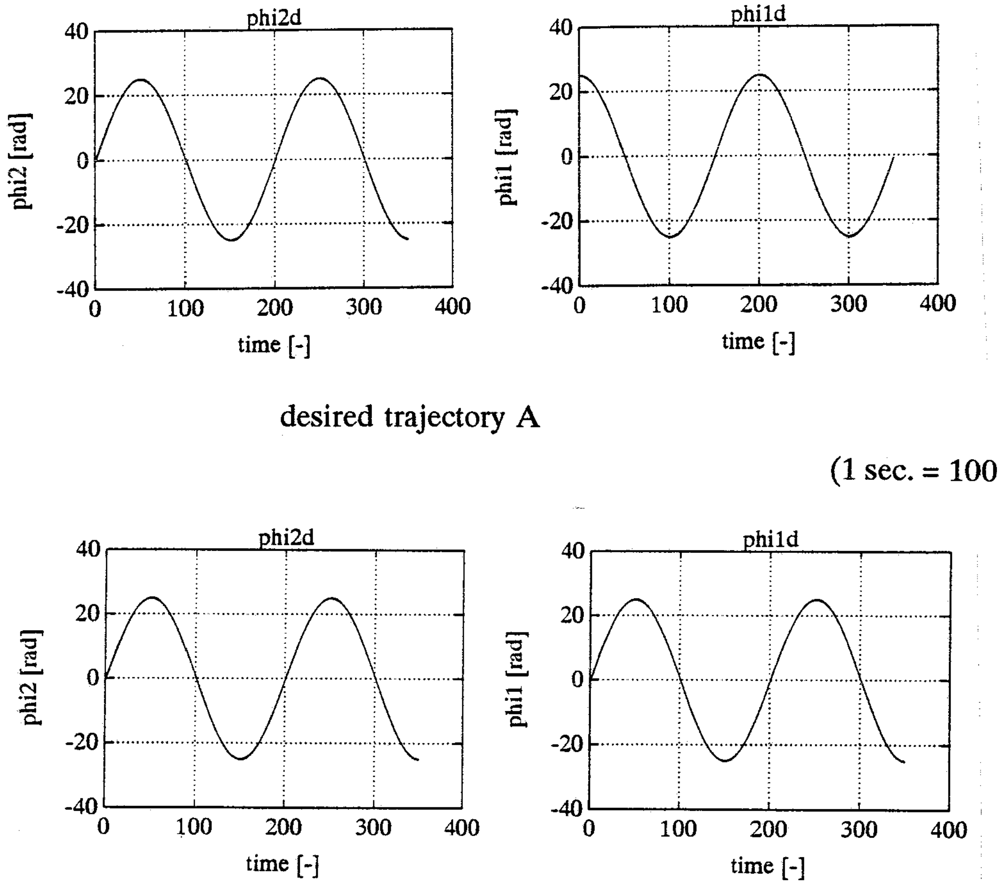
In first instance the goal of applying the C CTC law is to track the desired trajectories of φ_{1d}, φ_{2d} and the time derivatives. This is a state control routine. By executing simulations and experiments I have to find out how good the control law will answer to our desirements, and of which aspects the results are depending.

The simulation - and experimental results

During this part of the research I have executed a lot of simulations and experiments. Now I will show and discuss the most important results.

I have chosen the next desired trajectories:

(1 sec. = 100 time steps).



desired trajectory A

(1 sec. = 100 time steps)

desired trajectory B

fig 5.2 desired trajectories

I define the next <u>standard situation</u>:

- desired trajectory: A
- springconstant: k = 0.5 Nm/rad
- control gain parameters: $k_1 = k_2 = k_3 = 0.05$, $lab_1 = lab_2 = 25$
- observer-poles (used by the pole-placement routine): -25+0.2i
-25+0.3i
-25+0.4i
- measure-frequence: 1/0.005 herz → 1 second = 200 timesteps
- execution time: 3.5 s
- position t=0: $\varphi_1 = \varphi_2 = \varphi_3 = 0$ (middle of the xy table)
- velocity t=0: $\varphi_1 = \varphi_2 = \varphi_3 = 0$

The simulation results are:

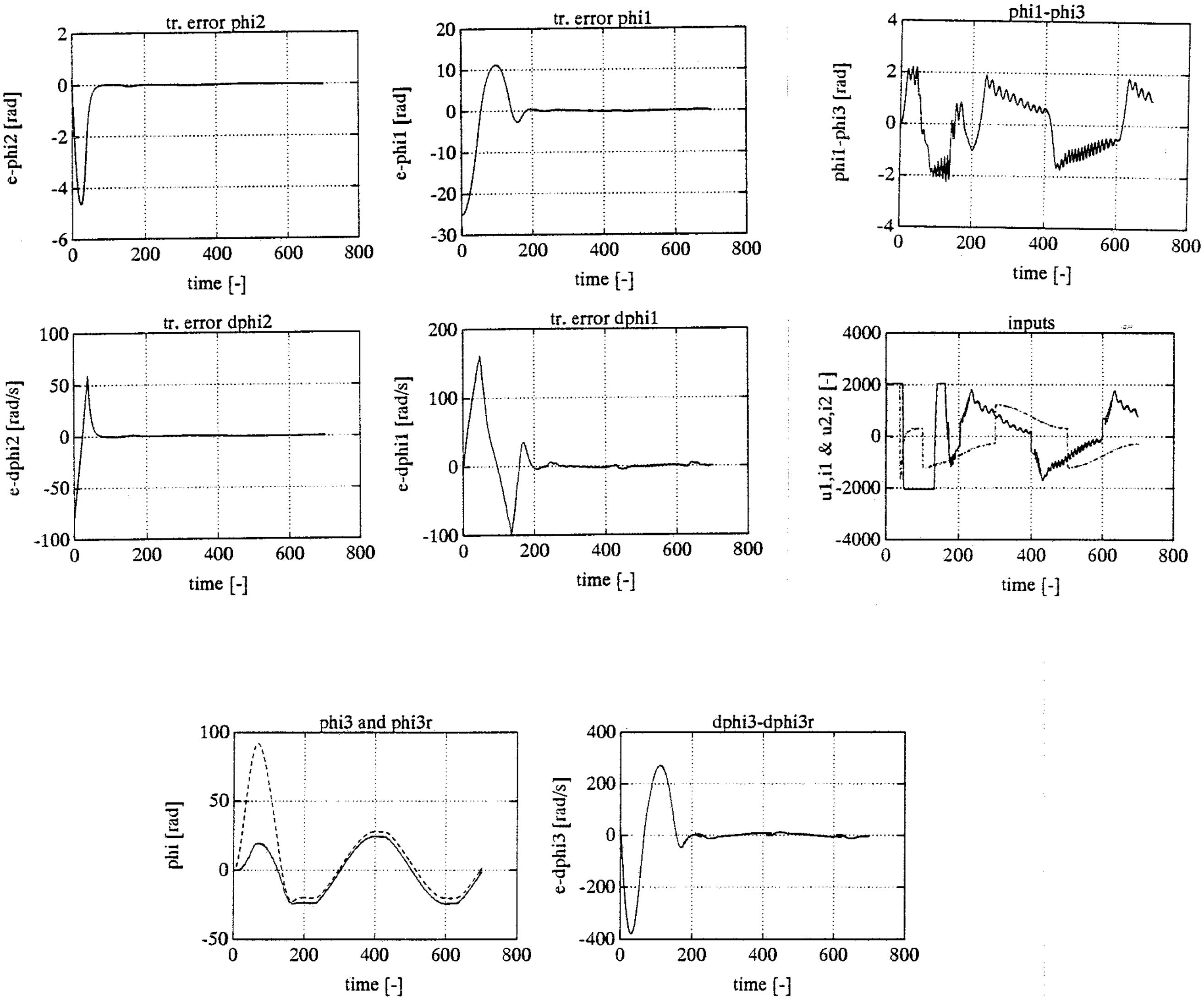


fig 5.3 simulation results standard situation

It appears that the simulation results are good. The system stays stable and the tracking - and velocity errors converge to zero (to small values).

At the first part of the simulation the controller has to control the start errors to zero. This leads to big control actions. It appears that the by the control routine calculated inputs u_s and u_e are clipped. In this case the clipping of the inputs doesn't lead to undesired system behaviour.

Now I will consider the experimental results of the standard situation:

Note: the experiment is executed with a measure-frequence of 1/0.01 Herz (1 sec. = 100 time steps).

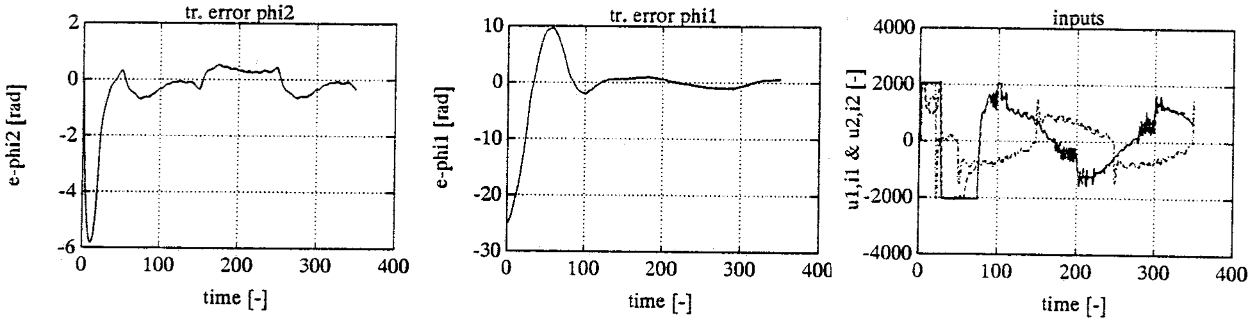


fig 5.3 experimental results standard situation

The difference between the simulation results and the experimental results is small. In the case of the experiments we have to do with unmodelled dynamics, wrong estimated parameters, inaccuracy amplifier, etc. It appears that this aspects will cause small tracking - and velocity errors. In the case of the simulations we don't have the influence of above mentioned aspects (see also discussion of robustness against wrong estimated parameters page 33).

Executing experiments with a measure-frequence of 1/0.005 Herz is not possible. The computersystem can't calculate the inputs u_s and u_e within 0.005 seconds. Of cause this will lead to undesired system behaviour.

Now I will change compared with the standard situation one or some aspects, and I will consider the influence of the changes.

If we change the desired trajectory we get:

- desired trajectory: B
- measure-frequence: 1/0.007 Herz (1 sec. = 142 timesteps)

Experimental results

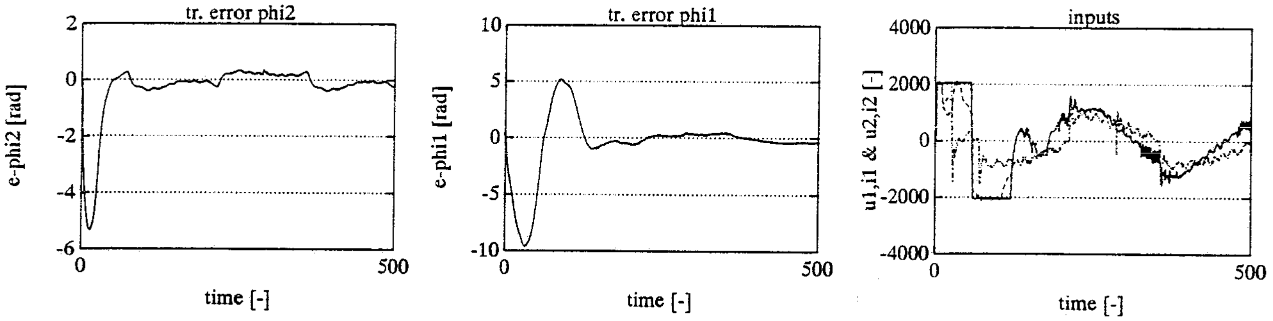


fig 5.4 influence of desired trajectory

It appears that the results are good. The choice of the desired trajectory has no influence to the control prestations (see also appendix E where I have shown some more simulation - and experimental results).
The measure-frequence of 1/0.007 Herz is the maximum value with which the experiments can be executed.

If we changed the springconstant k we get:

- springconstant: k=0.05
- desired trajectory: B

Simulationresults

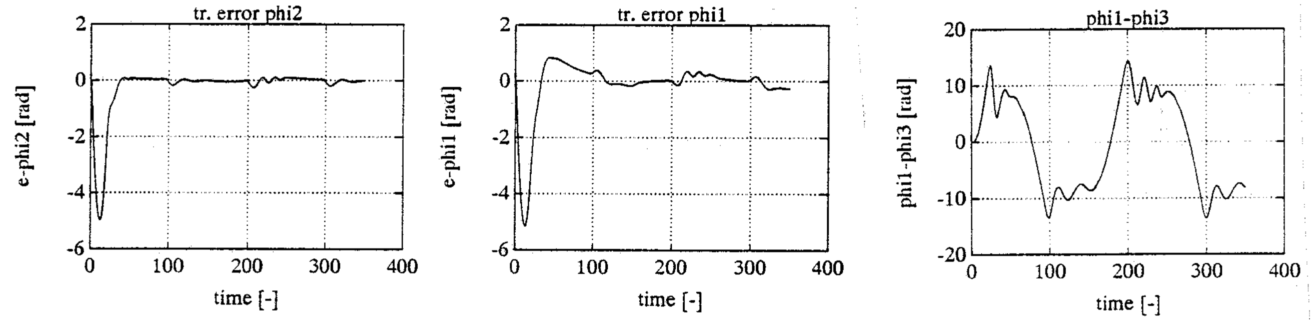


fig 5.5 influence of the springconstant k

If we change the elasticity it appears that the C CTC law can control the system. The tracking- and velocity errors stay small, even if we choose a small value of the springconstant.

Now I will consider the robustness of the controller against wrong estimated parameters. This research is executed with the simulator because of the fact that in that case the system behaviour is exactly known.

Compared with the standar situation I have changed:

1 wrong estimated parameter: reality : k=0.5 Nm/rad (used in the simulator)
(see also chapter 4) estimated parameter: k_e=0.1 Nm/rad
(used in the control law)

2 wrong estimated parameters: reality: w_1= w_3= 0.5 Nm, w_2= 0.15 Nm
estimated parameters: w_{1e}=w_{3e}= 0.3 Nm, w_{2e}=0.05 Nm

Simulation results:

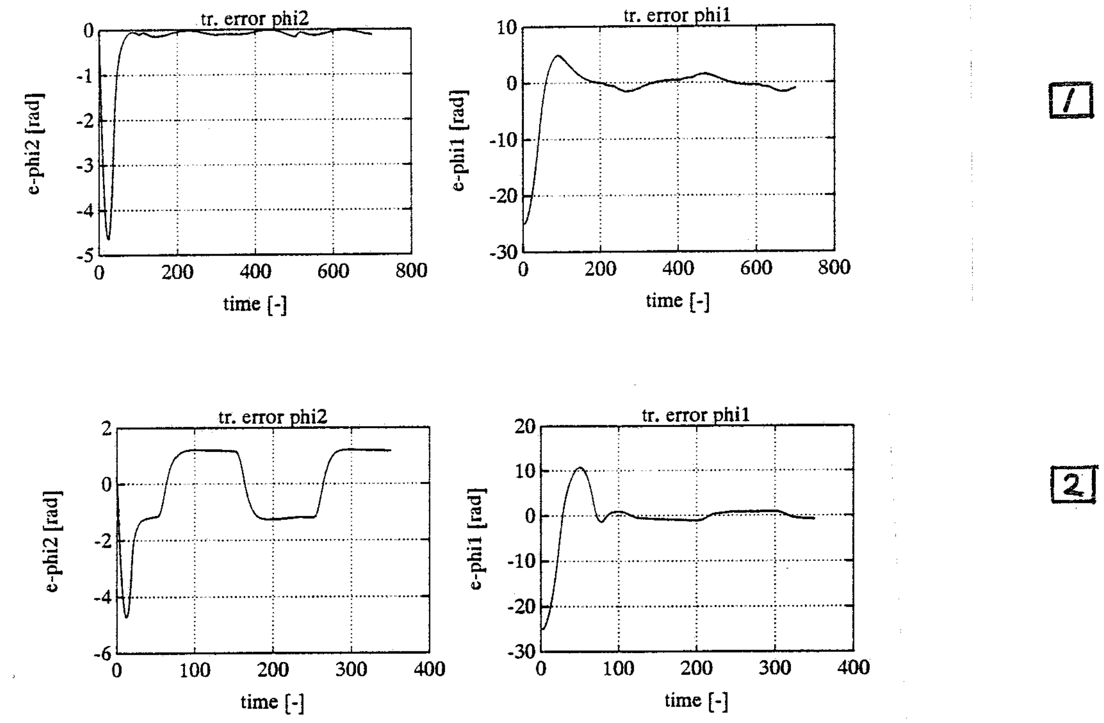


fig 5.6 influence of wrong estimated parameters

It appears that wrong estimated parameters leads to bigger tracking - and velocity errors. Wrong estimated springconstants and friction coefficients have a bigger influence than the other parameters (see appendix E).
It appears that the considerd wrong estimated parameters don't lead to instable system behaviour. So the robustness of the system is reasonable.

To get an idea of the control prestations, I have made a comparison between the control prestations of a C CTC controller and a CTC controller.
Till now the controlling of robots (with or without flexibilities) is in much cases executed by the CTC control strategy. To design a CTC controller the elasticity of the system is neglected. So, the starting point of designing a CTC controller to control the flexible xy table is a model of the xy table without flexibilities.
The model of the xy table with a stiff transmission is (see v.d. Molengraft):

$$\begin{aligned} m_2\ddot{\varphi}_2+n_{d22} &= u_s \\ m_1\ddot{\varphi}_1+n_{d11} &= u_e \end{aligned} \tag{5.8}$$

This model is used to design the CTC controller:

$$\begin{aligned} u_s &= m_2\ddot{\varphi}_{2r}+n_{d22}+k_2\dot{e}_{2r} \\ u_e &= m_1\ddot{\varphi}_{1r}+n_{d11}+k_1\dot{e}_{1r} \end{aligned} \tag{5.9}$$

I have used above CTC law to control the flexible xy table. The simulationresults are:

- desired trajectory A
- measure frequence 1/0.01 Herz → 1 sec. = 100 timesteps
- control gain parameters: $k_1 = k_2 = 0.05$, $\lambda_1 = \lambda_2 = 25$

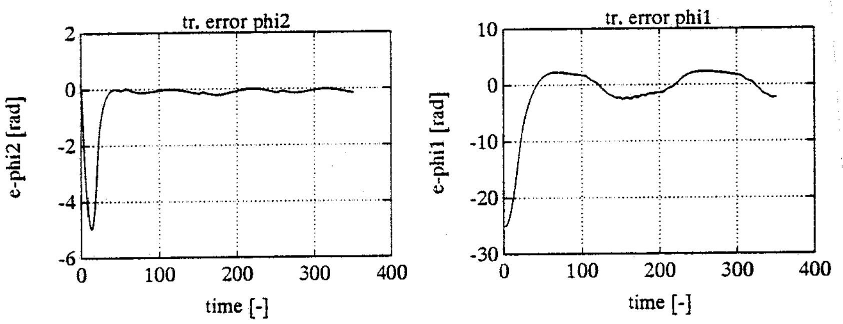


fig 5.7 simulation results CTC controller

If we make a comparison between the control prestations of the C CTC controller (see standard situation) and the CTC controller, it appears that the control prestations of the C CTC controller are (much) better. The flexibility of the system plays an important role, but

it appears that this flexibility doesn't lead to an instable system behaviour (see also appendix E).
Note: For a system with flexible transmissions controlled by the general CTC law it is not possible to guarantee the stability.

The mean conclusion of this part of the research is that the C CTC law answers to our desirements. The tracking - and velocity errors converge in all considered cases to zero (to small values). The speed of controlling is reasonable. Further it appers that the robustness of the control system is reasonable, and that the control prestations of the C CTC controller are better than the control prestations of the CTC controller.

6. CONTROLLING THE END-EFFECTOR

The mean reason of controlling the xy table is that the end-effector will track a certain desired path. To achieve this, the designed C CTC control routine has to be changed from a state control routine ($\underline{q}_1 \rightarrow \underline{q}_{1d}$) to an output control routine ($[x,y] \rightarrow [x_d,y_d]$), see also page 12. During this part of the research I have try to find a suitable way to change the C CTC state control routine in a C CTC output control routine.
In this chapter I will show and discuss how the C CTC routine is changed, and I will show and discuss some simulation - and experimental results.

Note: this part of the research has to be considered as trial and error research.

Output control

In order to get reasonable tracking of the desired end-effector path, I have try to find a routine to generate the desired trajectories φ_{2d} and φ_{1d} on-line. For the on-line generation of the desired trajectories φ_{2d} and φ_{1d}, the obliqueness of the slideway of the xy table (= φ_1-φ_3) is considered. This obliqueness is discounted in the on-line generation. The desired trajectories φ_{2d} and φ_{1d} are generated in such a way that the end-effector will track a certain desired path $[x_d,y_d]$, no matter how big the value of the obliqueness (φ_1-φ_3).

For the x-position of the end-effector it follows:

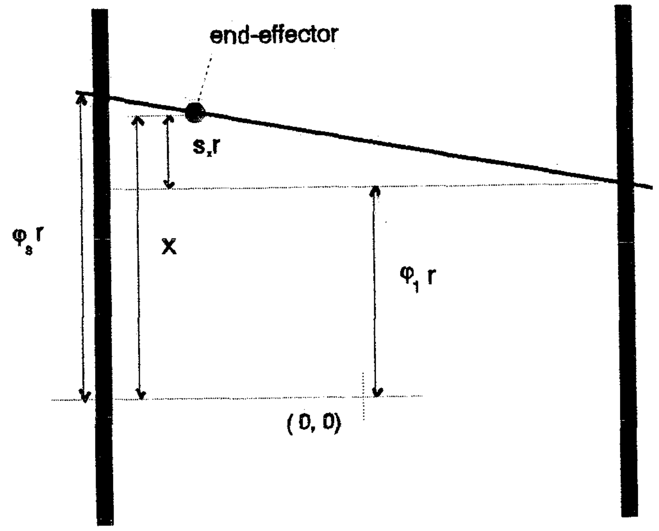


fig 6.1 x-position end-effector

$$x = \varphi_1 r + s_x r \tag{6.1}$$

See appendix F for s_x

The desired path in x-direction of the end-effector is x_d. With equation (6.1) it is possible to define an on-line generation routine of the desired trajectory of φ_1:

$$\varphi 1d_{(t+\Delta t)} = \frac{1}{r} xd_{(t+\Delta t)} - sx_{(t)} \tag{6.2}$$

On point of time t the routine generates the value of the desired trajectory φ_{1d} belonging to point of time t+Δt. So, on every discret timestep the routine generates the next value of φ_{1d}.

Now I will consider the y-position of the end-effector. It follows:

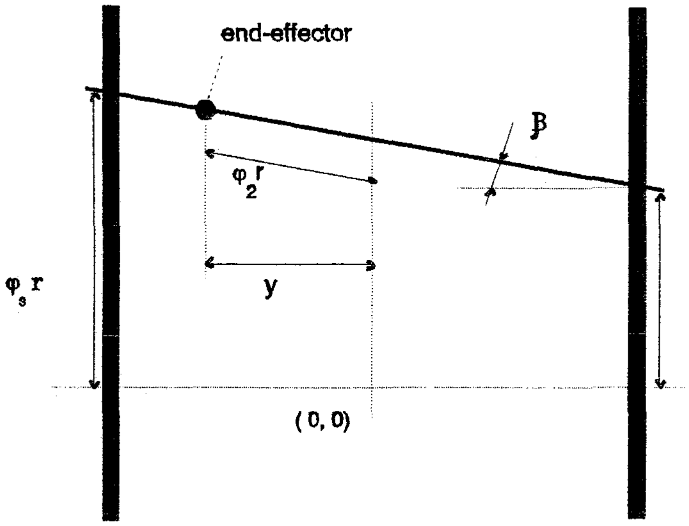


fig 6.2 y-position end-effector

$$y = \varphi_2 r \cos\beta \tag{6.3}$$

Equation (8.3) can be written in the following form (see appendix F):

$$y = \varphi_2 r - s_y \tag{6.4}$$

The desired trajectorie of φ_2 can be defined as:

$$\varphi 2d_{(t+\Delta t)} = \frac{(yd_{(t+\Delta t)} + sy_{(t)})}{r} \tag{6.5}$$

Now we have defined an output control routine which is based on the C CTC strategy. This routine generates with (6.2) and (6.5) on-line the desired trajectories φ_{2d} and φ_{1d} .

Simulations and experiments

I have executed some simulations with the simulator mentioned in chapter 5, and I have executed some experiments. Now I will give the most important simulation results (see also appendix F where I have shown some simulation - and experimental results).

First I will consider a situation where the end-effector is controlled with the general C CTC law (no on-line generation of the desired trajectories φ_{2d} and φ_{1d}). This is the state control routine. I have chosen the following situation:

- desired trajectory: x_d= 0.25*sin(3.14*t) , y_d= 0.25*sin(3.14*t) (see fig. 6.3)
- springconstant: k = 0.5 Nm/rad
- control gain parameters: k_1= k_2= k_3= 0.05, lab_1= lab_2= 25
- observer-poles (used by the pole-placement routine): -25+0.2i
-25+0.3i
-25+0.4i
- measure-frequence: 1/0.005 herz → 1 second = 200 timesteps
- execution time: 3.5 s
- position t=0: φ_1= φ_2= φ_3= 0 (middle of the xy table)
- velocity t=0: φ_1= φ_2= φ_3= 0

The simulation results are:

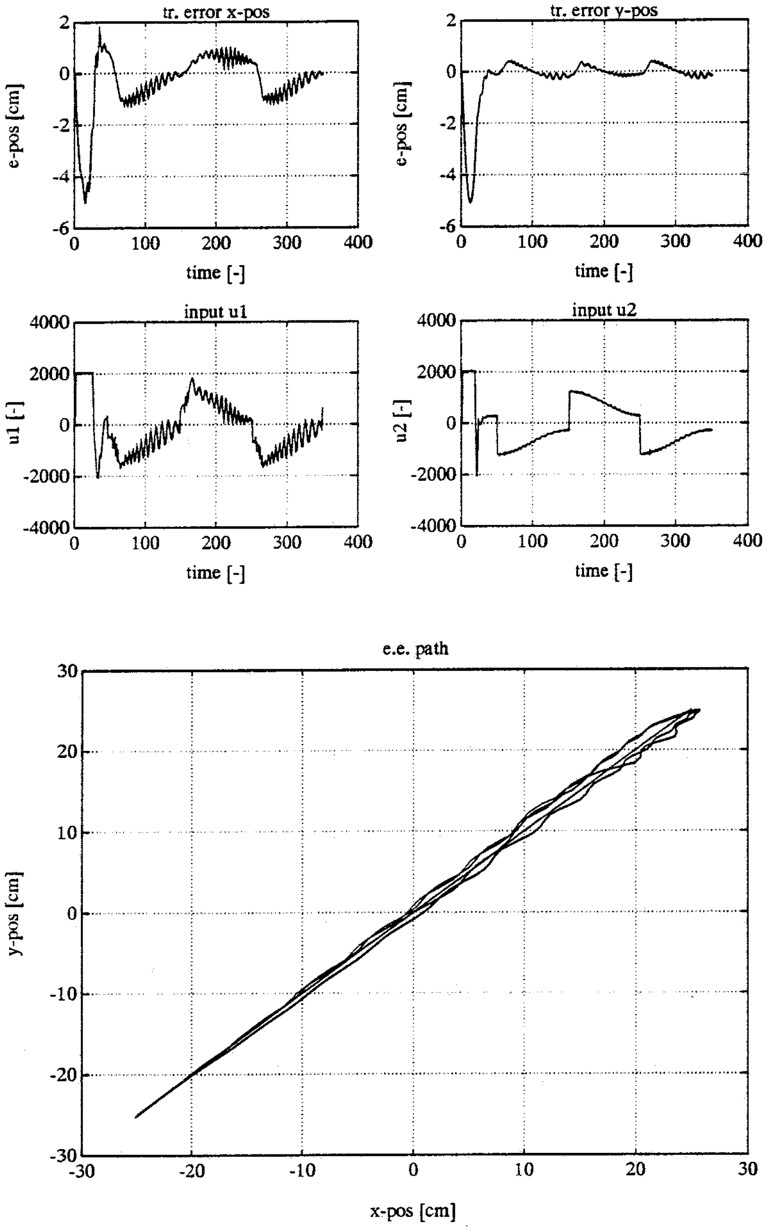


fig 6.3 simulation results of controlling the end-effector with the general C CTC strategy.

It appears that the control results are moderate. With the state control routine it is possible to control the end-effector, but the tracking errors don't converge to zero, but to certain values.

Now I will consider the designed C CTC output controller (on-line generation of the desired trajectories φ_{2d} and φ_{1d} with 6.2 and 6.5):

The simulation results are:

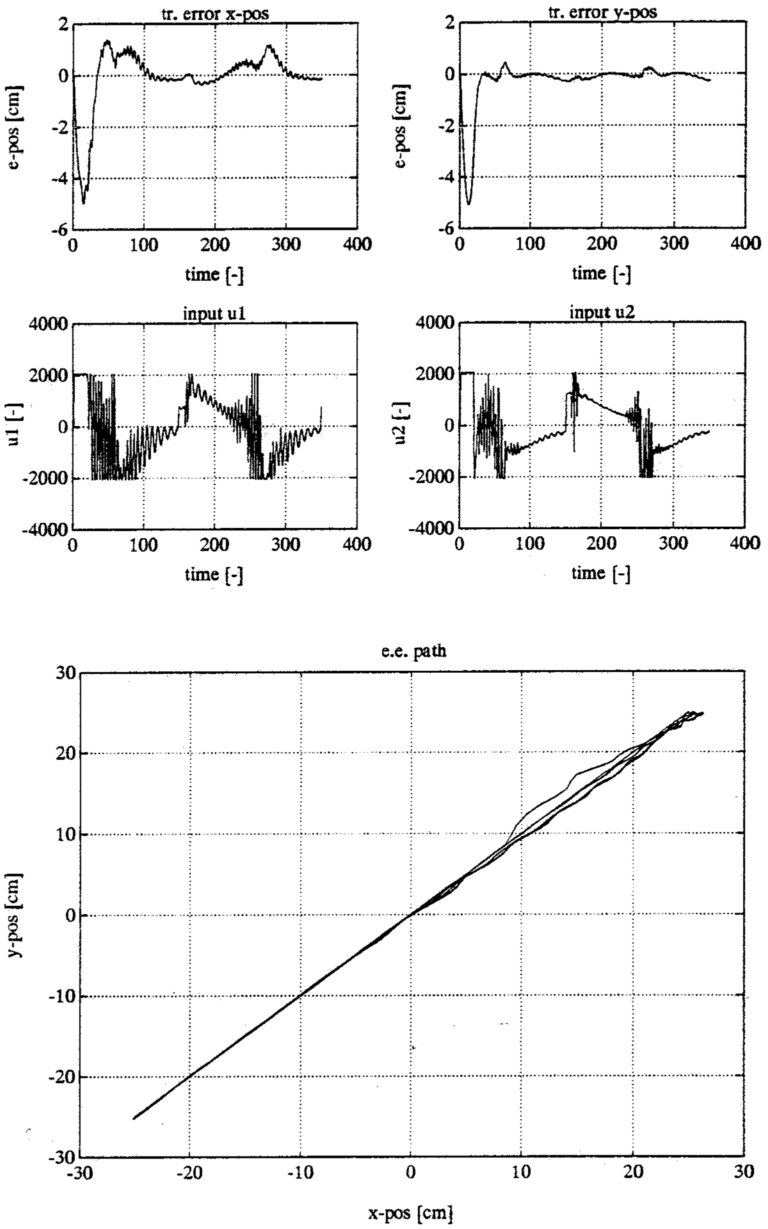


fig 6.4 simulation results of controlling the end-effector with a C CTC law with an on-line generation of the desired trajectories

It appears that the control results are becoming better but are not satisfying. It appears that the calculated inputs are growing too big and are clipped. This clipping of the inputs causes tracking errors of the end-effector.

In order to get better control results I have tried (trial and error research !) to find a way to change the on-line generation routine of the desired trajectories φ_{2d} and φ_{1d}. To reduce the inputs I have had the idea to scale the terms s_x and s_y of this generation routine (see 6.2 and 6.5):

x-direction:

$$\varphi 1d_{(t+\Delta t)} = \frac{1}{r} xd_{(t+\Delta t)} - a_x sx_{(t)} \tag{6.6}$$

a_x= scaling factor x-direction

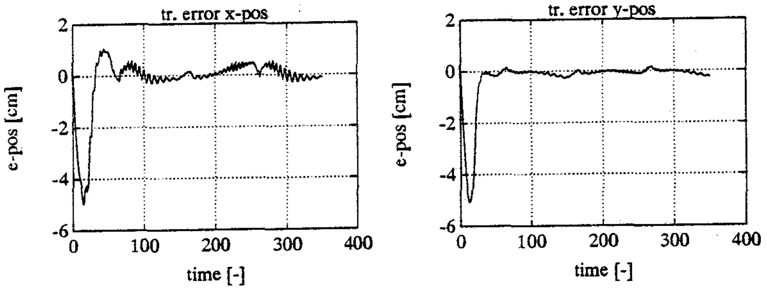
y-direction:

$$\varphi 2d_{(t+\Delta t)} = \frac{(yd_{(t+\Delta t)} + a_y sy_{(t)})}{r} \tag{6.7}$$

a_y= scaling factor y-direction

Now I will consider the C CTC controller with on-line generation of the desired trajectories φ_{2d} and φ_{1d} with the scaled values of s_x and s_y. The optimal values of a_x and a_y are found by execution of trial and error research.
The simulation results are:

- a_x= a_y= 0.75



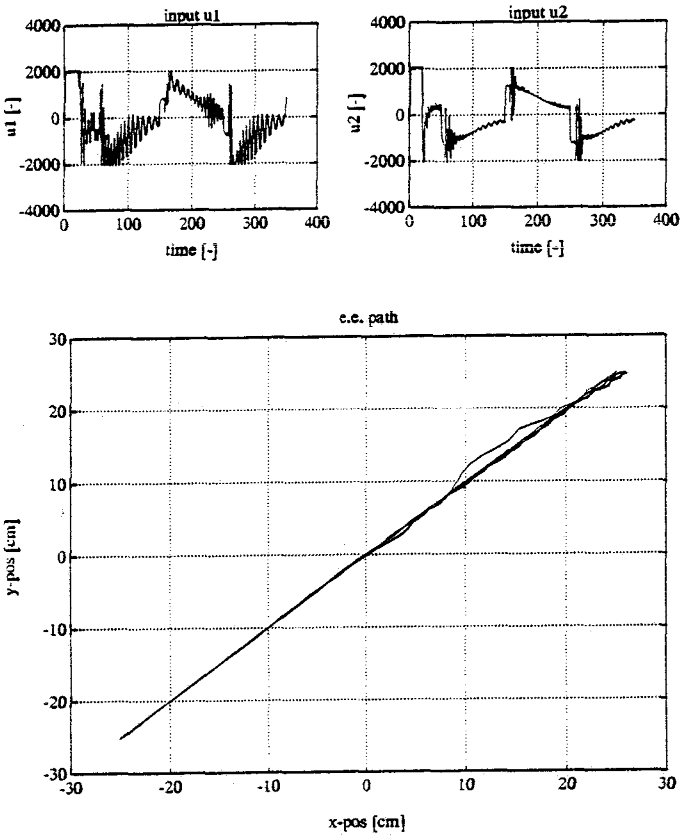


fig 6.5 simulation results of controlling the end-effector with a C CTC law with an on-line generation of the desired trajectories. Now s_x and s_y are scaled.

It appears that this on-line generation routine leads to the best control results.

The mean conclusion of this part of the research is that the by trial and error research designed output control routine leads to reasonable control results.
In order to get better results this part of the research has to be executed on a more fundamental way.

7. CONCLUSIONS AND RECOMMENDATIONS

Conlusions

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_{1d}= $[\varphi_{2d}\ \varphi_{1d}]^T$ and its time derivative. This is a state control routine.

It can be proved that applying of the C CTC law to control the flexible xy table 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.

From the 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 (robust system behaviour, small tracking errors).

The implementation of the C CTC law and an observer algorithm (for the reconstruction of the angular velocities) in the control system of the xy table is succeeded.

For the testing of designed control laws or system configurations and for the execution of a part of the research I have designed a simulator. The simulator fits the reality on a close way.

The new software designed by Jos Banens meant for the total organisation of executing experiments and simulations, is easy and pleasant to use.

The experimental results and accessory simulation results answers to our desirements. The tracking - and velocity errors converge to zero (to small values). The control speed is reasonable. Further it appears that the robustness of the control system is reasonable and that the control results are better that the control results of the general CTC law.

For the controlling of the end-effector I have designed (by trial and error) a C CTC output control routine. This routine generates on-line the desired trajectories of φ_1 and φ_2 . The results of this control method are reasonable.

part 2

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

theoretical analysis, simulations and experiments

by: Gert Brevoord

Professor: prof. dr. ir. J.J. Kok
Coach: ir. I.M.M. Lammerts

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

Report no.: 92.054

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 (see v.d. Molengraft, 1989).

The model of the xy table:

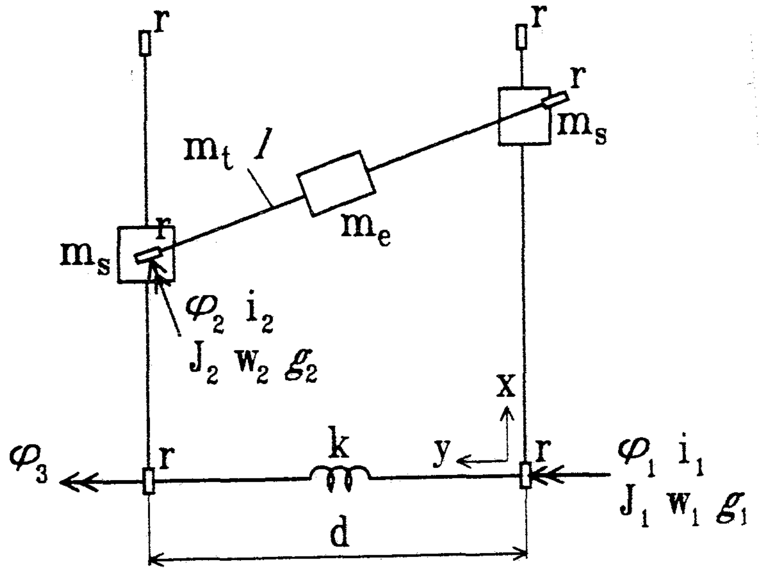


fig a1 model of the xy table

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

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

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

Note: At the second part of the research (chapter 5 and chapter 6) I have used an other friction model. This friction model fits the reality on a closer way. The friction model I have used during the second part of the research is a dry friction model:

$$\underline{n}_d = \begin{bmatrix} n_{d22} \\ n_{d33} \\ n_{d11} \end{bmatrix} \qquad \text{(A1.4)}$$

where,

$$\begin{aligned} &if\ \dot{\varphi}_1 \neq 0 \quad \rightarrow \quad n_{d11} = 0.25\ Nm \\ &if\ \dot{\varphi}_2 \neq 0 \quad \rightarrow \quad n_{d22} = 0.15\ Nm \\ &if\ \dot{\varphi}_3 \neq 0 \quad \rightarrow \quad n_{d33} = 0.25\ Nm \end{aligned} \qquad \text{(A1.5)}$$

$$\begin{gathered} if\ \dot{\varphi}_2 = 0 \quad \rightarrow \\ n_{d22} = u_s - m_{22}\ddot{\varphi}_2 - m_{23}\ddot{\varphi}_3 - c_{23}\dot{\varphi}_3 - c_{21}\dot{\varphi}_1 \end{gathered} \qquad \text{(A1.6)}$$

$$\begin{gathered} if\ \dot{\varphi}_3 = 0 \quad \rightarrow \\ n_{d33} = -m_{31}\ddot{\varphi}_1 - m_{32}\ddot{\varphi}_2 - m_{33}\ddot{\varphi}_3 - c_{32}\dot{\varphi}_2 - c_{33}\dot{\varphi}_3 - c_{31}\dot{\varphi}_1 + k(\varphi_1 - \varphi_3) \end{gathered} \qquad \text{(A1.7)}$$

$$\begin{gathered} if\ \dot{\varphi}_1 = 0 \quad \rightarrow \\ n_{d11} = u_e - m_{11}\ddot{\varphi}_1 - m_{13}\ddot{\varphi}_3 - c_{12}\dot{\varphi}_2 - c_{13}\dot{\varphi}_3 - c_{11}\dot{\varphi}_1 - k(\varphi_1 - \varphi_3) \end{gathered} \qquad \text{(A1.8)}$$

APPENDIX B: SIMULATION RESULTS OF SOME THEORETICAL SITUATIONS OF THE FIRST PART OF THE RESEARCH

In this appendix I will show 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: $\varphi_{2d}= \alpha_2+\alpha_2\cos(\omega_2 t)$; $\varphi_{2d}= -\alpha_2\omega_2\sin(\omega_2 t)$; ..
$\varphi_{1d}= \alpha_1+\alpha_1\cos(\omega_1 t)$; $\varphi_{1d}= -\alpha_1\omega_1\sin(\omega_1 t)$; ..
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: $\Delta\varphi_2(0)= 0$, $\Delta\varphi 1(0)= 0$, $\Delta\varphi_3(0)= 0$

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: $\Delta\varphi_2(0)$= 100 rad/s, $\Delta\varphi_1(0)$= 100 rad/s, $\Delta\varphi_3(0)$= 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:

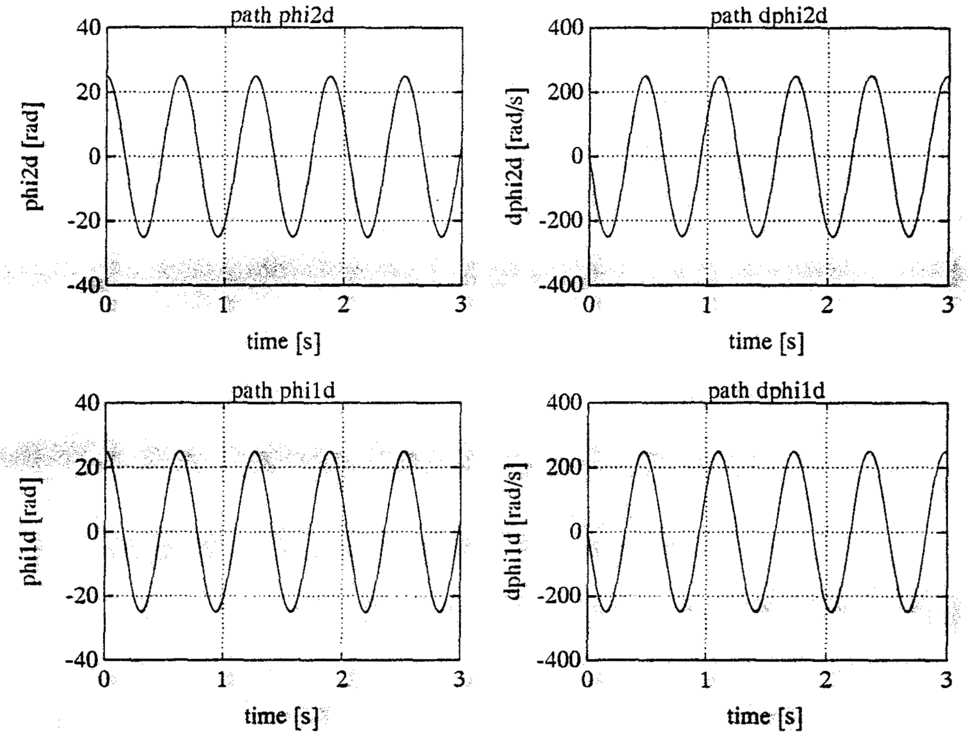
1	tracking error $\varphi_{2d}-\varphi_2$
2	tracking error $\varphi_{1d}-\varphi_1$
3	velocity error $\varphi_{2d}-\varphi_2$
4	velocity error $\varphi_{1d}-\varphi1$
5	$\varphi_{3r}-\alpha_1$ and $\varphi_3-\alpha_1$
6	velocity error $\varphi3r-\varphi_3$
7	elasticity $\varphi_3-\varphi_1$
8	inputs u_s and u_e

The structure of a page with plots is:

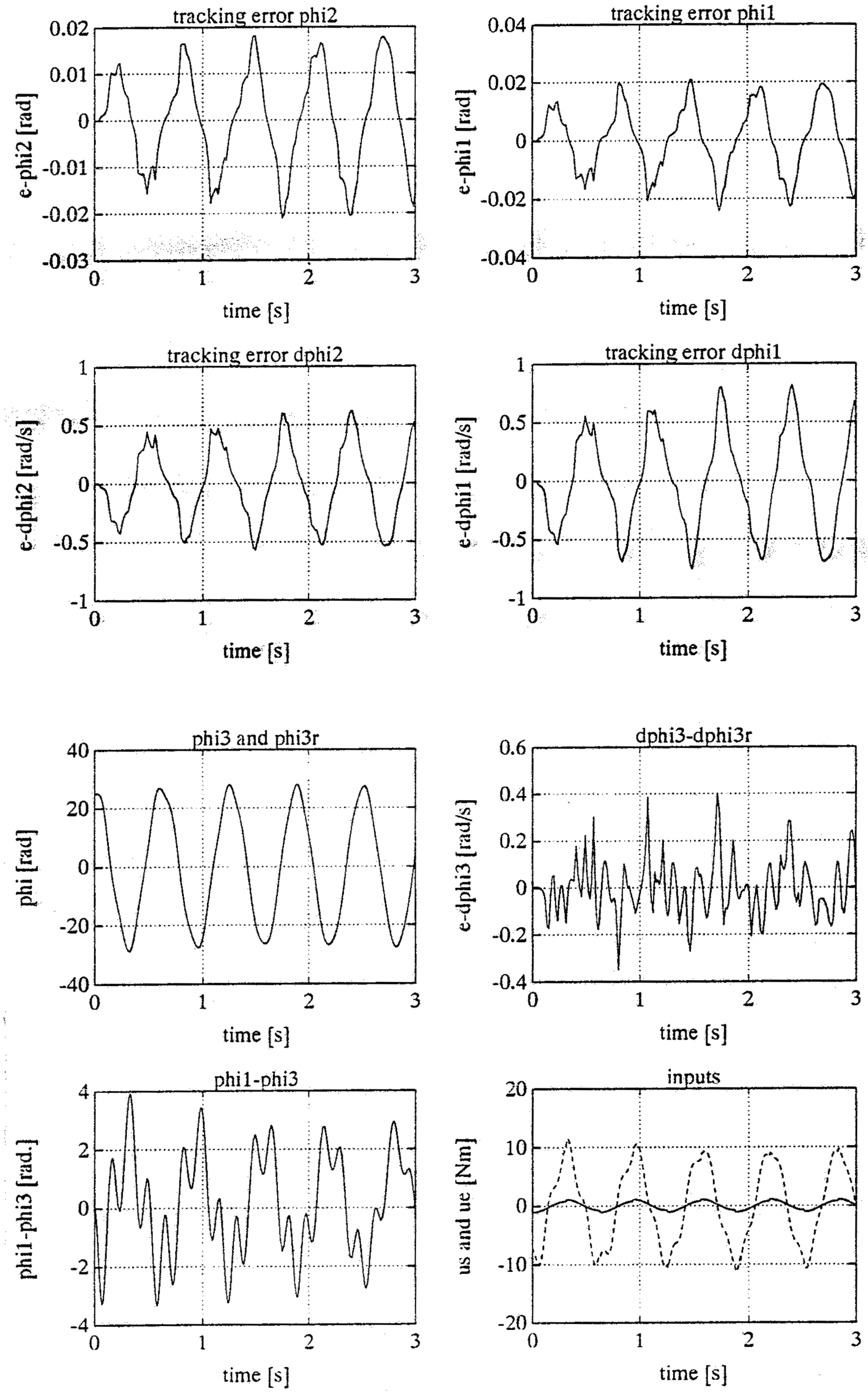
$$\begin{bmatrix} (1) & (2) \\ (3) & (4) \\ (5) & (6) \\ (7) & (8) \end{bmatrix}$$

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

Desired trajectorie: plot 0

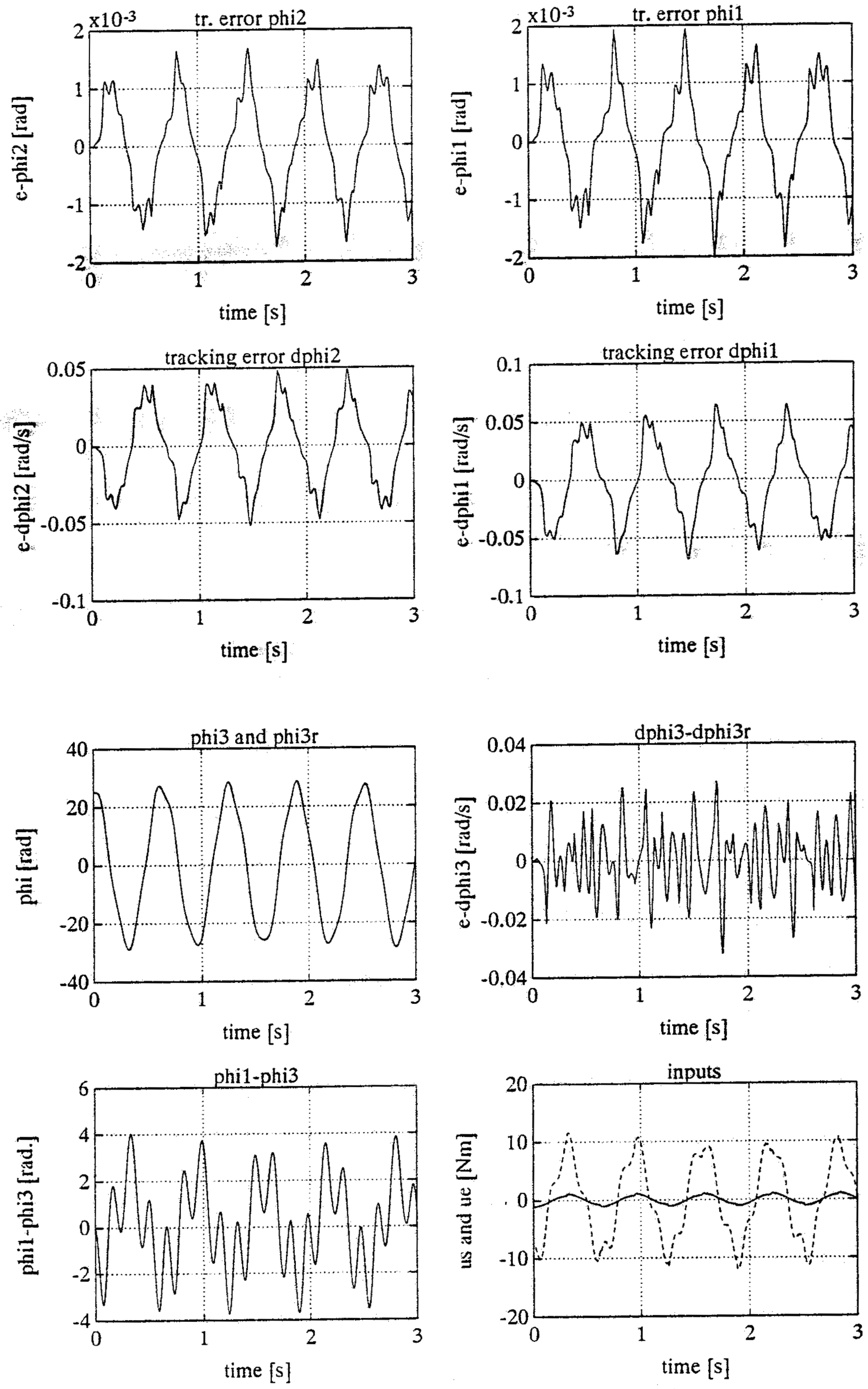


situation 1



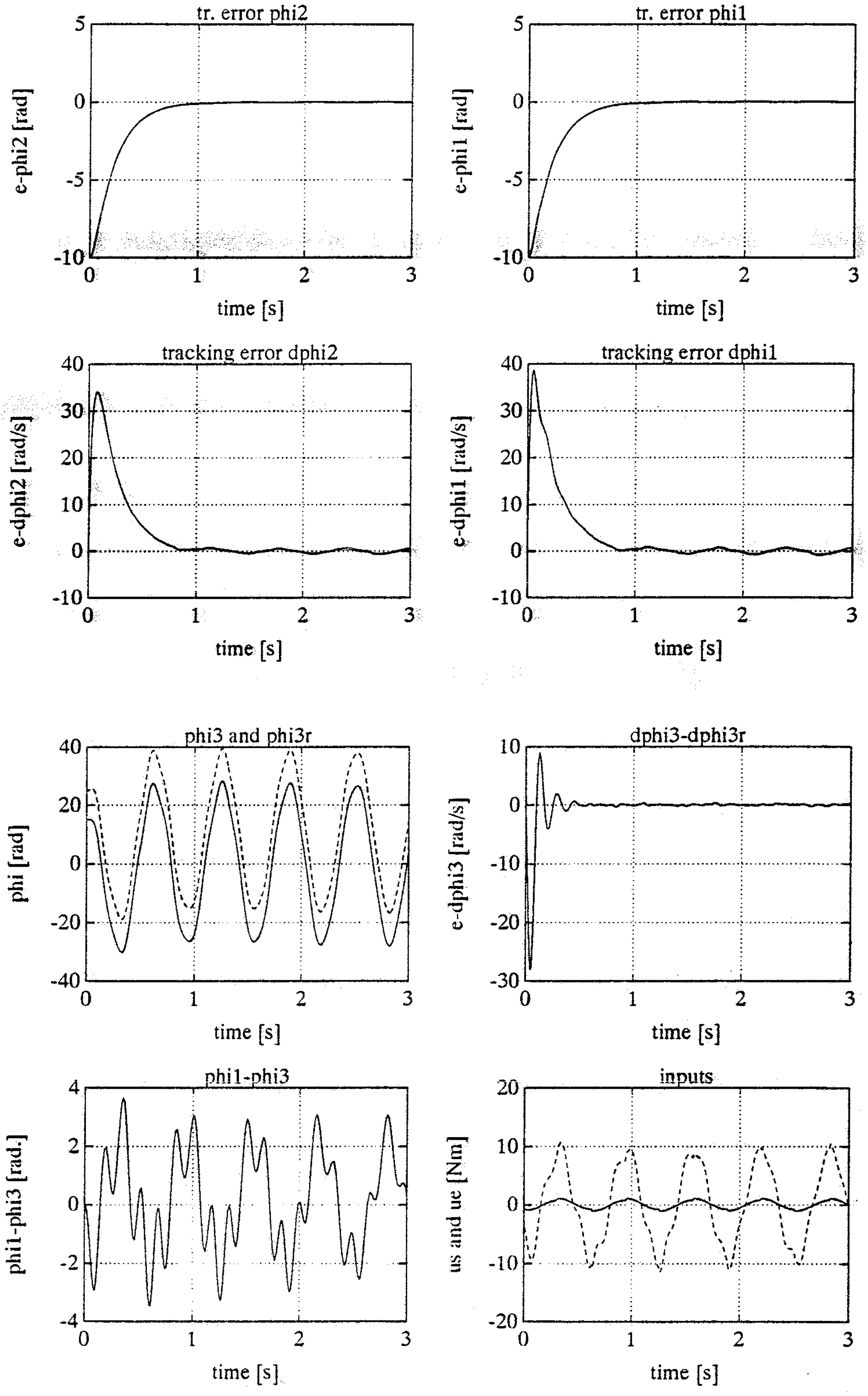
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.1, k_2= 0.01, k_3= 0.01, λ_1= 5 and λ_2= 5
integration accuracy: tol= 0.01, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 2



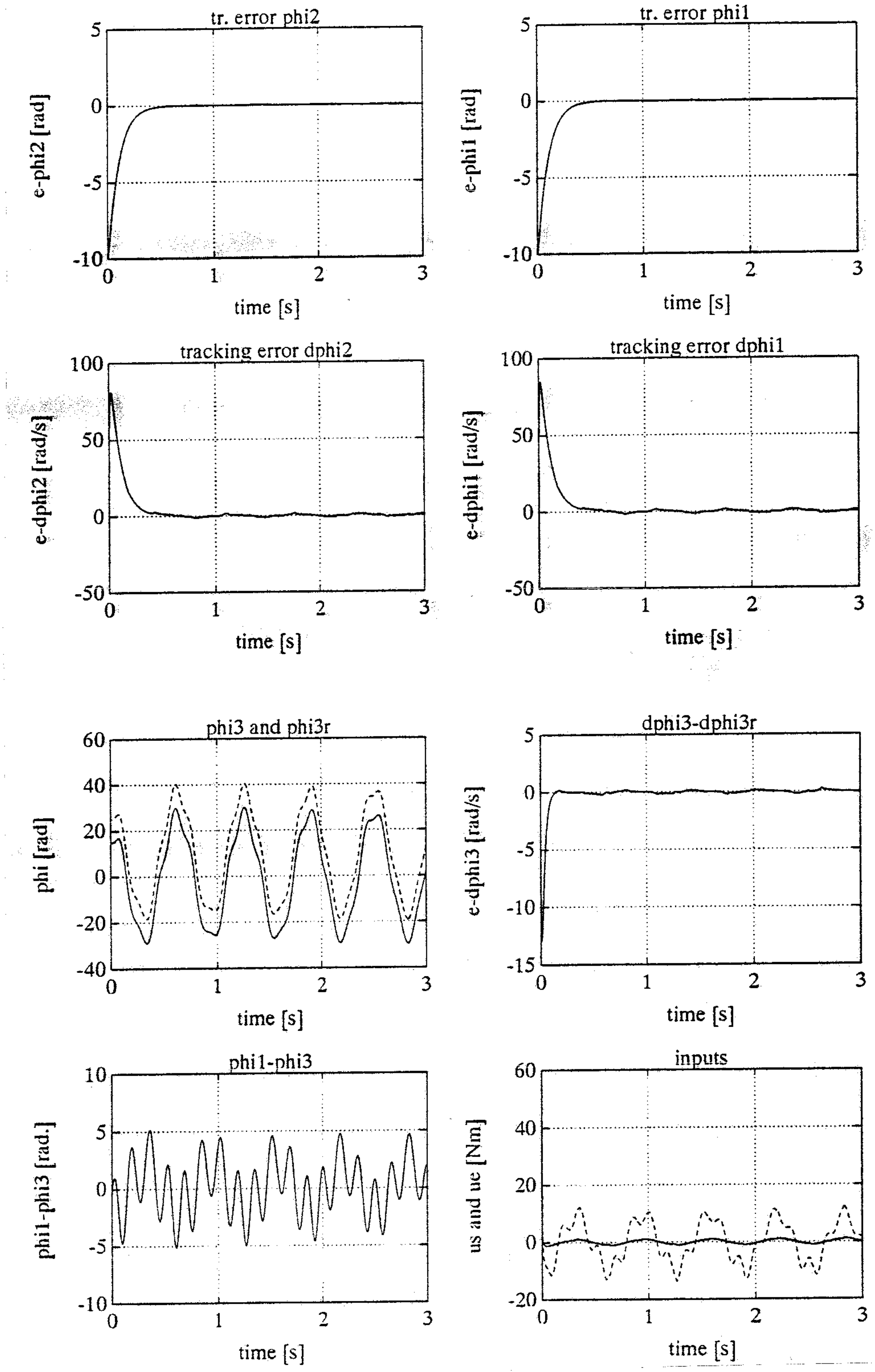
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.1, k_2= 0.01, k_3= 0.01, λ_1= 5 and λ_2= 5
integration accuracy: tol= 0.001, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 3



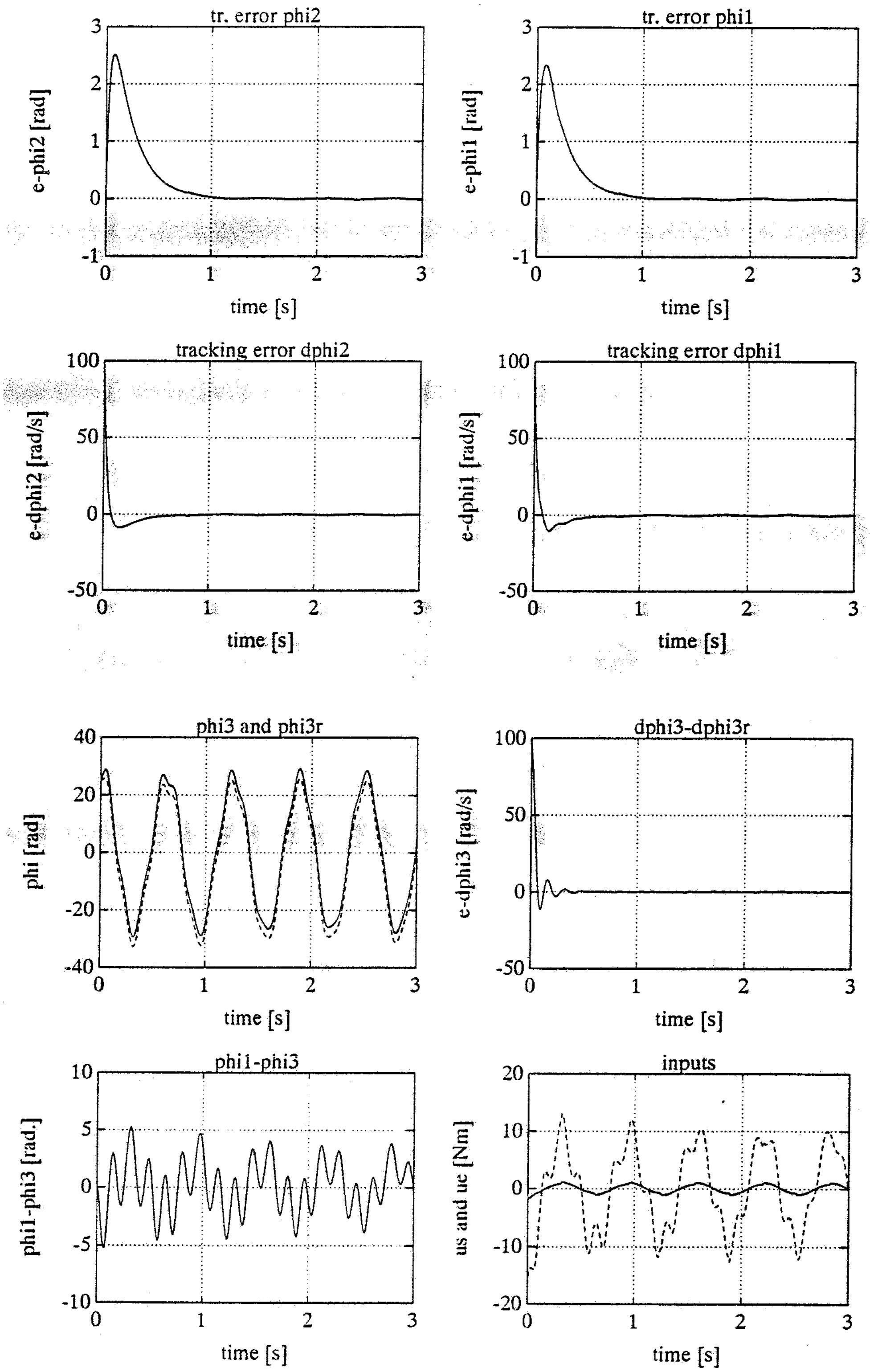
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.1, k_2= 0.01, k_3= 0.01, λ_1= 5 and λ_2= 5
integration accuracy: tol= 0.01, tracking error t=0: $\Delta\varphi_2(0)$= -10 rad, $\Delta\varphi_1(0)$= -10 rad, $\Delta\varphi_3(0)$= -10 rad
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 4



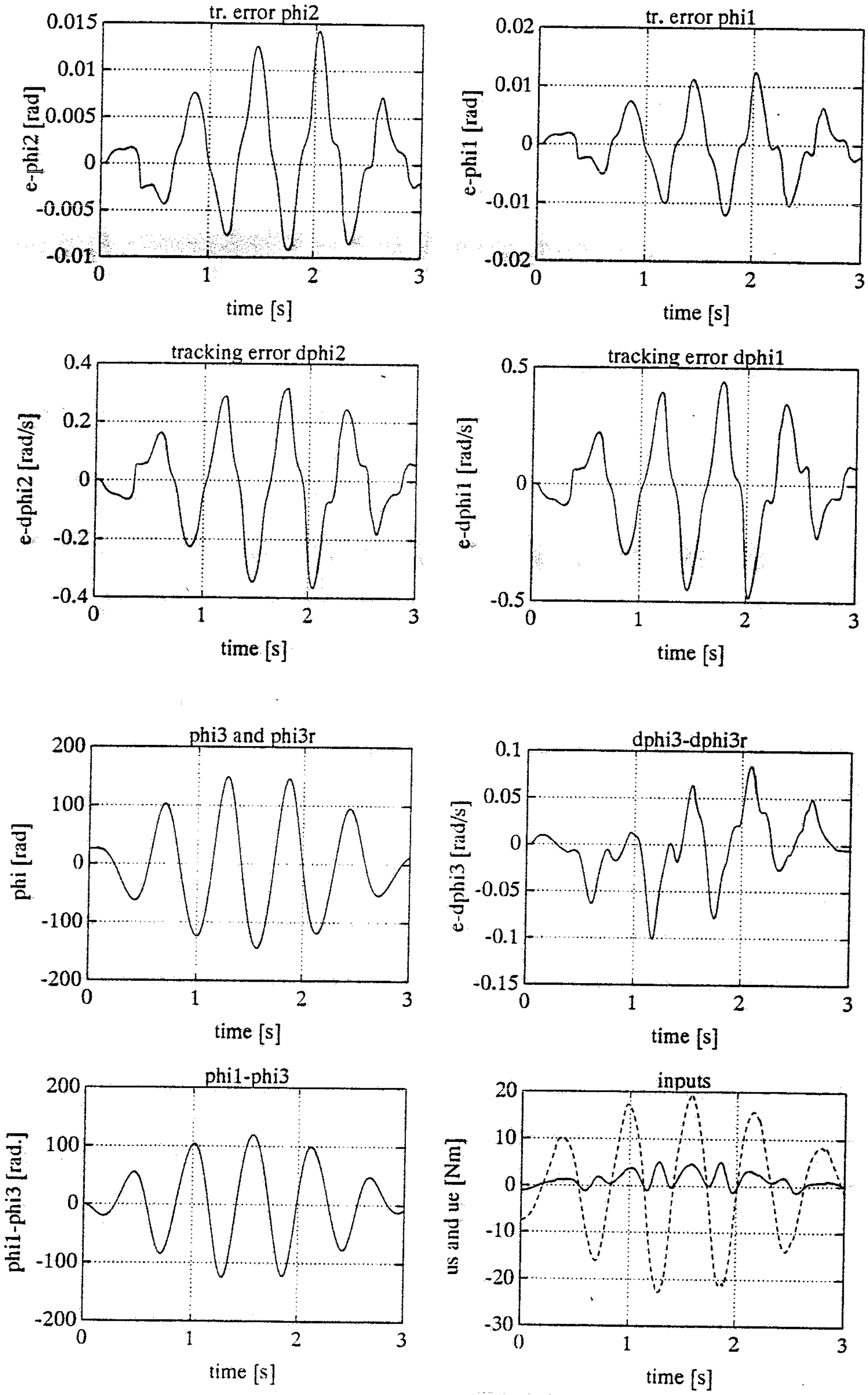
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
integration accuracy: tol= 0.01, tracking error t=0: $\Delta\varphi_2(0)$= 10 rad, $\Delta\varphi_1(0)$= 10 rad, $\Delta\varphi_3(0)$= 10 rad
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 5



desired trajectory: plot 0, k= 1 Nm/rad, control par.: 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: $\Delta\varphi_2(0)$= 100 rad/s, $\Delta\varphi1(0)$= 100 rad/s, $\Delta\varphi_3(0)$= 100 rad/s

situation 6



desired trajectory: plot 0, k= 0.1 Nm/rad, control par.: 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: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

APPENDIX C: SIMULATION RESULTS OF SOME PRACTICAL SITUATIONS OF THE FIRST PART OF THE RESEARCH

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: $\varphi_{2d}= \alpha_2+\alpha_2\cos(\omega_2 t)$, ...
$\varphi_{1d}= \alpha_1+\alpha_1\cos(\omega_1 t)$, ...
$\alpha_2= 25$ rad, $\omega_2= 1$ rad/s and $\alpha_1= 25$ rad, $\omega_1= 1$ rad/s

- spring constant: k= 1 Nm/rad

- control gain parameters: $k_1= 0.5$, $k_2= 0.05$, $k_3= 0.05$, $\lambda_1= 10$ and $\lambda_2= 10$

- integration accuracy: can not be chosen

- discretization: $\Delta t_r= 0.001$ s, $\Delta t_s= 0.005$ s ($\Delta t_m= 0.006$ s)

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

2) As situation 1 but now:

- $\omega_2= 10$ rad/s, $\omega_1= 10$ rad/s.

3) As situation 2 but now:

- discretization: $\Delta t_r= 0.002$ s, $\Delta t_s= 0.008$ s ($\Delta t_m= 0.01$ s)

4) As situation 2 but now:

- discretization: $\Delta t_r= 0.005$ s, $\Delta t_s= 0.005$ s ($\Delta t_m= 0.01$ s)

5) As situation 2 but now:

- discretization: Δt_r= 0.005 s, Δt_s= 0 s (Δt_m= 0.005 s)

6) As situation 2 but now:

- discretization: Δt_r= 0.005 s, Δt_s= 0.003 s (Δt_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: $\Delta\varphi_2$= -250 rad/s

9) As situation 2 but now:

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

10) As situation 2 but now:

- unmodelled dynamics: reality: w1= 0.47 Nm model: w1e= 0 Nm
  w2= 0.15 Nm w2e= 0 Nm

11) As situation 2 but now:

- wrong estimated parameters: reality: J1= 2.15 10^{-3} model: J1m= 1.075 10^{-3}
  J2= 1.45 10^{-4} J2m= 0.72 10^{-4}

12) As situation 2 but now:

- unmodelled dynamics: reality: k= 1 Nm/rad model: ke= 0.7 Nm/rad

13) As situation 2 but now:

- unmodelled dynamics: of situation 9,11 and 12.

14) As situation 3 but now:

- spring constant: k= 0.1 Nm/rad

15) As situation 3 but now:

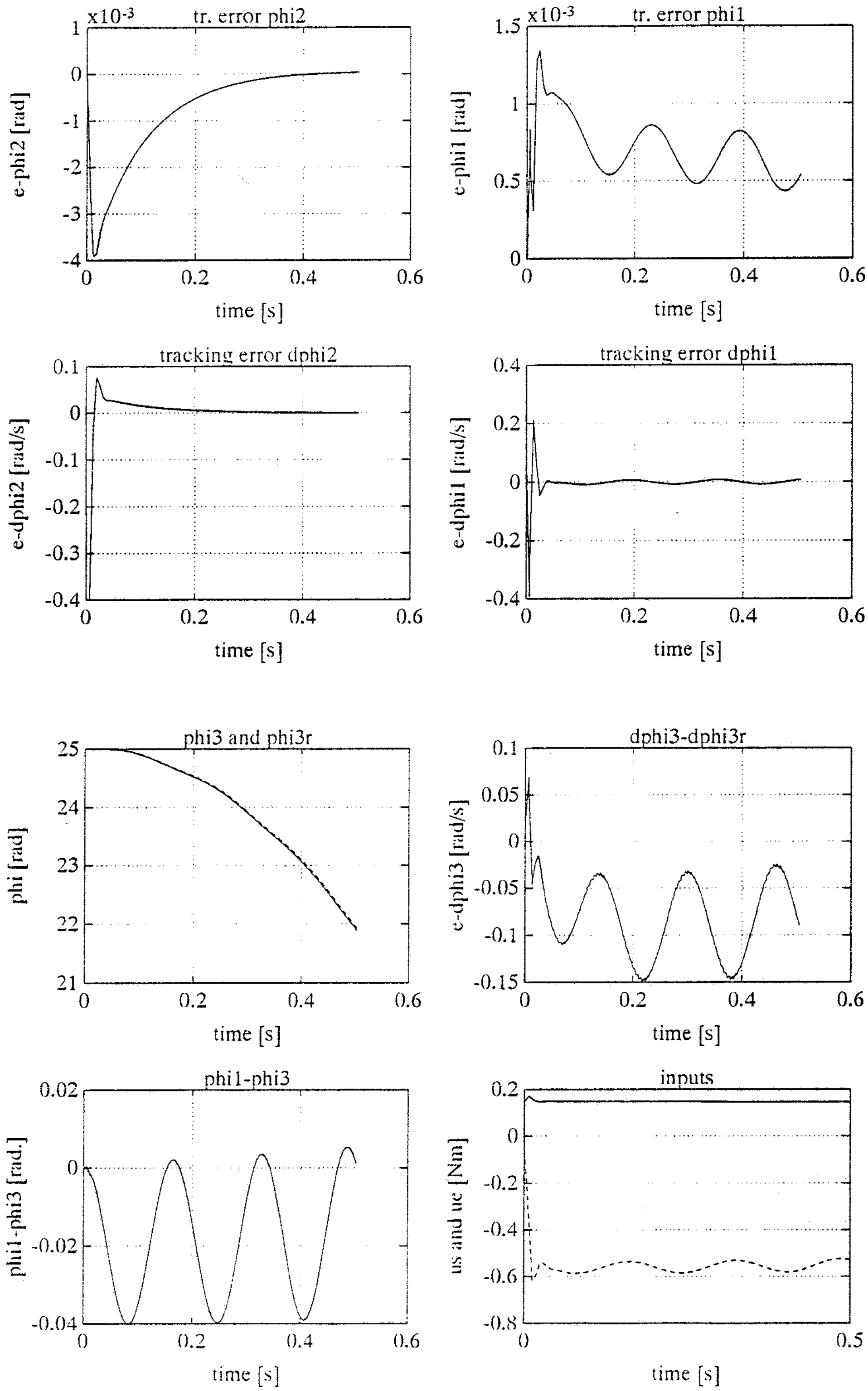
- measure errors: - angular position → white noise v_p= a_{vp}rand(t) → a_{vp}= 0.1
  - angular velocity → white noise v_v= a_{vv}rand(t) → a_{vv}= 1

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

Simulation results

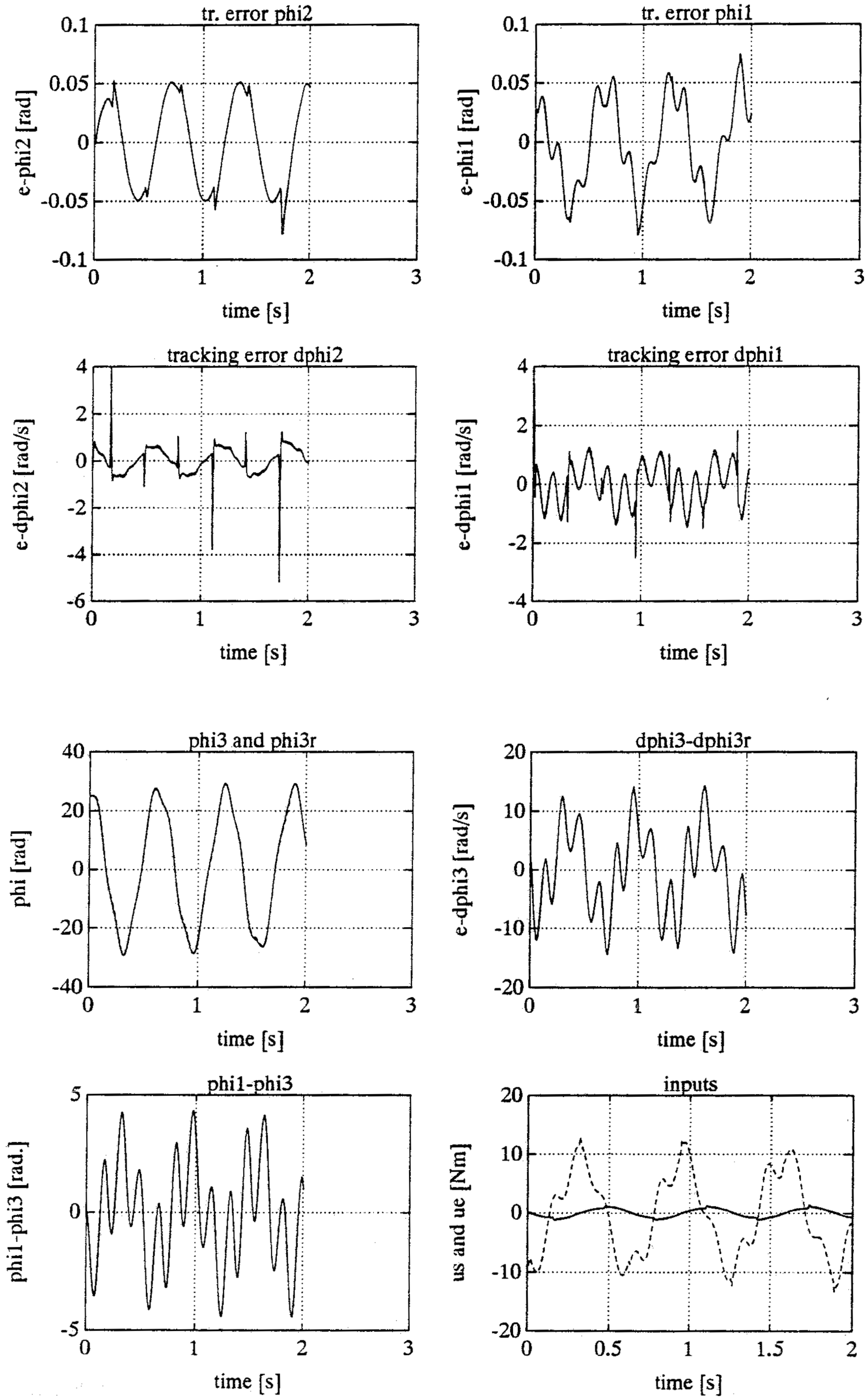
I have made the same plots of every situation as of the theoretical situation (see appendix B).
The structure of a page with plots is the same as given in apendix B.
Now I will give per situation the simulation results (see the following pages).

situation 1



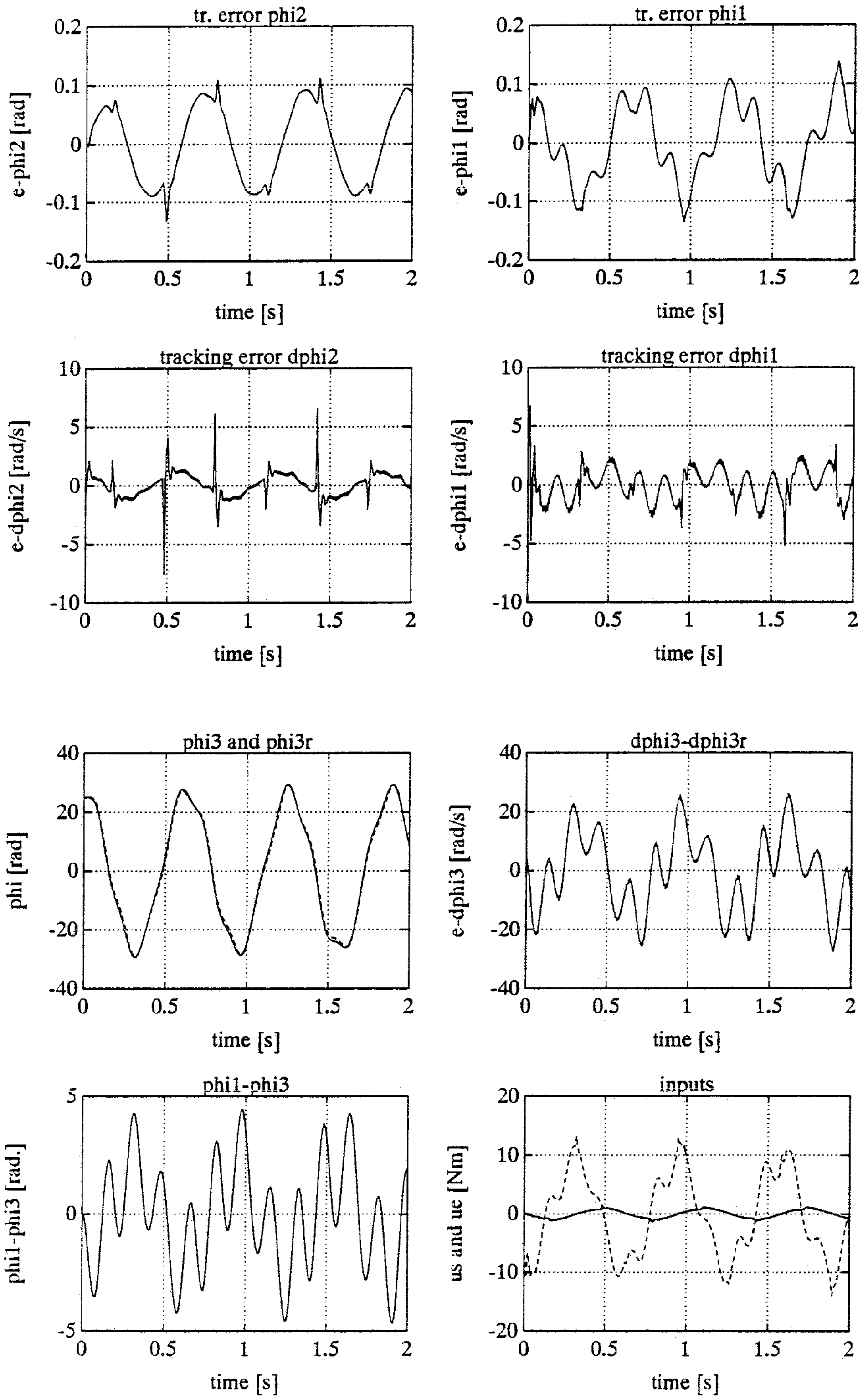
desired trajectory: plot 0 now: ω_1 =ω_2= 1 rad/s, k= 1 Nm/rad, con. par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 2



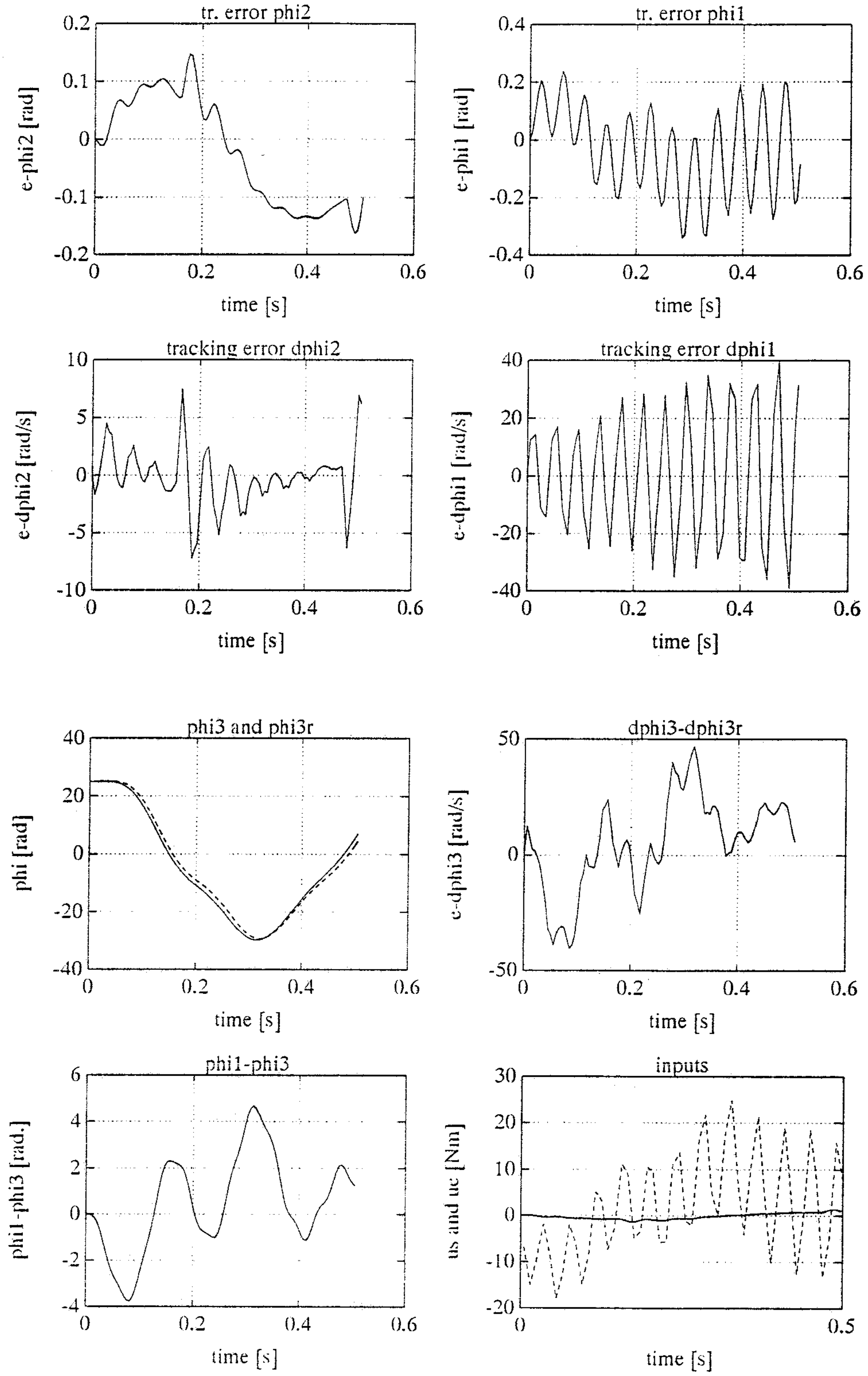
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 3



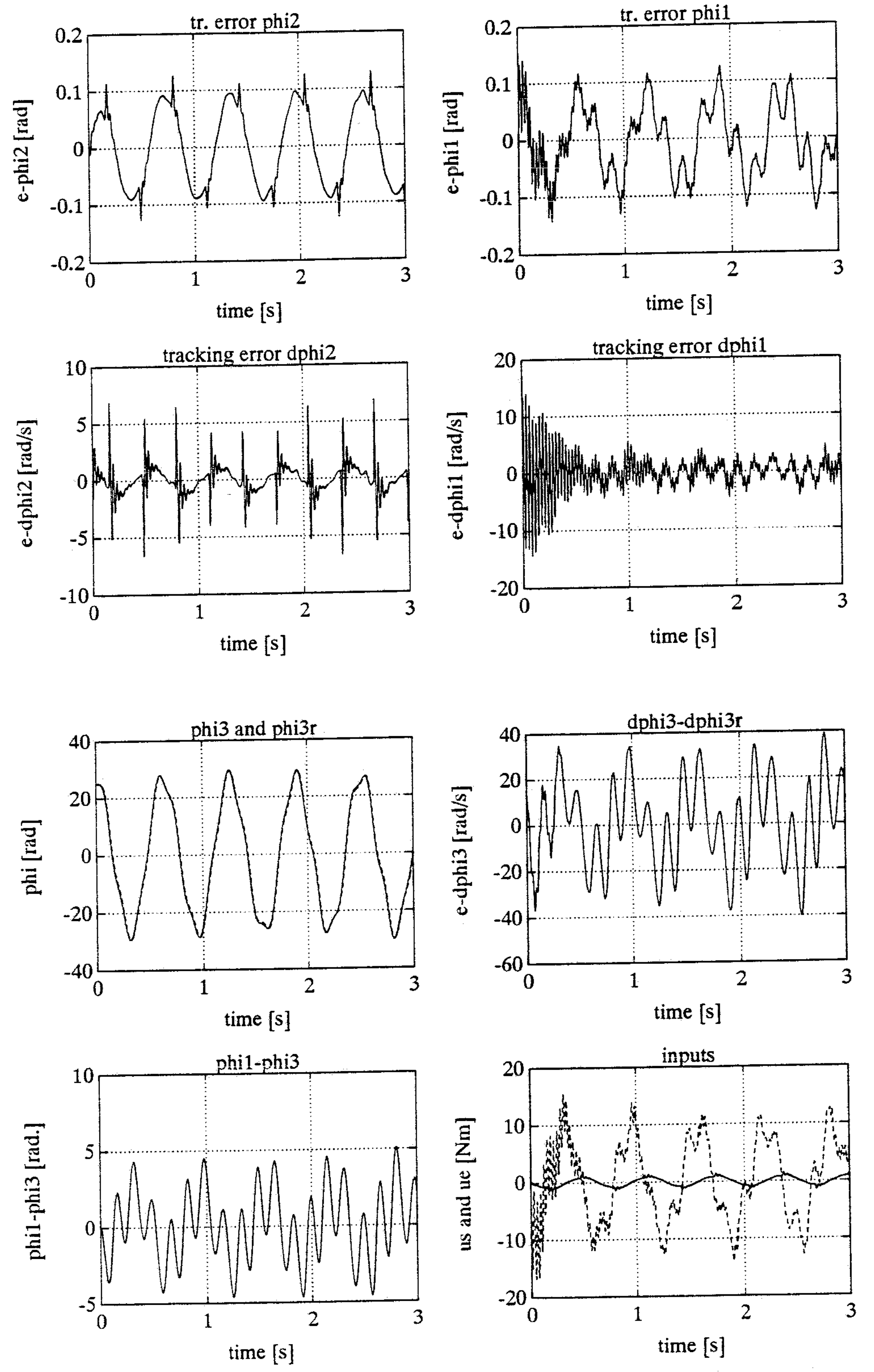
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.002 s, Δt_s= 0.008 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi$1(0)= 0, $\Delta\varphi_3(0)$= 0

situation 4



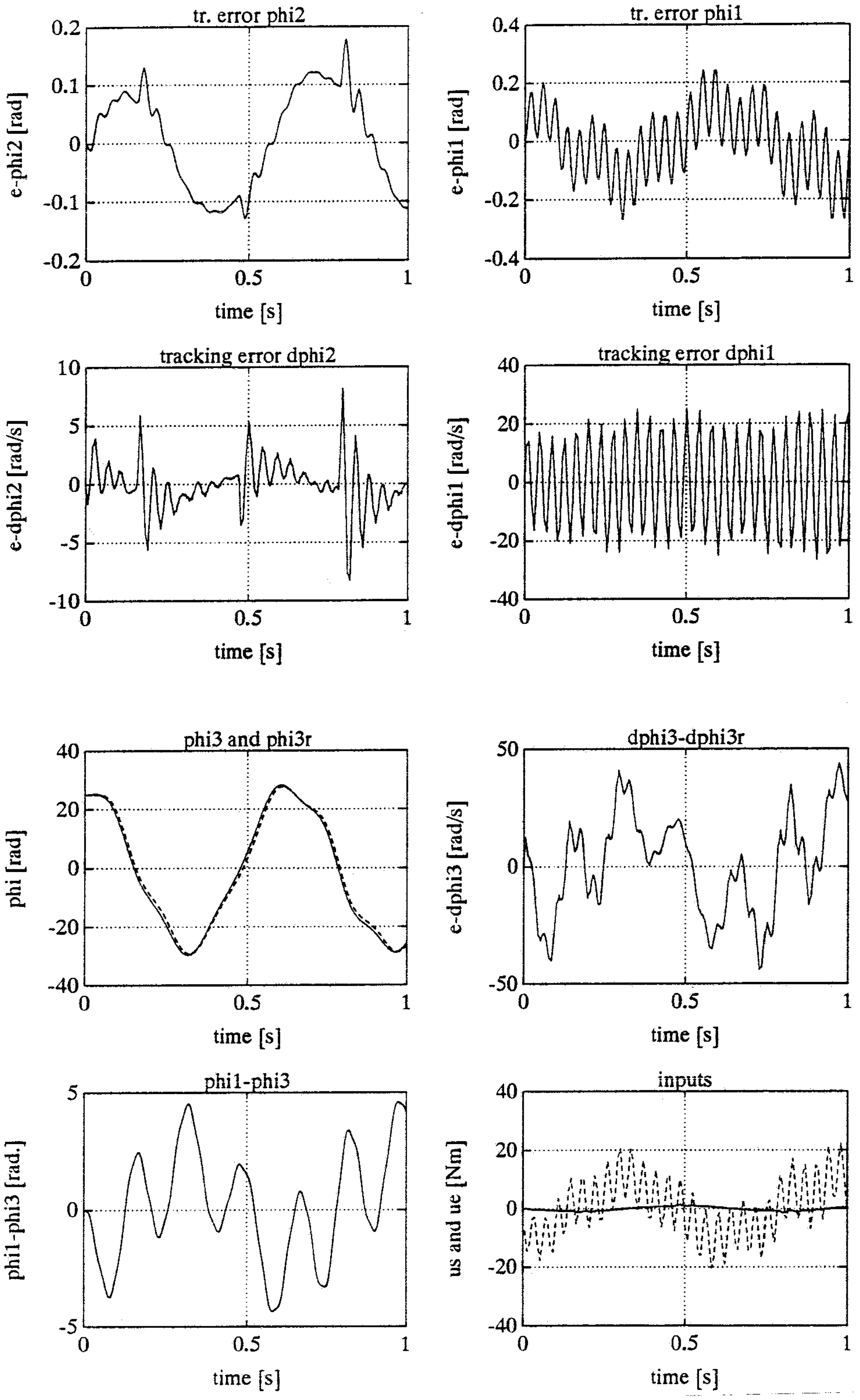
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.005 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 5



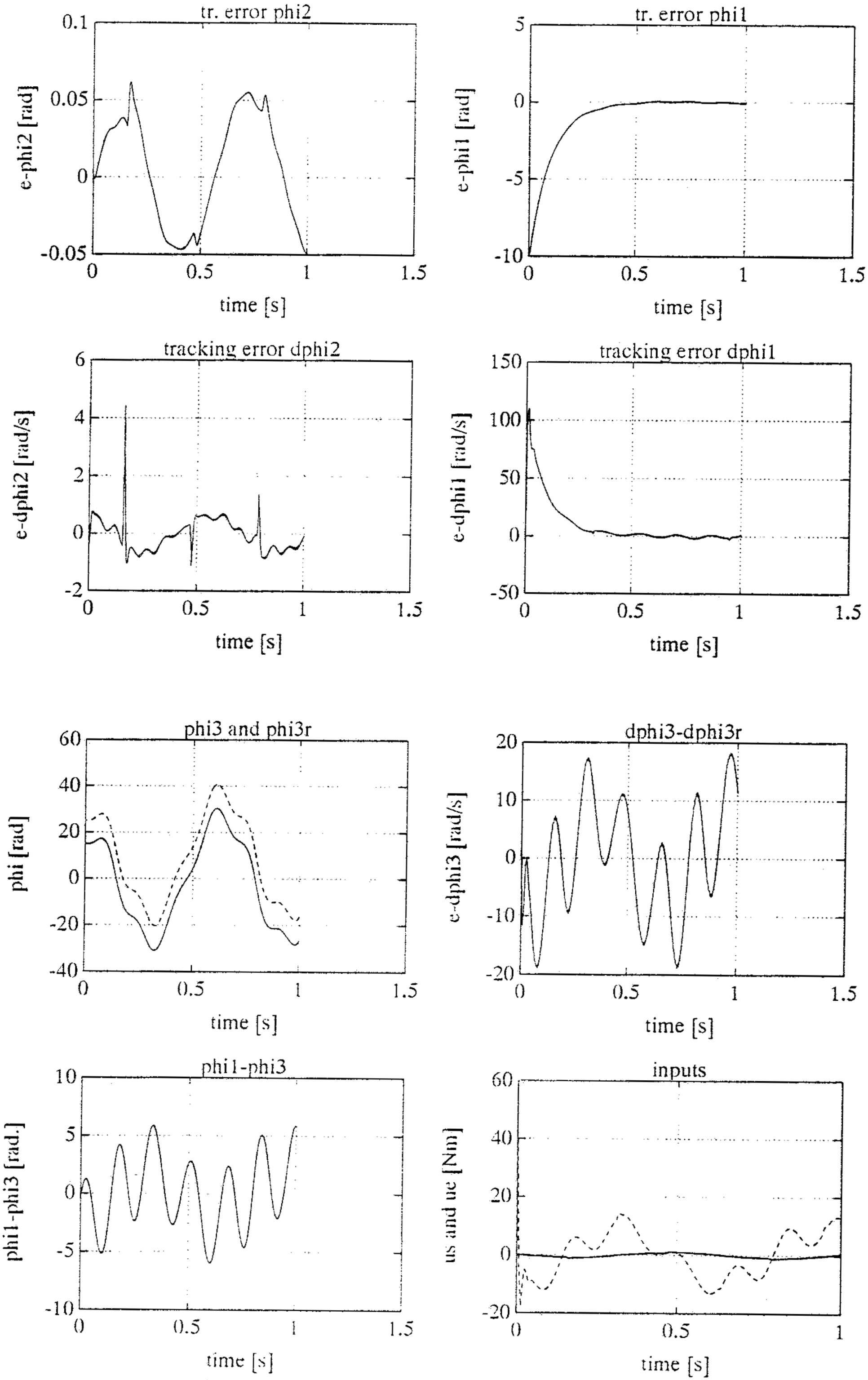
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.005 s, Δt_s= 0 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 6



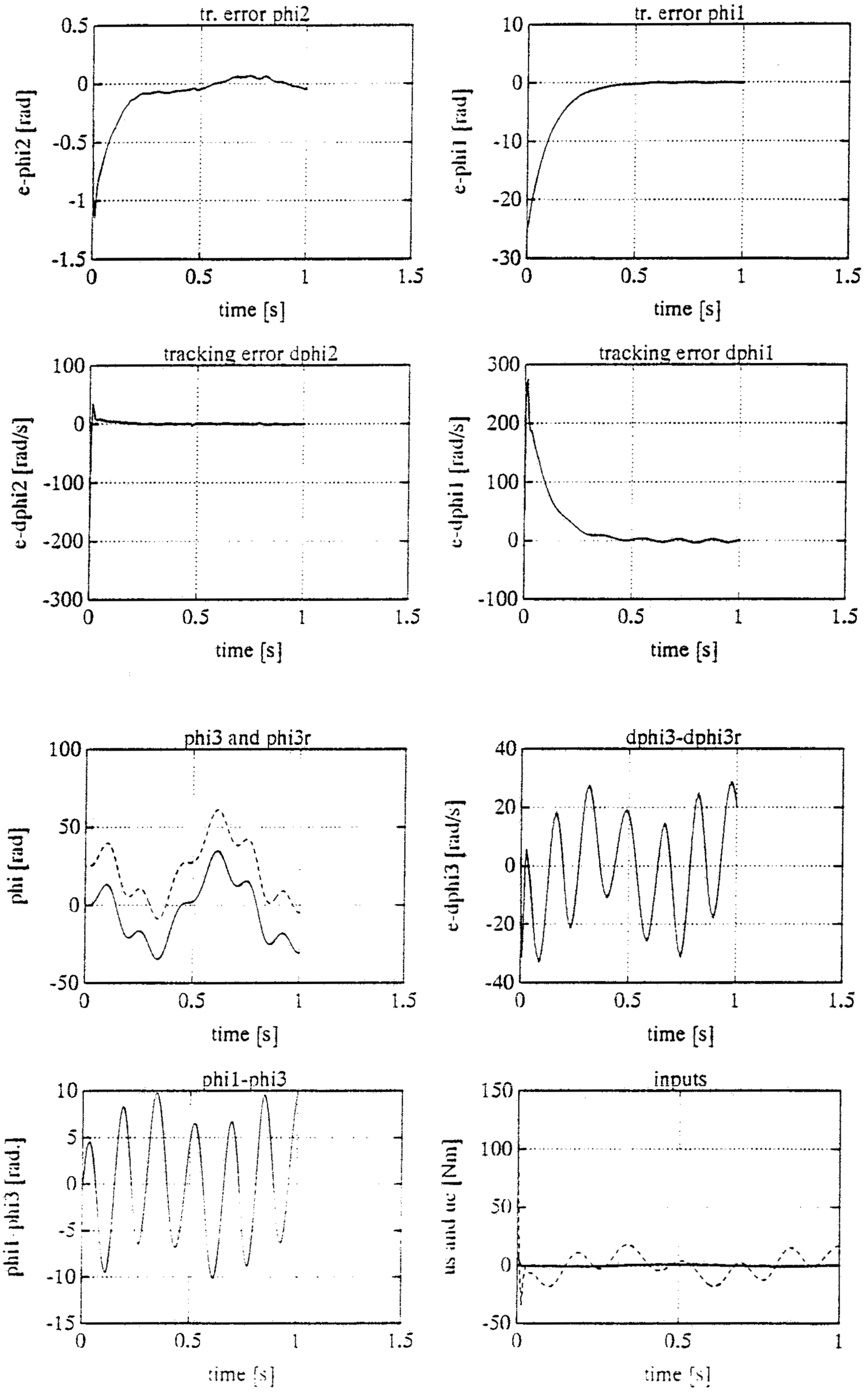
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.005 s, Δt_s= 0.003 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 7



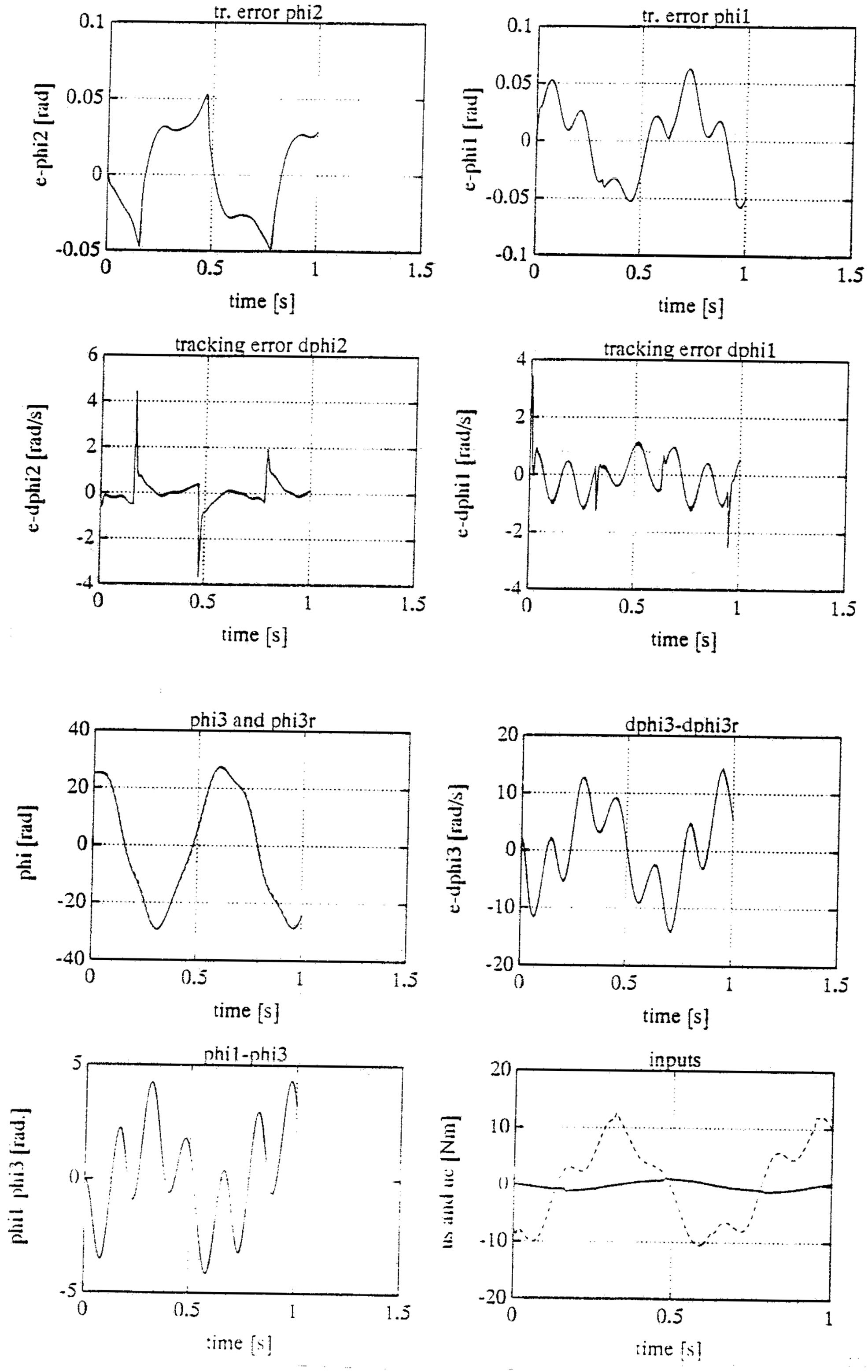
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= -10 rad, $\Delta\varphi_3(0)$= -10 rad
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 8



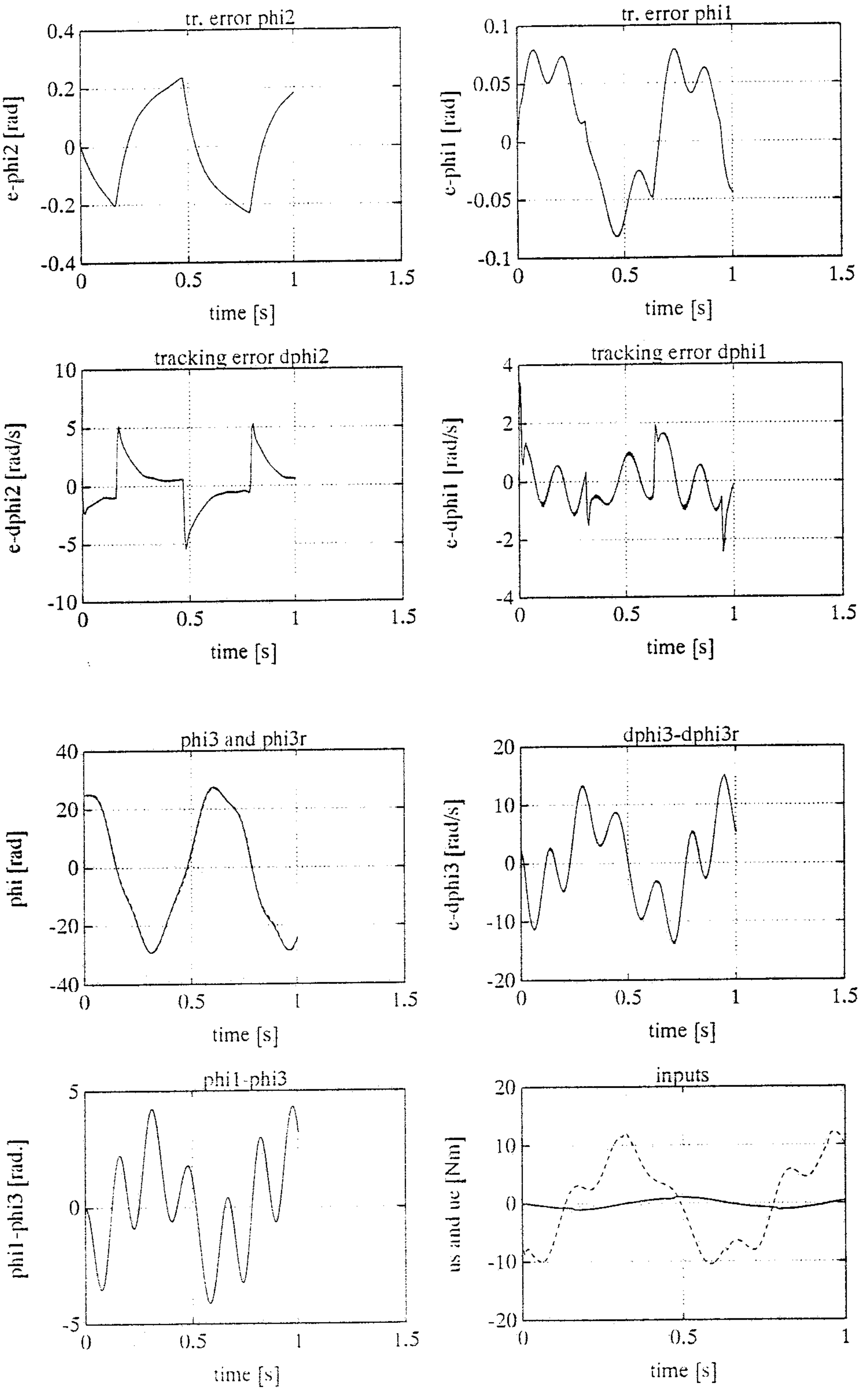
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= -25 rad, $\Delta\varphi_3(0)$= -25 rad
velocity error t=0: $\Delta\varphi_2(0)$= -250 rad/s, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 9



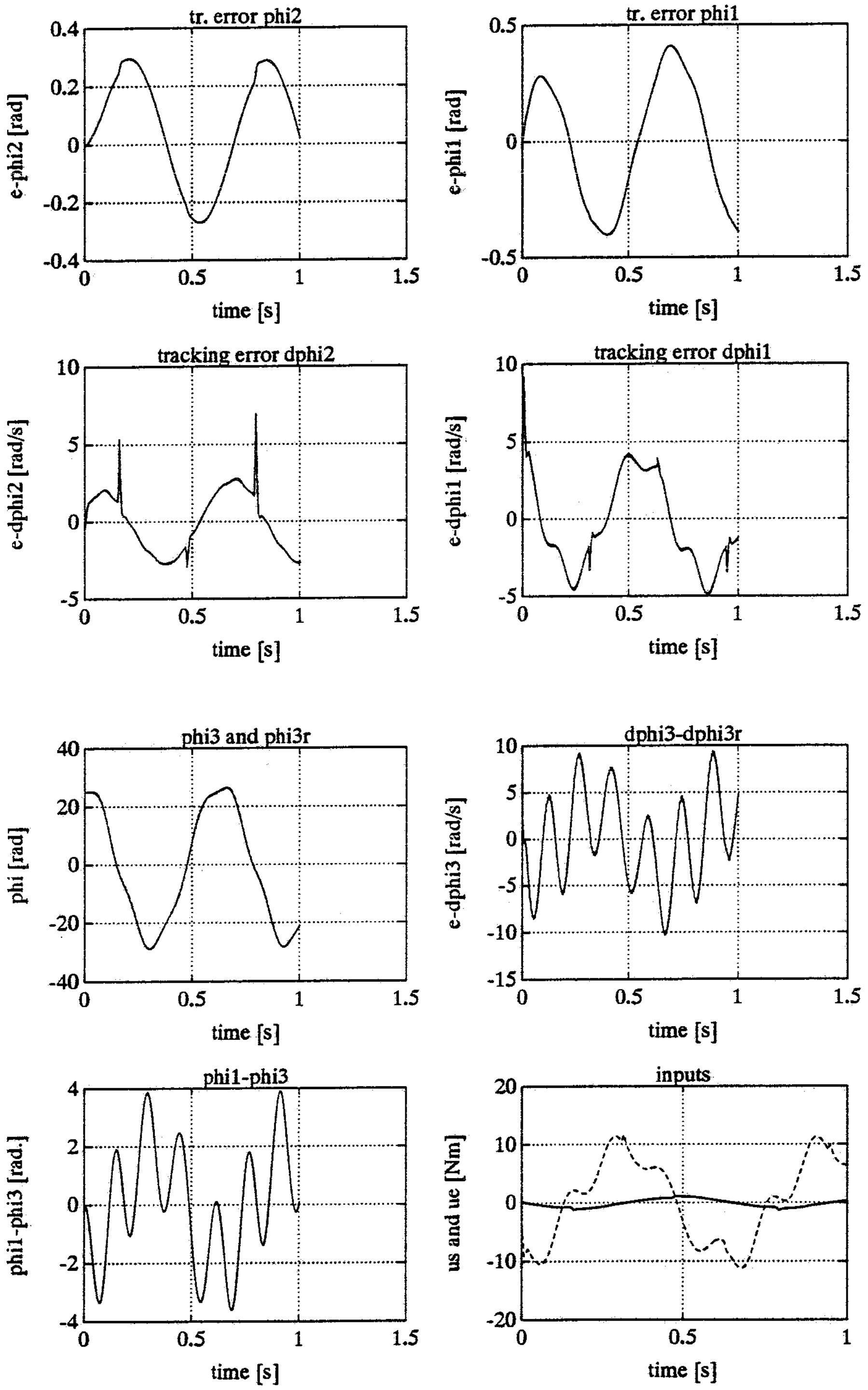
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0,
wrong estimated par.: (w_1= 0.47 Nm, w_{1e}= 0.3 Nm) (w_2= 0.15 Nm, w_{2e}= 0.1 Nm)

situation 10



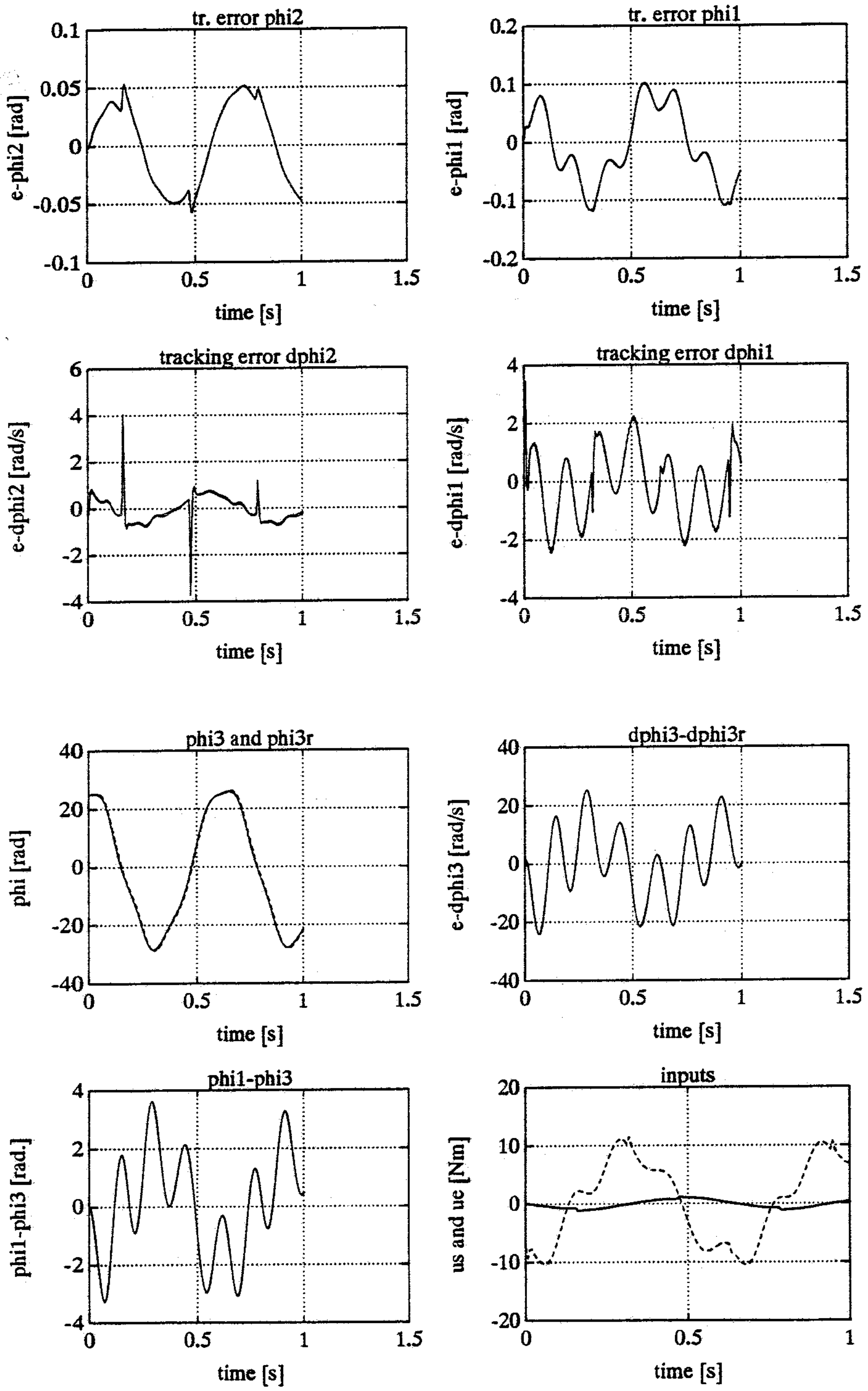
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par.: (w_1= 0.47 Nm, w_{1e}= 0 Nm) (w_2= 0.15 Nm, w_{2e}= 0 Nm)

situation 11



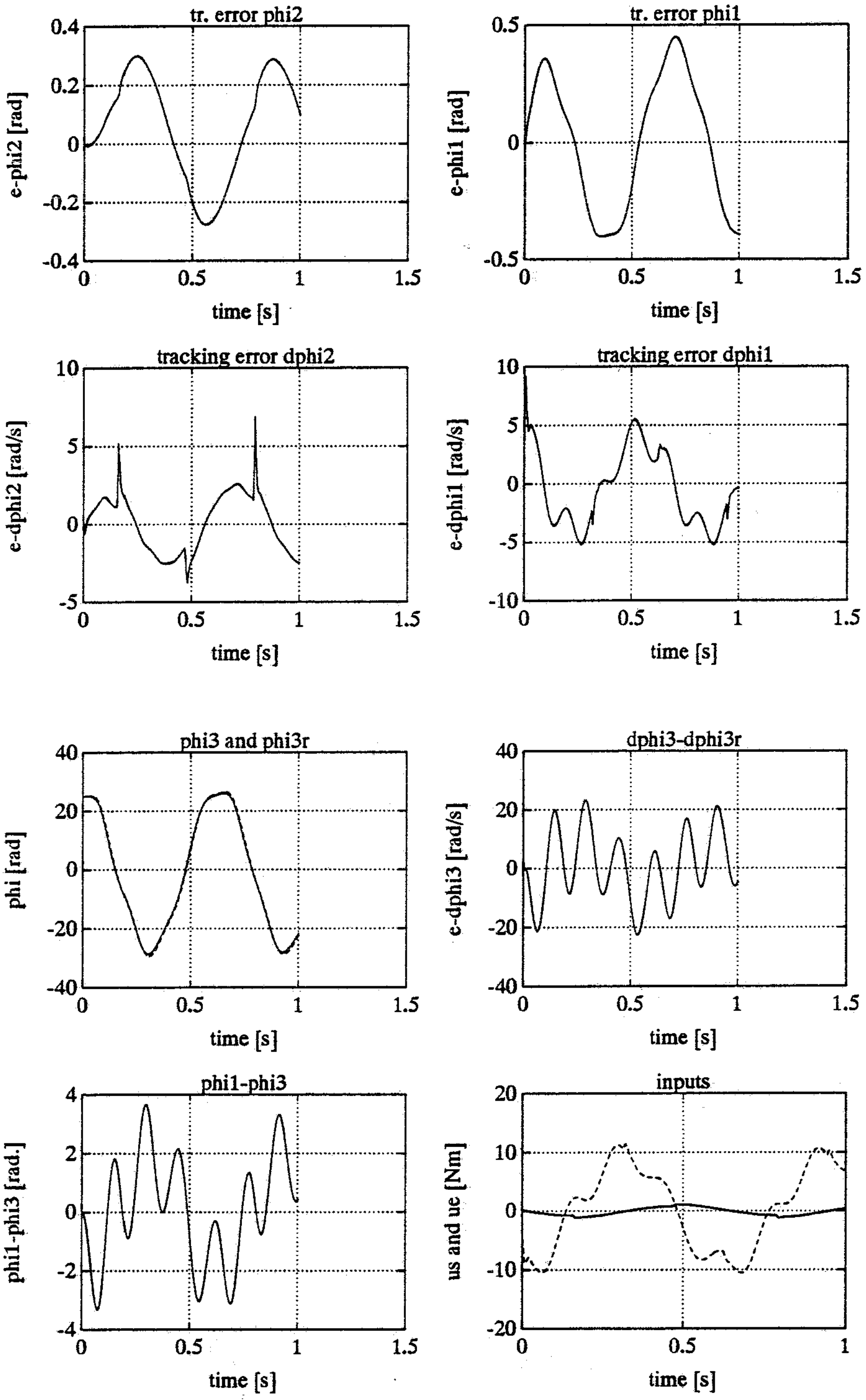
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : (J1= 2.15e-3 J1e= 1.075e-3) (J2= 1.45e-4 J2e= 0.72e-4)

situation 12



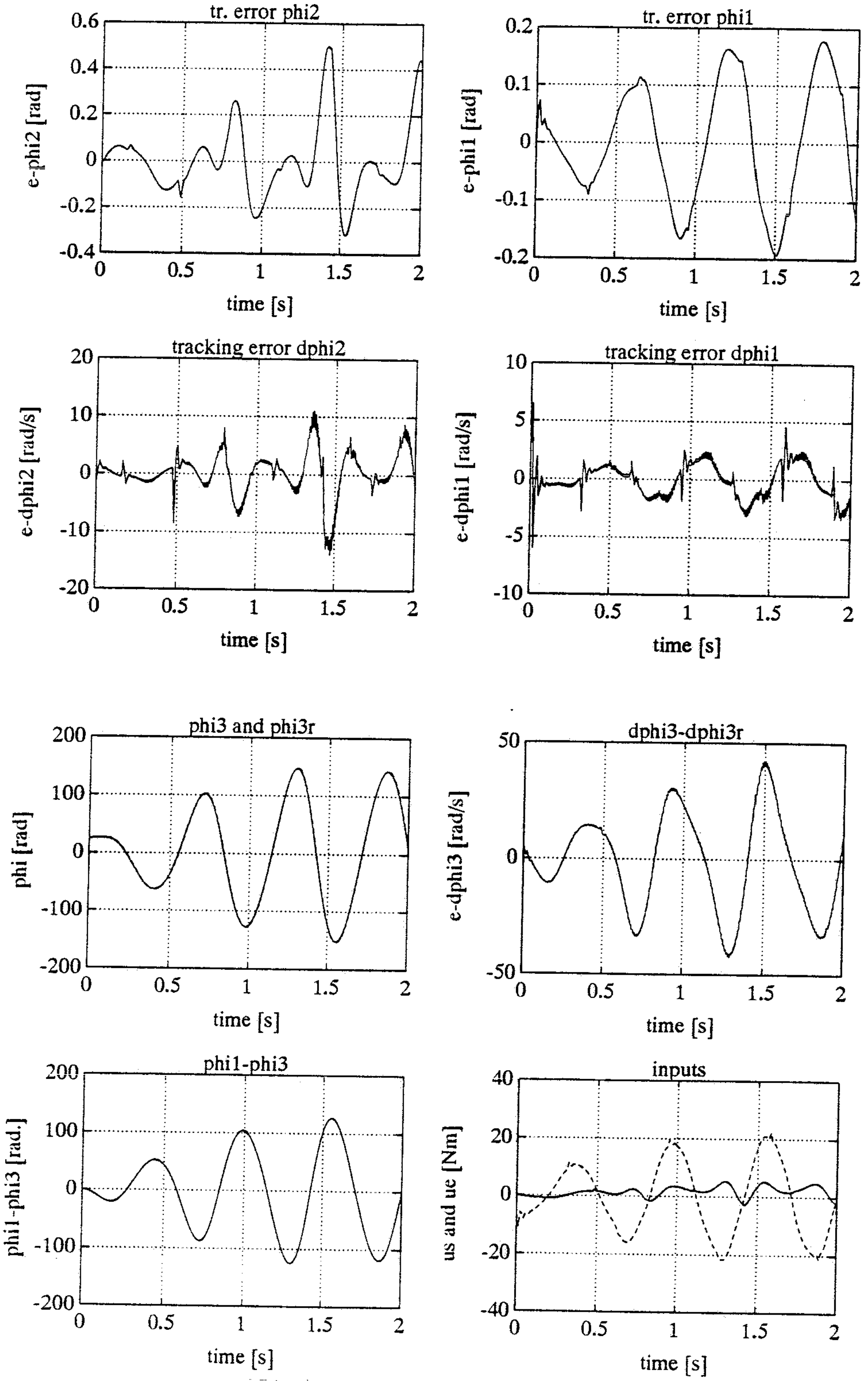
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : k= 1 Nm/rad k_e= 0.7 Nm/rad

situation 13



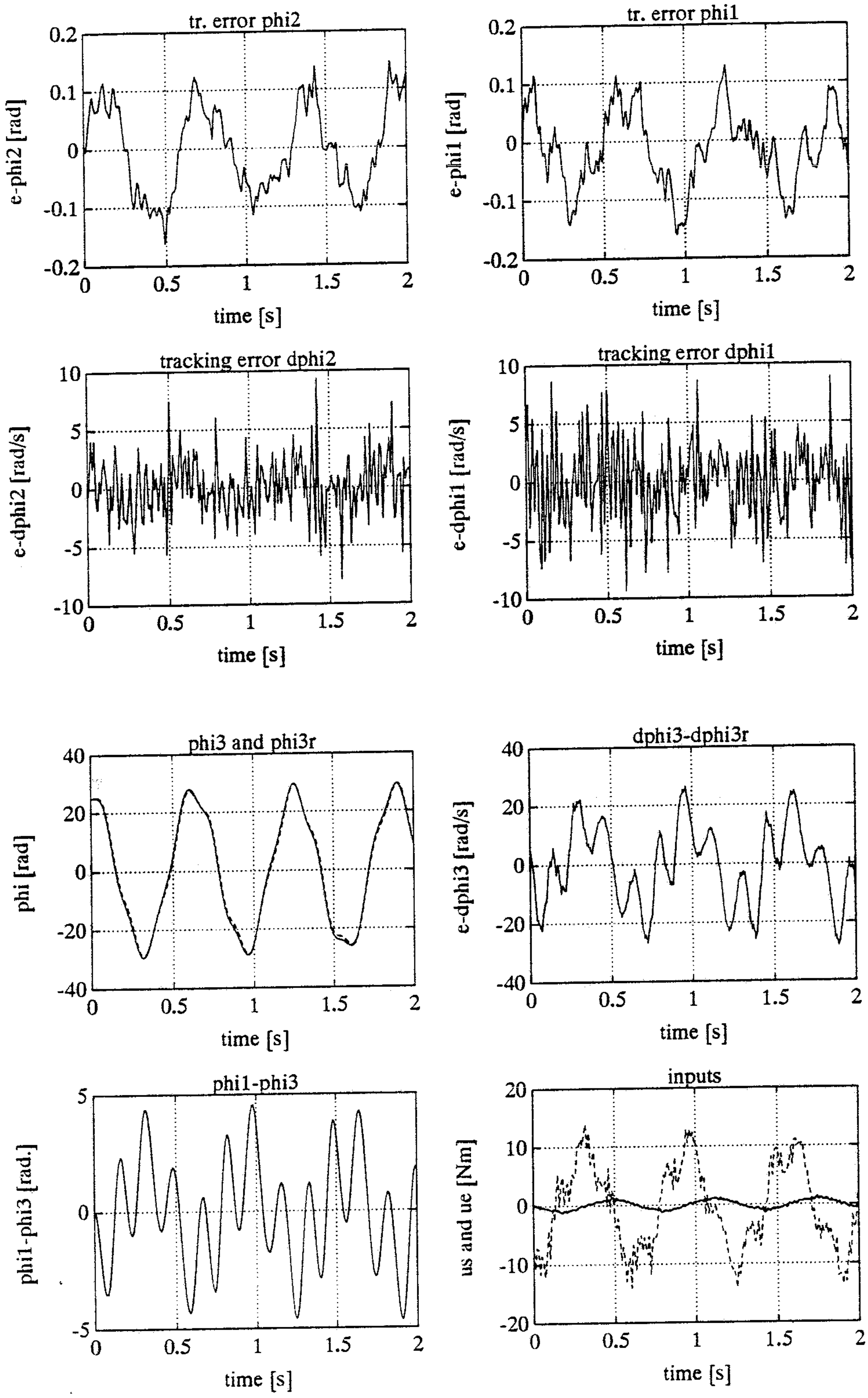
desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : w_{1e}= 0.3 Nm, w_{2e}= 0.1 Nm), J1e= 1.075e-3, J2e= 0.72e-4, k_e= 0.7 Nm/rad

situation 14



desired trajectory: plot 0, k= 0.1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.002 s, Δt_s= 0.008 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi$1(0)= 0, $\Delta\varphi_3(0)$= 0

situation 15



desired trajectory: plot 0, k= 1 Nm/rad, control par.: k_1= 0.5, k_2= 0.05, k_3= 0.05, λ_1= 10 and λ_2= 10
dicretization: Δt_r= 0.001 s, Δt_s= 0.005 s, tracking error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity error t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
measure noise: a_{vp}= 0.1, a_{vv}= 1

APPENDIX D: THE OBSERVER ALGORITHM

For the designing of the observer I use the theory which is presented in Regelen 2 (Kok,1990).

The system behaviour has been written in the state space form:

$$\underline{\dot{x}}(t) = A\underline{x}(t)+B\underline{u}_s(t) \tag{D.1}$$

$\underline{x}$= state
$\underline{u}_s$= inputs
A, B= system matrices

$$\begin{aligned}\underline{y}(t) &= C\underline{x}(t)\\ \underline{m}(t) &= C\underline{x}(t)+\underline{v}(t)\end{aligned} \tag{D.2}$$

$\underline{y}$= output
$\underline{m}$= measurements of the output
C= system matrix
$\underline{v}$= measure noise

Out of the measurements $\underline{m}$ and the known (or estimated) systembehaviour (D.1) it is possible to reconstruct the state $\underline{x}$. From chapter 5.1 of Regelen 2 it follows:

$$\underline{\dot{x}}_{rec}(t) = (A-KC)\underline{x}_{rec}(t)+B\underline{u}_s(t)+K\underline{m}(t) \tag{D.3}$$

x_{rec}= reconstructed state $\underline{x}$
K= an arbitrary matrix

For the reconstruction error $\underline{e}$= $\underline{x}$-$\underline{x}_{rec}$ it follows:

$$\underline{\dot{e}}(t) = (A-KC)\underline{e}(t)+\underline{n}(t) \tag{D.4}$$

$\underline{n}$= vector with (small) values due to noise

There are several ways to chose the matrix K. One of the possibilities is pole-placement. We have to chose K in such a way that the poles of (A-KC) have certain optained values (e.g. have negative real values). This pole-placement routine can be executed with a Matlab program (K is calculated).
An other possibility is the optimal observer or Kalman-Bucy filter. For the execution of this routine we have to know something about the measure - and system noise. The Kalman-Bucy routine (see chapter 5.2 regelen 2) can also be executed with a Matlab program (K is calculated). Now that I have define a general method for the reconstruction, I can work out this method for the case of the xy table.

The reconstruction of the angular velocities of the xy table

First I have to define the state space form. I use equation (3.1) with dry friction:

$$M\ddot{\underline{q}}+C\dot{\underline{q}}+K_k\underline{q}+\underline{n}_d = H\underline{u} \qquad \text{(D.5)}$$

$\underline{q}=[\varphi_2, \varphi_1, \varphi_3]^T$
M,C and $\underline{n}_d$ are (partial) depending on $\underline{q}$ and or $\dot{\underline{q}}$

I split up equation (D.5) into the following way:

$$M_c\ddot{\underline{q}}+M_v\ddot{\underline{q}}+C_v\dot{\underline{q}}+K_k\underline{q}+\underline{n}_d = H\underline{u} \qquad \text{(D.6)}$$

M_c= exists of the constant terms of M
M_v= exists of the variable terms of M (terms which are depending on $\underline{q}$ and $\underline{q}$)
C_v= C= exists only of variable terms

I will explain the reason of splitting up equation (D.5) later on in this appendix.

After rewriting (D.6) it follows:

$$\ddot{\underline{q}} = -M_c^{-1}K_k\underline{q}+M_c^{-1}(H\underline{u}-M_v\ddot{\underline{q}}-C_v\dot{\underline{q}}-\underline{n}_d) \qquad \text{(D.7)}$$

Out of (D.7) the state space form (D.1) can be determined. I define:

$\underline{x}= [\dot{\varphi}_2, \varphi_2, \dot{\varphi}_1, \varphi_1, \dot{\varphi}_3, \varphi_3]^T$
$E= -M_c^{-1}K_k$

It follows that (see D.1):

$$A = \begin{bmatrix} 0 & E(1,1) & 0 & E(1,2) & 0 & E(1,3) \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & E(2,1) & 0 & E(2,2) & 0 & E(2,3) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & E(3,1) & 0 & E(3,2) & 0 & E(3,3) \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{D.8}$$

$$B = \begin{bmatrix} M_c(1,1) & M_c(1,2) & M_c(1,3) \\ 0 & 0 & 0 \\ M_c(2,1) & M_c(2,2) & M_c(2,3) \\ 0 & 0 & 0 \\ M_c(3,1) & M_c(3,2) & M_c(3,3) \\ 0 & 0 & 0 \end{bmatrix} \tag{D.9}$$

$$\underline{u}_s = (H\underline{u}-M_v\underline{\ddot{q}}-C_v\underline{\dot{q}}-\underline{n}_d) \tag{D.10}$$

Now I consider the output- and the measure equation (D.2). I define:

$y= [\varphi_2, \varphi_1, \varphi_3]^T$, $m= [\varphi_2+n_2, \varphi_1+n_1, \varphi_3+n_3]^T$

It follows:

$$C=\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{D.11}$$

Now it is possible with a pole placement routine or a Kalman-Bucy filter to determine a K matrix. Because of the fact that A,B and C are constant matrices, the determination can be done before the execution of experiments, and has been done once.
If we had not split up equation (D.5) the system matrices A and B were not constant matrices. In that case the matrix K has been calculated during the execution of the experiments. The on-line calculation of K can cause problems, this because of the fact that the calculation of K costs much computing time (if the computing time grows too big, the system can't calculate all the variables within one timestep).

The last step of the designing of the observer is the dicretization of equation (D.3):

$$\underline{x}_{rec_{i+1}} = \underline{x}_{rec_i} + \Delta t((A-KC)\underline{x}_{rec_i} + B\underline{u}_{s_i} + K\underline{m}_i) \tag{D.12}$$

Equation (D.12) can be implemented in the control system of the xy table. With (D.12) it is possible to reconstruct the state $\underline{x}=[\ \dot{\varphi}_2, \varphi_2, \dot{\varphi}_1, \varphi_1, \dot{\varphi}_3, \varphi_3]^T$.

During the research I have try first to execute the Kalman-Bucy routine for the calculation of K. The problem of executing this routine is the fact that the measure - and system noise is not exactly known. The reconstruction results are reasonable but not good enough.
However, if we execute the pole-placement routine, we can find better results (smaller reconstruction errors). If we choose the poles of (A-KC) with negative real values of 25 we can obtain good reconstruction results.

Now I have define a way for the reconstruction of the angular velocities. With equation (D.12) I can execute the reconstruction, and with a pole-placement routine I can calculate a suitable K matrix.

APPENDIX E: SIMULATION - AND EXPERIMENTAL RESULTS OF THE SECOND PART OF THE RESEARCH

In this appendix I will show some simulation - and experimental results of the second part of the research (see chapter 5).

Situations

I have considered the next desired trajectories:

A)
- φ_{2d}= $\alpha\sin(\omega t)$; φ_{2d}= $\alpha\omega\cos(\omega t)$; ..
- φ_{1d}= $\alpha\sin(\omega t)$; φ_{1d}= $\alpha\omega\cos(\omega t)$; ..

B)
- φ_{2d}= $\alpha\sin(\omega t)$; φ_{2d}= $\alpha\omega\cos(\omega t)$; ..
- φ_{1d}= $\alpha\cos(\omega t)$; φ_{1d}= $-\alpha\omega\sin(\omega t)$; ..

C)
- φ_{2d}= $\alpha\sin(\omega_2 t)$; φ_{2d}= $\alpha\omega_2\cos(\omega_2 t)$; ..
- φ_{1d}= $\alpha\sin(\omega_1 t)$; φ_{1d}= $\alpha\omega_1\cos(\omega_1 t)$; ..

The reference trajectories are showed on page E.5 - E.7 (1 sec. = 100 timesteps)

I have considered the next situations:

1
- simulation
- desired trajectory: A, α= 25 rad ω= 3.14 rad/s
- springconstant: k = 0.5 Nm/rad
- control gain parameters: k_1= k_2= k_3= 0.05, lab_1= lab_2= 25
- observer-poles (used by the pole-placement routine): -25+0.2i
  -25+0.3i
  -25+0.4i
- measure frequence: 1/0.01 Herz (1 sec. = 100 timesteps)
- execution time: 2 s
- position t=0: φ_1= φ_2= φ_3= 0 (middle of the xy table)
- velocity t=0: φ_1= φ_2= φ_3= 0
- particularies: none

2) As situation 1 but now:

- experiment
- execution time: 3.5 s

3) As situation 2 but now:

- control gain parameters k_1= k_2= k_3= 0.05, lab_1= 50, lab_2= 25
- observer-poles: -50+0.2i, -50+0.3i, -50+0.4i
- measure frequency: 1/0.007 Herz (1 sec. = 142 timesteps)

4) As situation 2) but now:

- trajectory A, α= 25 rad ω= 1.57 rad/s
- measure frequency: 1/0.007 Herz (1 sec. = 142 timesteps)

5) As situation 1) but now:

- trajectory B, α= 25 rad ω= 3.14 rad/s
- execution time: 3.5 s

6) As situation 5) but now:

- experiment

7) As situation 1) but now:

- trajectory C, α= 25 rad ω_1= 3.14 rad/s, ω_2= 1.57 rad/s
- execution time: 3.5 s

8) As situation 7) but now:

- experiment
- control gain parameters k_1= k_2= k_3= 0.05, lab_1= 50, lab_2= 25
- observer-poles: -50+0.2i, -50+0.3i, -50+0.4i
- measure frequency: 1/0.007 Herz (1 sec. = 142 timesteps)

9) As situation 1) but now:

- springconstant: k=0.05

10) As situation 1) but now:

- springconstant: k=1

11) As situation 1) but now:

We don't use the C CTC law but the general CTC law. This control law can be used to control the xy table with a stiff transmission.

The CTC law:

$$
\begin{aligned}
u_s &= m_2\ddot{\varphi}_{2r}+n_{d22}+k_2\dot{e}_{2r}\\
u_e &= m_1\ddot{\varphi}_{1r}+n_{d11}+k_1\dot{e}_{1r}
\end{aligned}
$$

12) As situation 11) but now:

- trajectory B, α= 25 rad ω= 3.14 rad/s

13) As situation 5) but now:

- wrong estimated parameter: reality : k=0.5 Nm/rad (used in the simulator)
  (see also chapter 4) estimated parameter: k_e=0.1 Nm/rad
  (used in the control law)

14) As situation 5) but now:

- wrong estimated parameter: reality: k=0.5 Nm/rad
  estimated parameter: k_e=1 Nm/rad

15) As situation 5) but now:

- wrong estimated parameters: reality: w_1= w_3= 0.25 Nm, w_2= 0.15 Nm
  estimated parameters: w_{1e}= w_{3e}= 0.1 Nm, w_{2e}= 0.05 Nm

16) As situation 5) but now:

- wrong estimated parameters: reality: w_1= w_3= 0.25 Nm, w_2= 0.15 Nm
  estimated parameters: w_{1e}= w_{3e}= 0.5 Nm, w_{2e}= 0.3 Nm

17) As situation 5) but now:

- wrong estimated parameters: reality: J_1= 2.15 10^{-3} kgm^2, J_2= 1.45 10^{-4} kgm^2
  est. par.: J_{1e}= 3 10^{-3} kgm^2, J_{2e}= 2 10^{-4} kgm^2

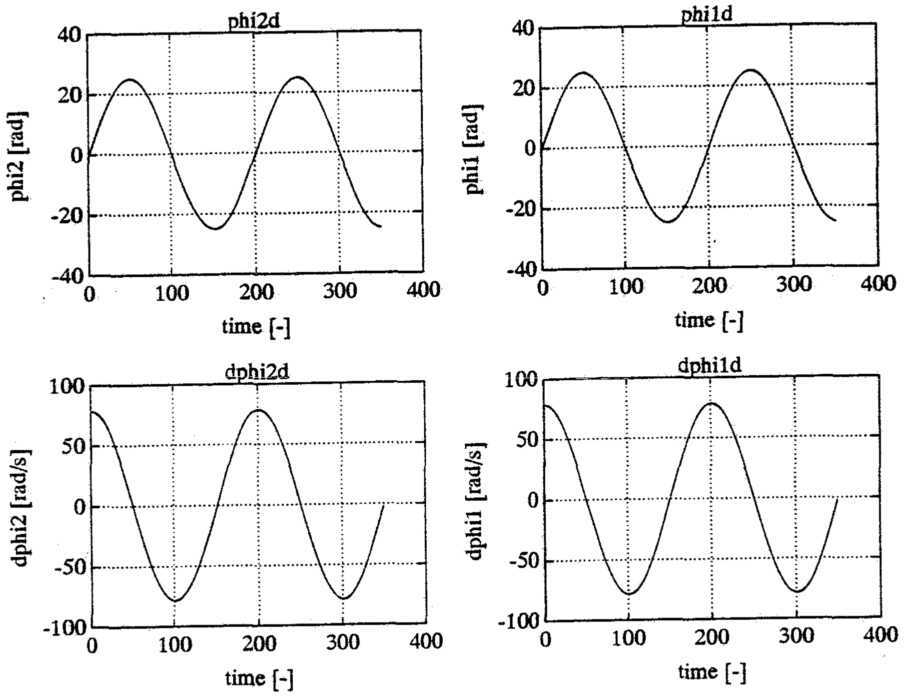
18) As situation 5) but now:

wrong estimated parameters: reality: ms= 2.3 kg, m_t= 8.5 kg, m_e= 2.3 kg
est. par.: m_{se}= 3kg, m_{te}= 10 kg, m_{ee}= 3 kg

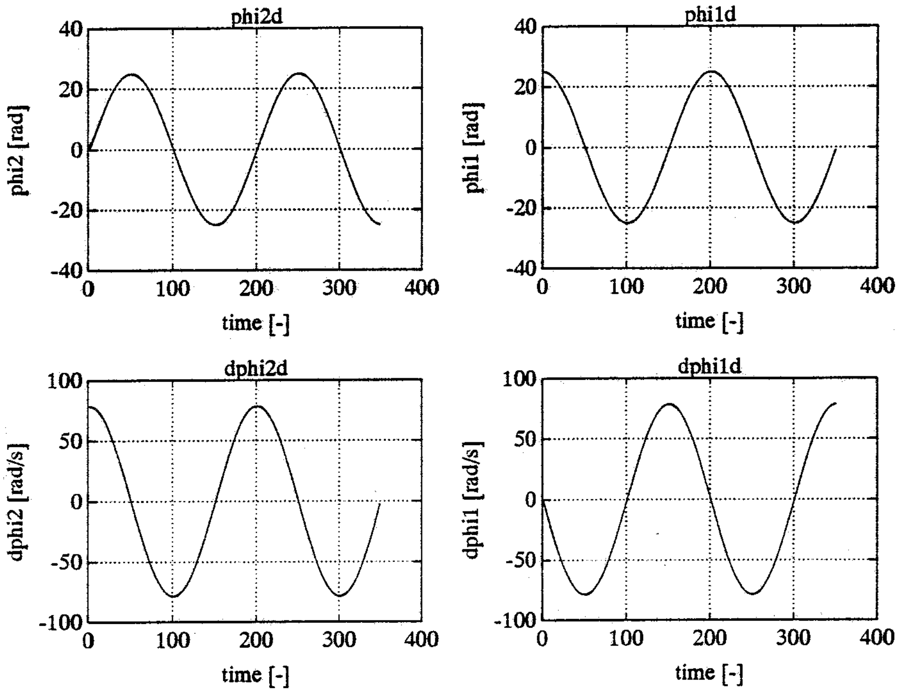
Simulation results

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

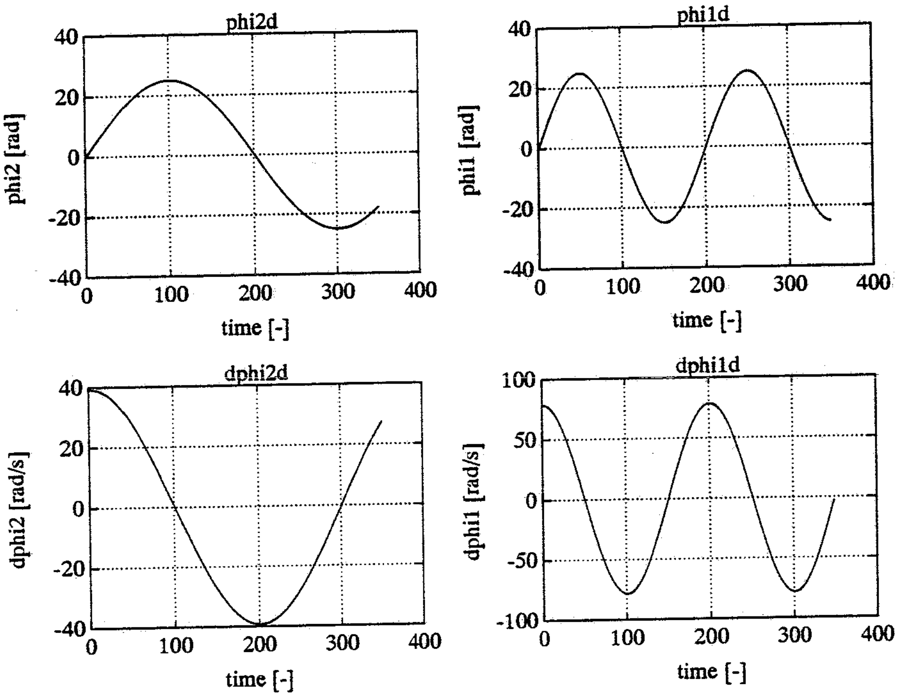
trajectory A



trajectory B

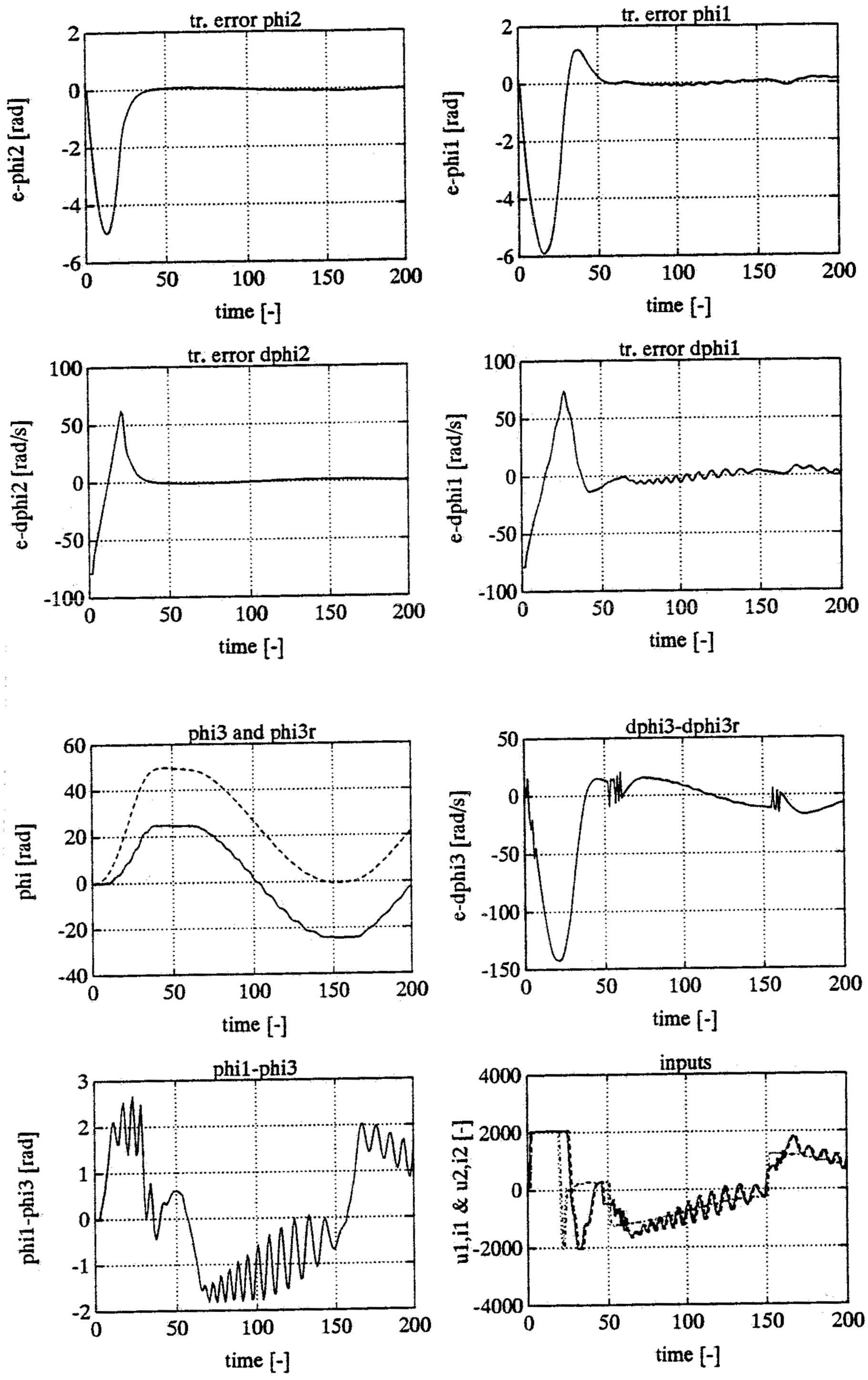


trajectory C



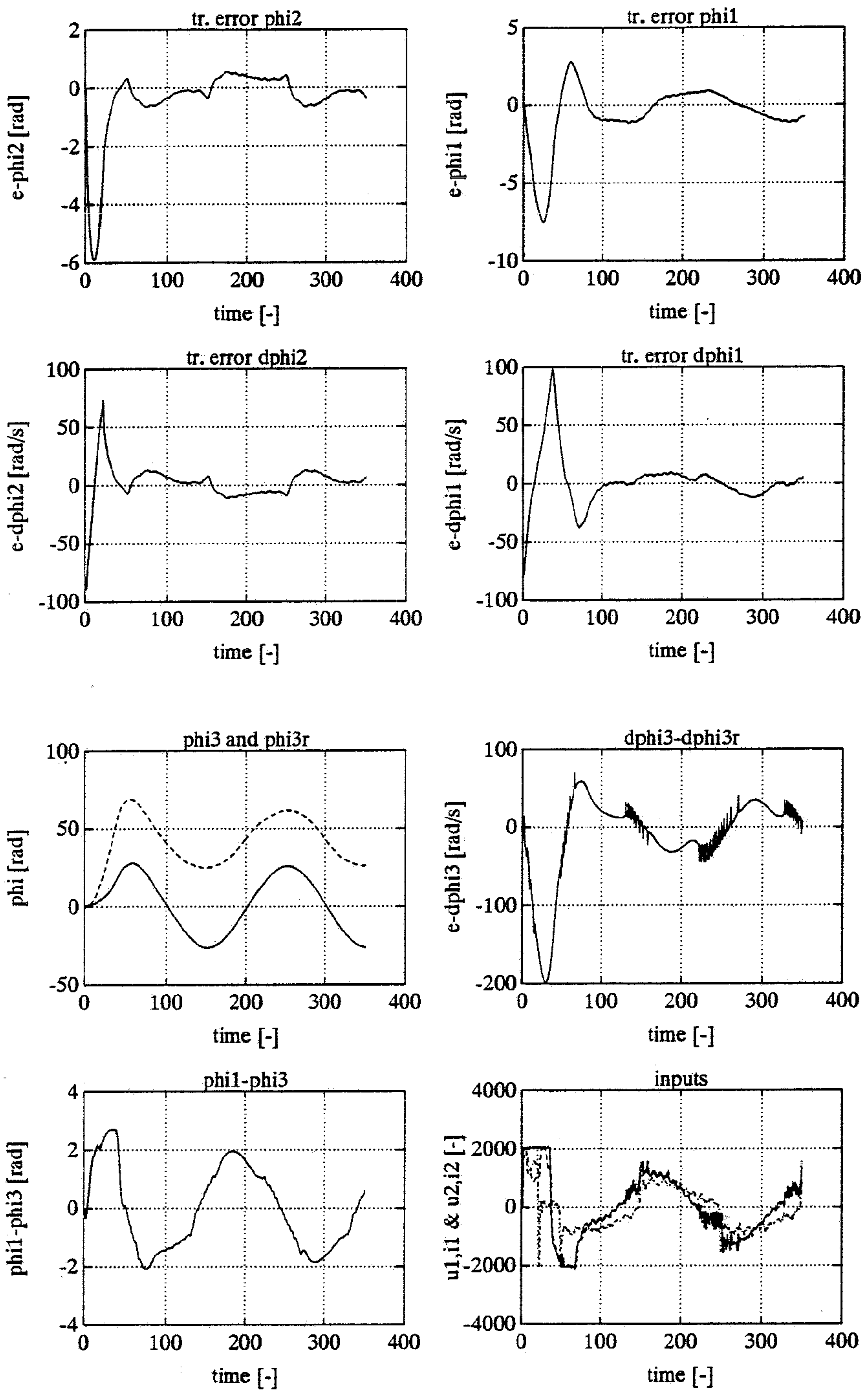
E: Simulation - and experimental results of the second part of the research

situation 1



simulation, desired trajectory: A , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 2



tr. error phi2

tr. error phi1

tr. error dphi2

tr. error dphi1

phi3 and phi3r

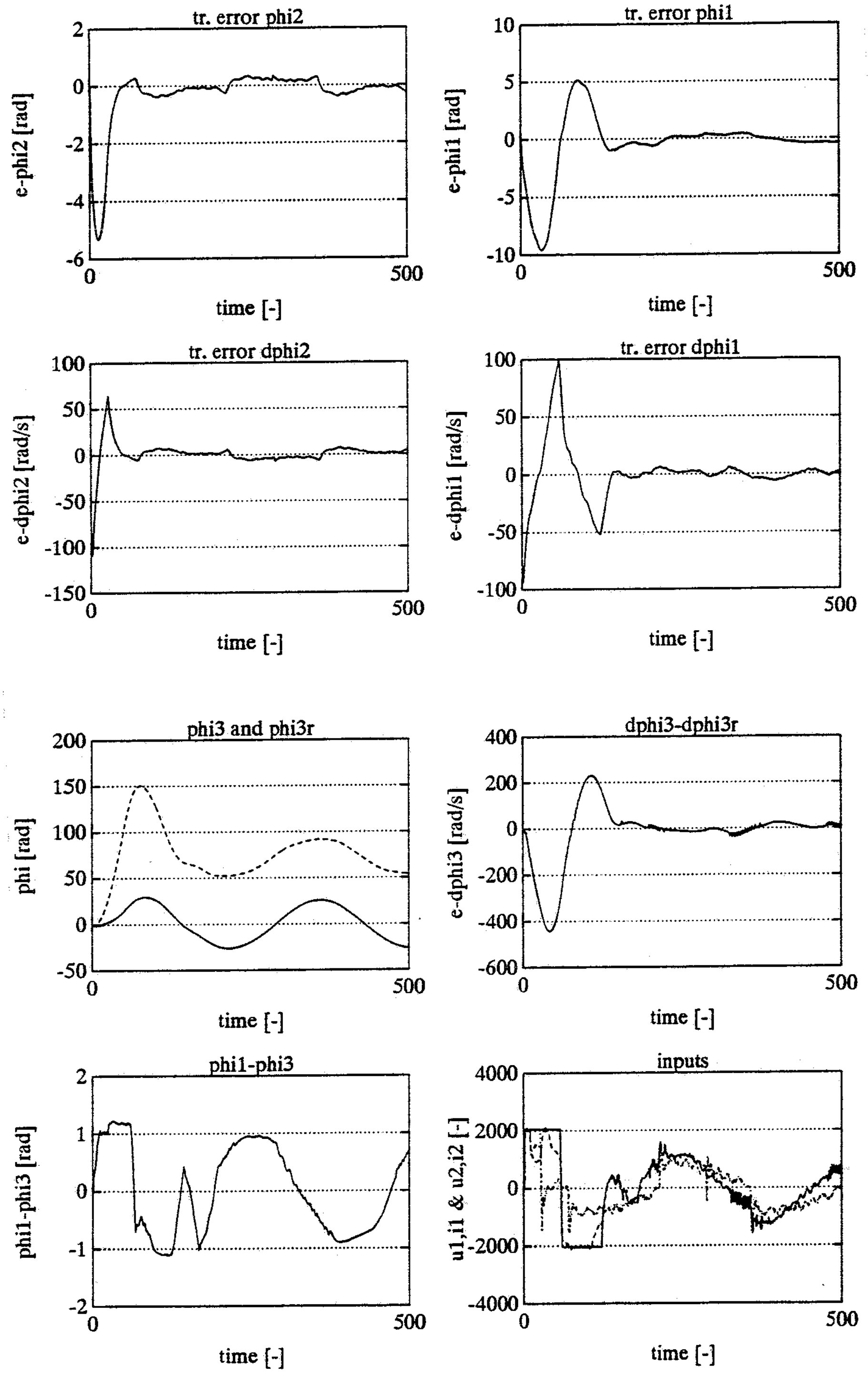
dphi3-dphi3r

phi1-phi3

inputs

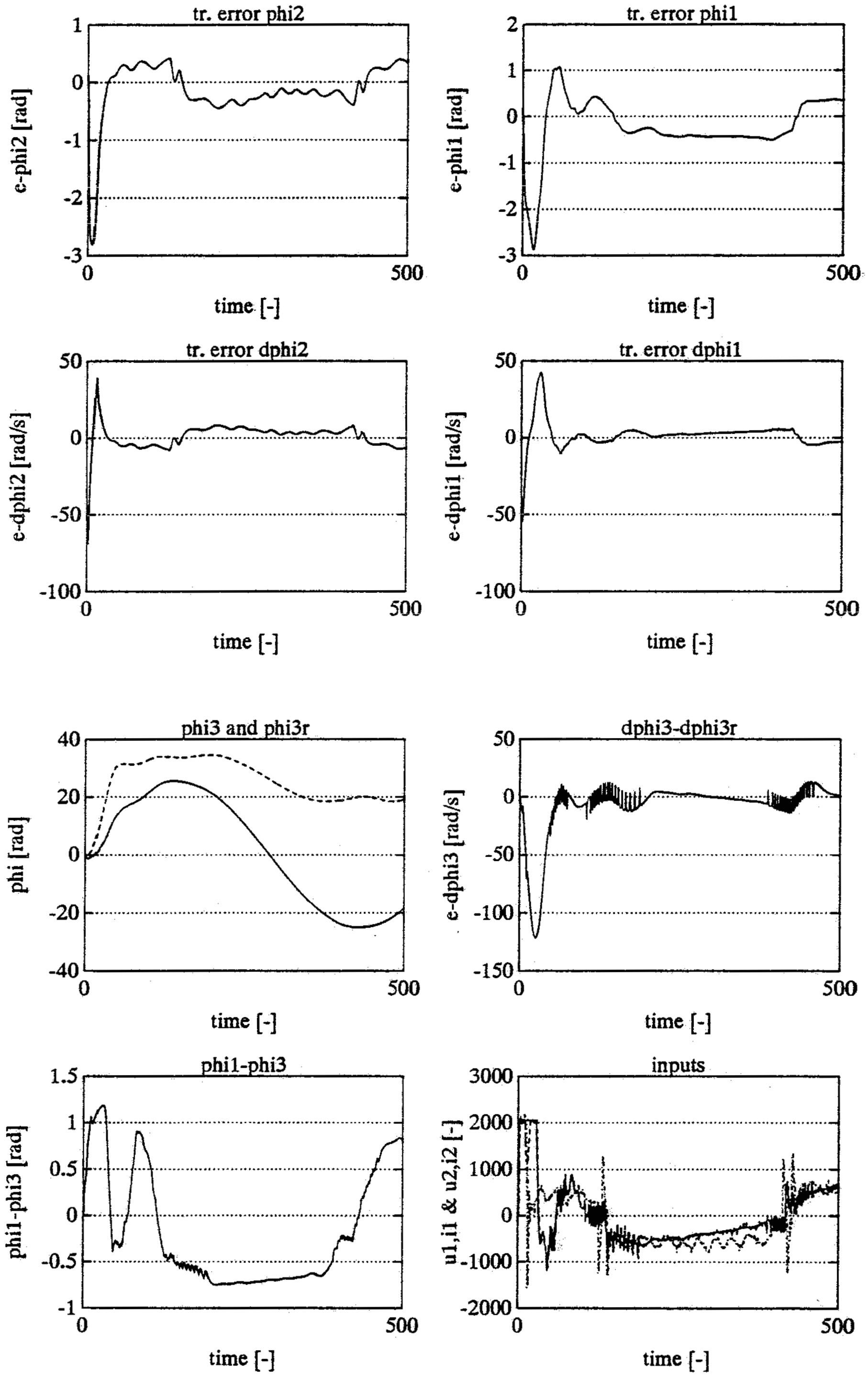
experiment, desired trajectory: A , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi$1(0)= 0, $\Delta\varphi_3(0)$= 0

situation 3



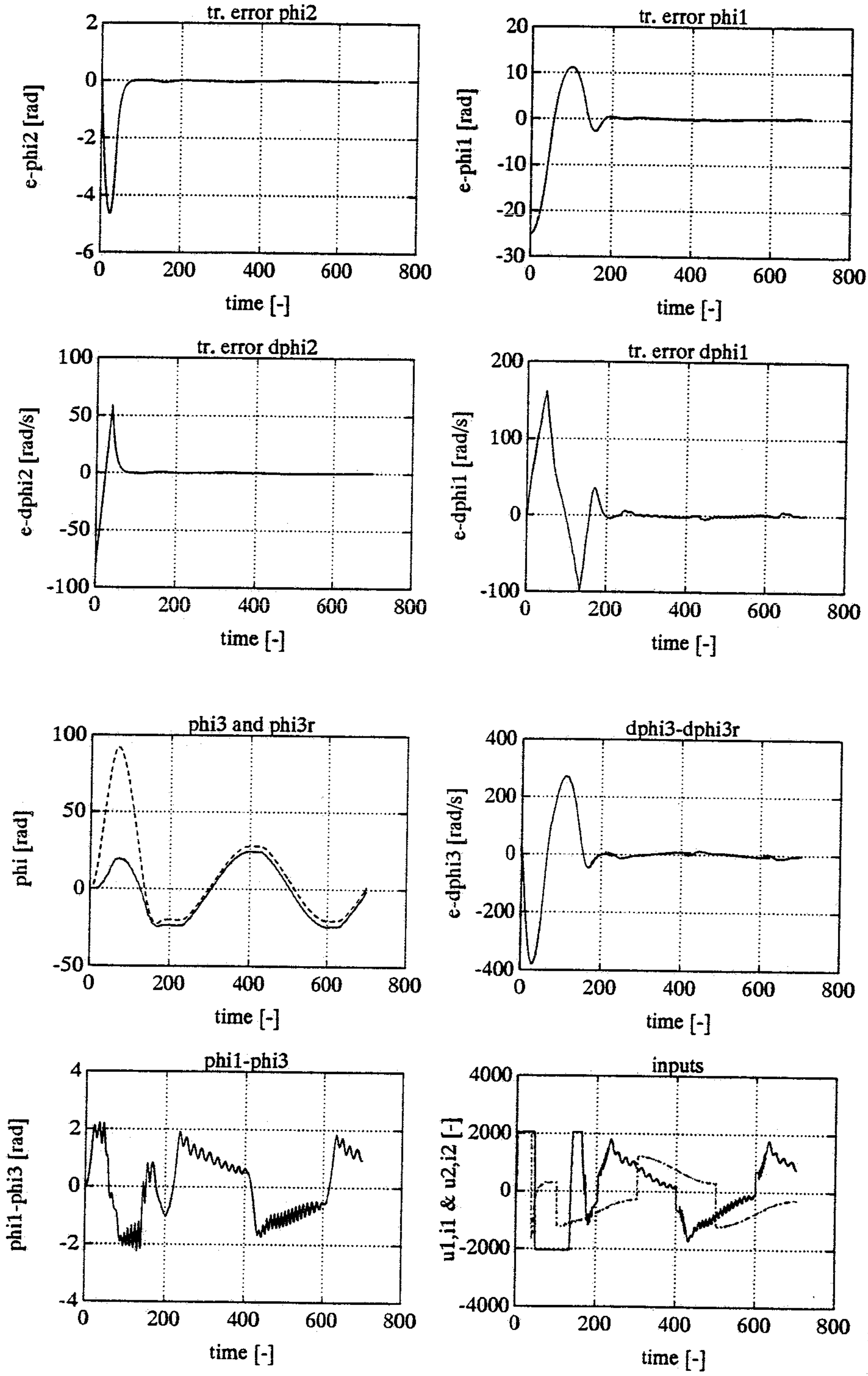
experiment, desired trajectory: A , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 50 and λ_2= 25
meas. freq.: 1/0.007 Herz, observ. poles: -50, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 4



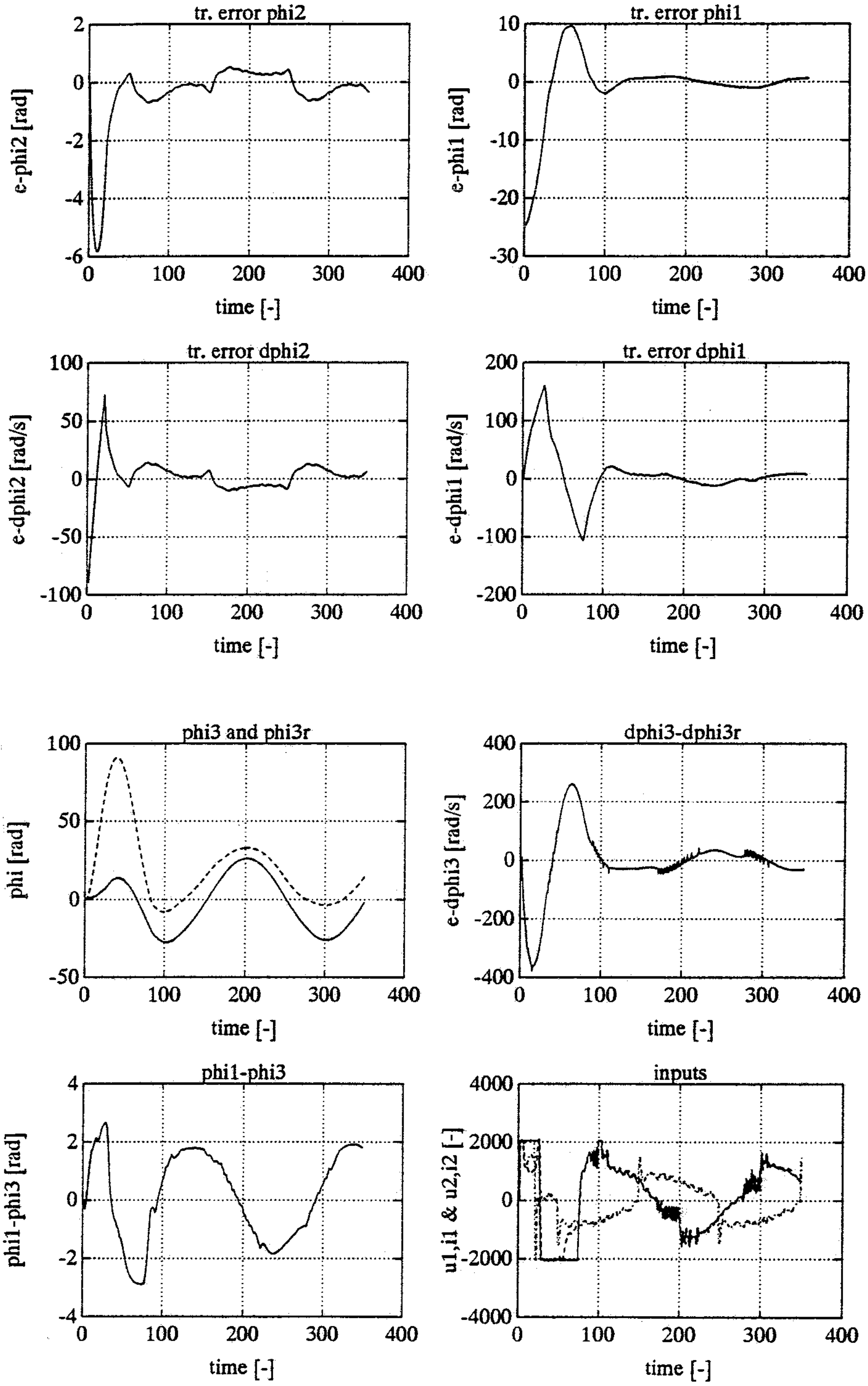
experiment, desired traj: A with ω=1.57, k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.007 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 5



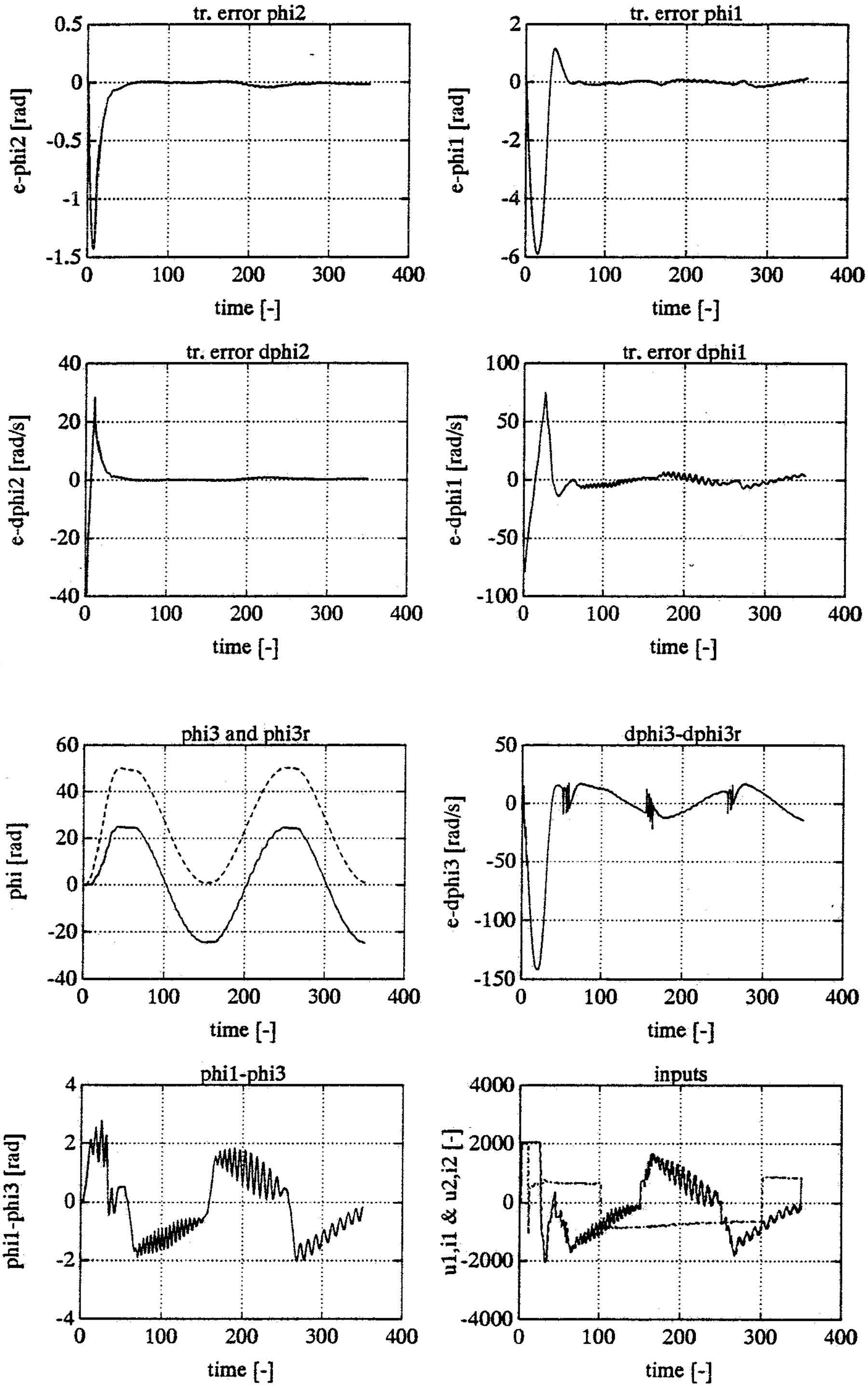
simulation, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.005 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 6



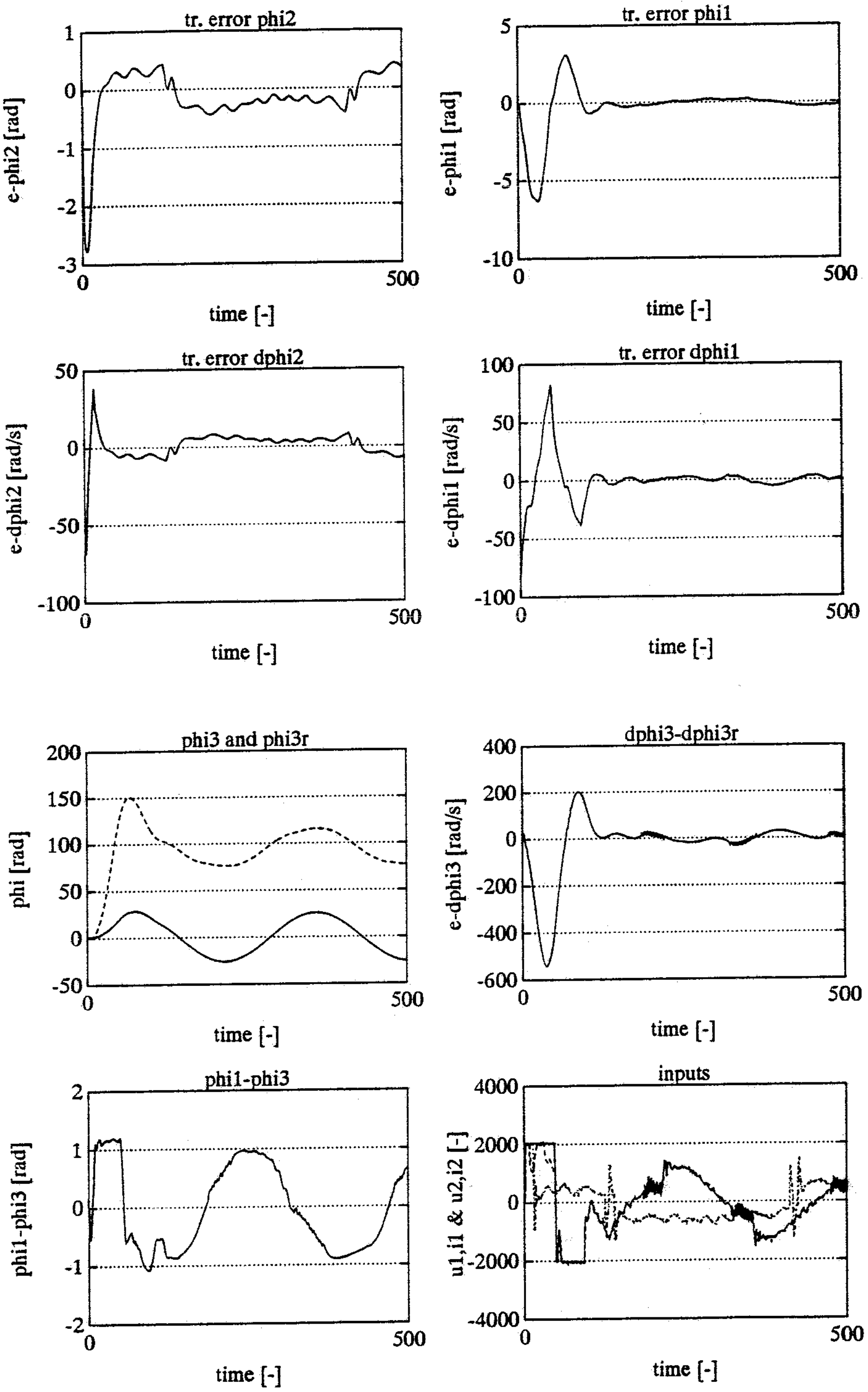
experiment, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 7



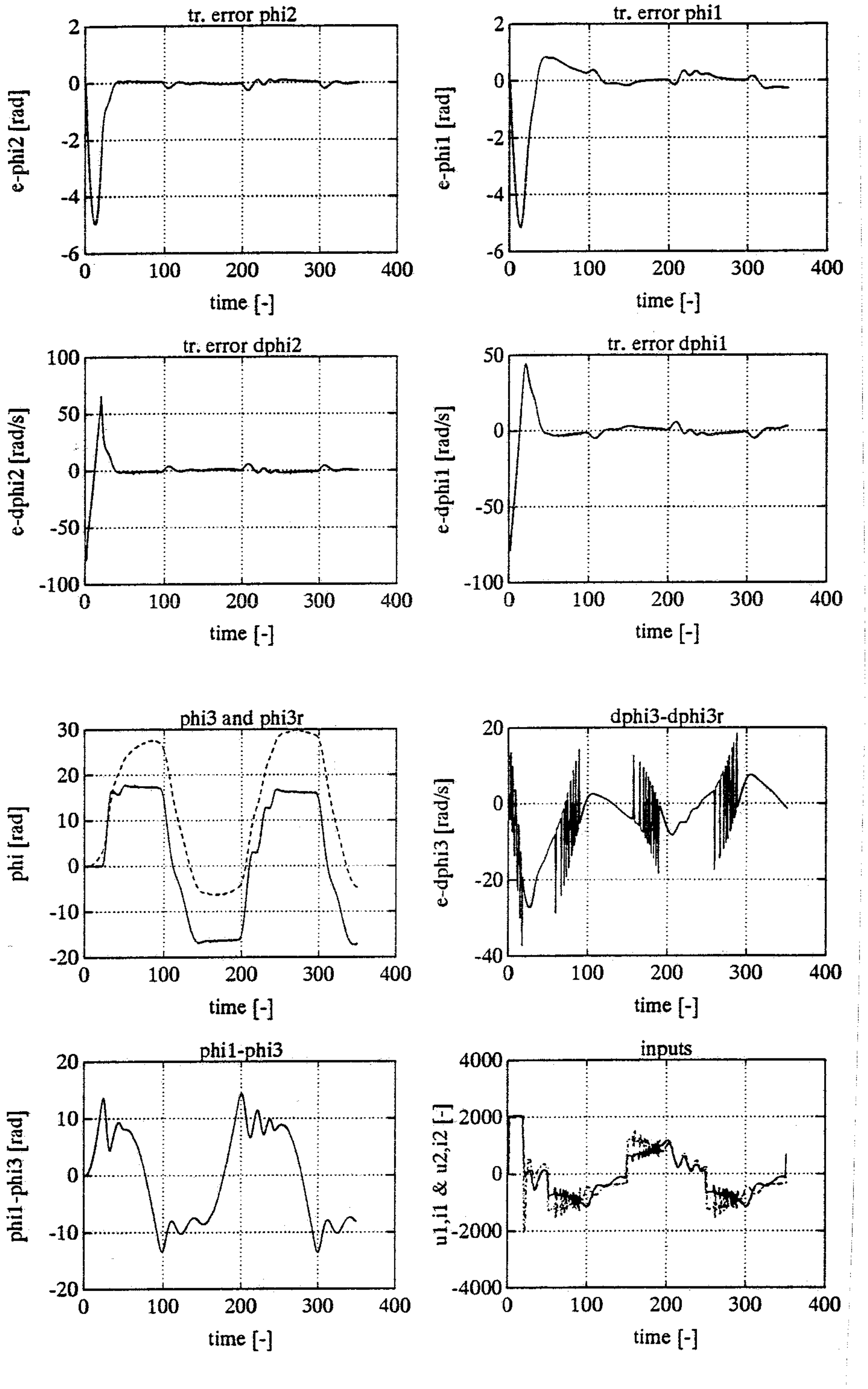
simulation, desired trajectory: C , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 8



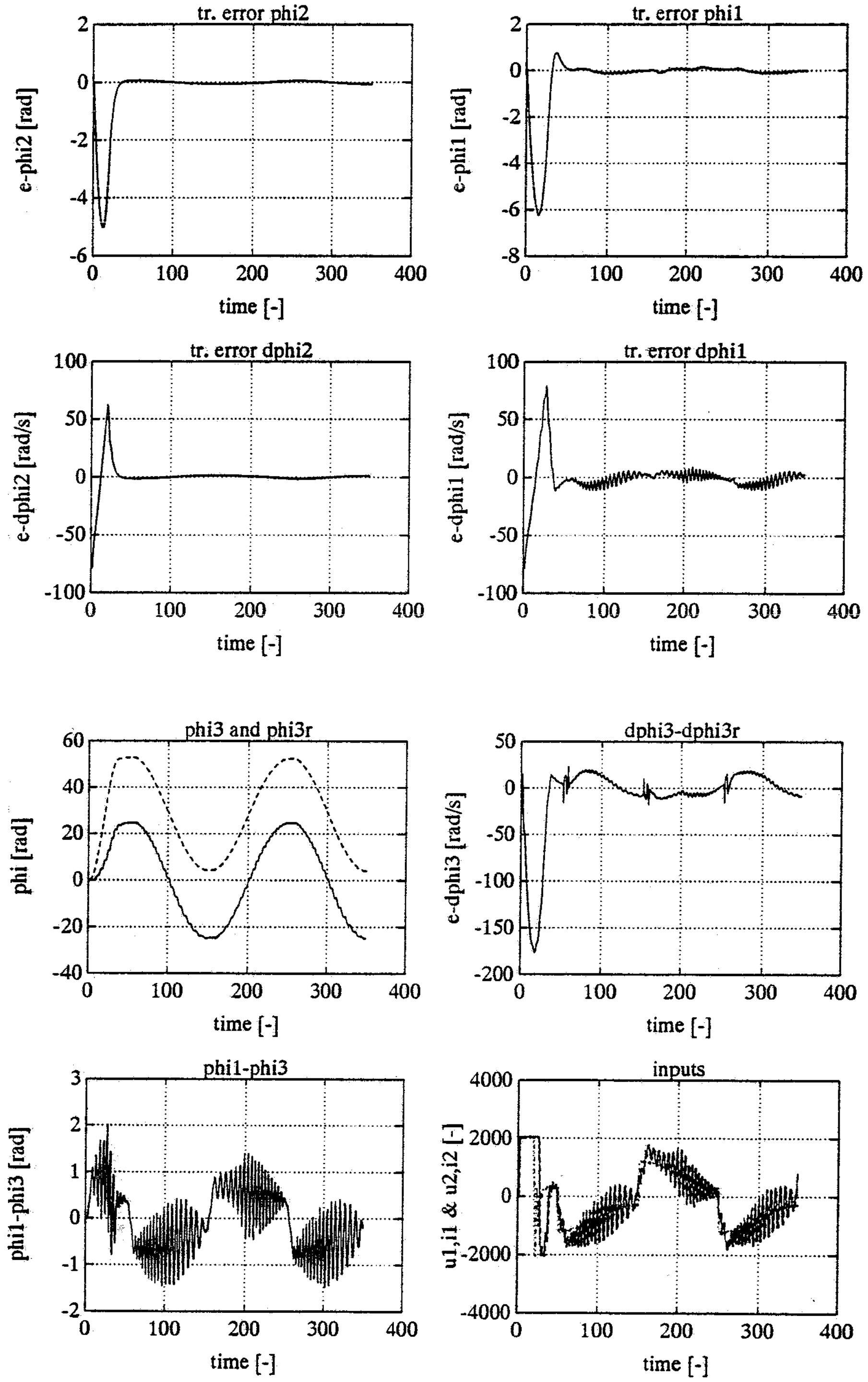
experiment, desired trajectory: C , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 50 and λ_2= 25
meas. freq.: 1/0.007 Herz, observ. poles: -50, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 9



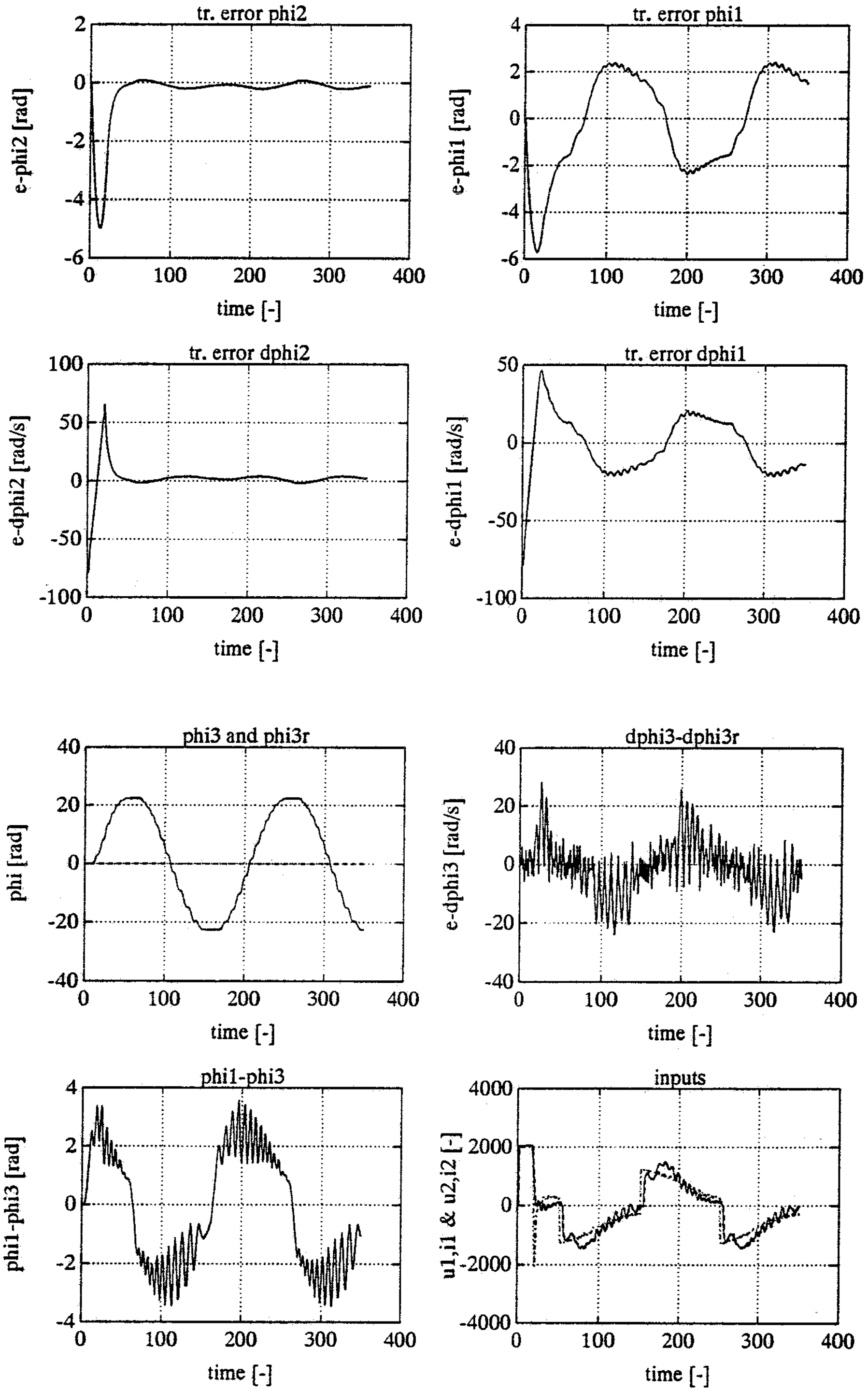
simulation, desired trajectory: A , k= 0.05 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 10



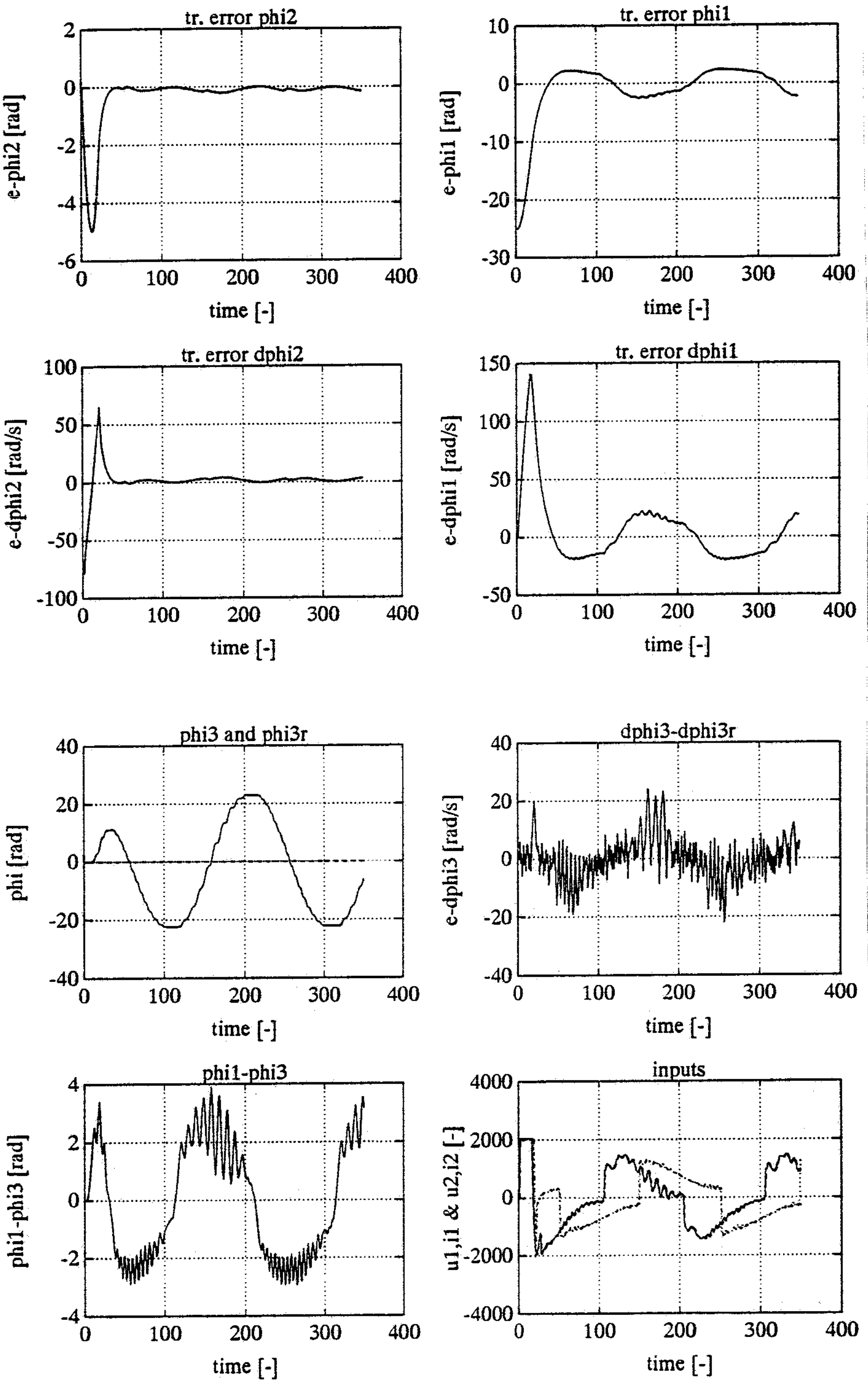
simulation, desired trajectory: A , k= 1 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 11



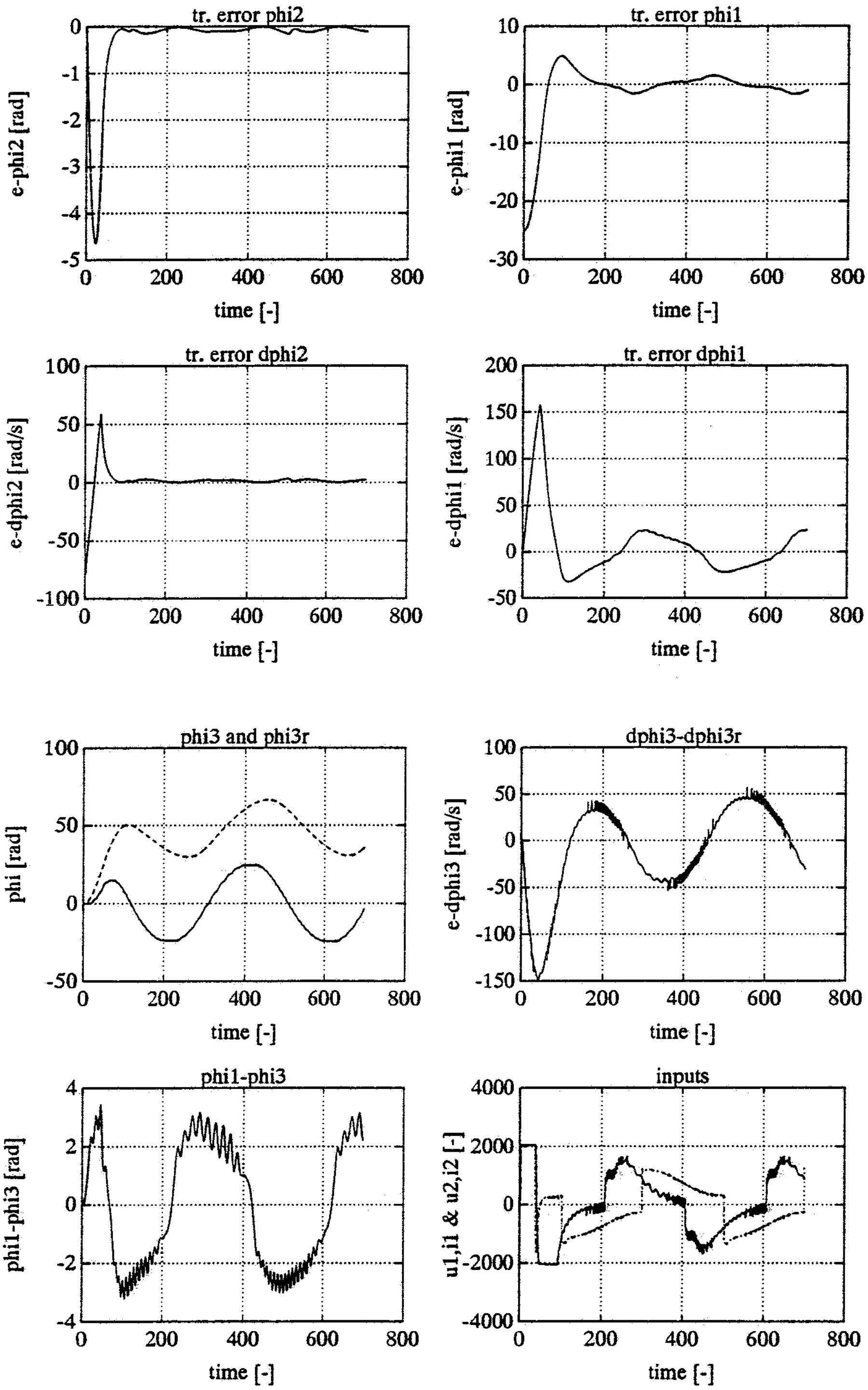
simulation with CTC controller, desired trajectory: A , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, λ_1= 25, λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 12



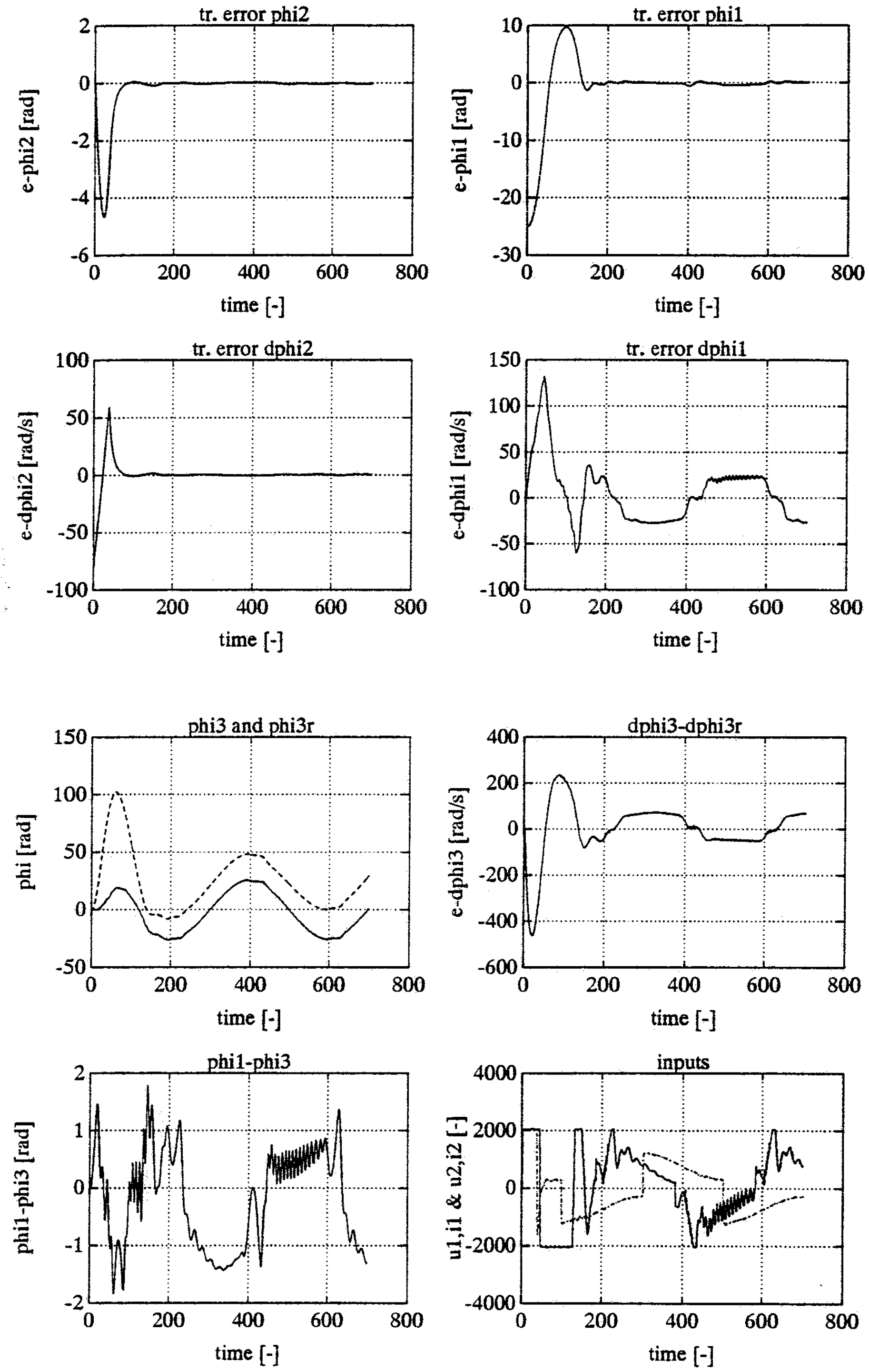
simulation with CTC controller, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, λ_1= 25, λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0

situation 13



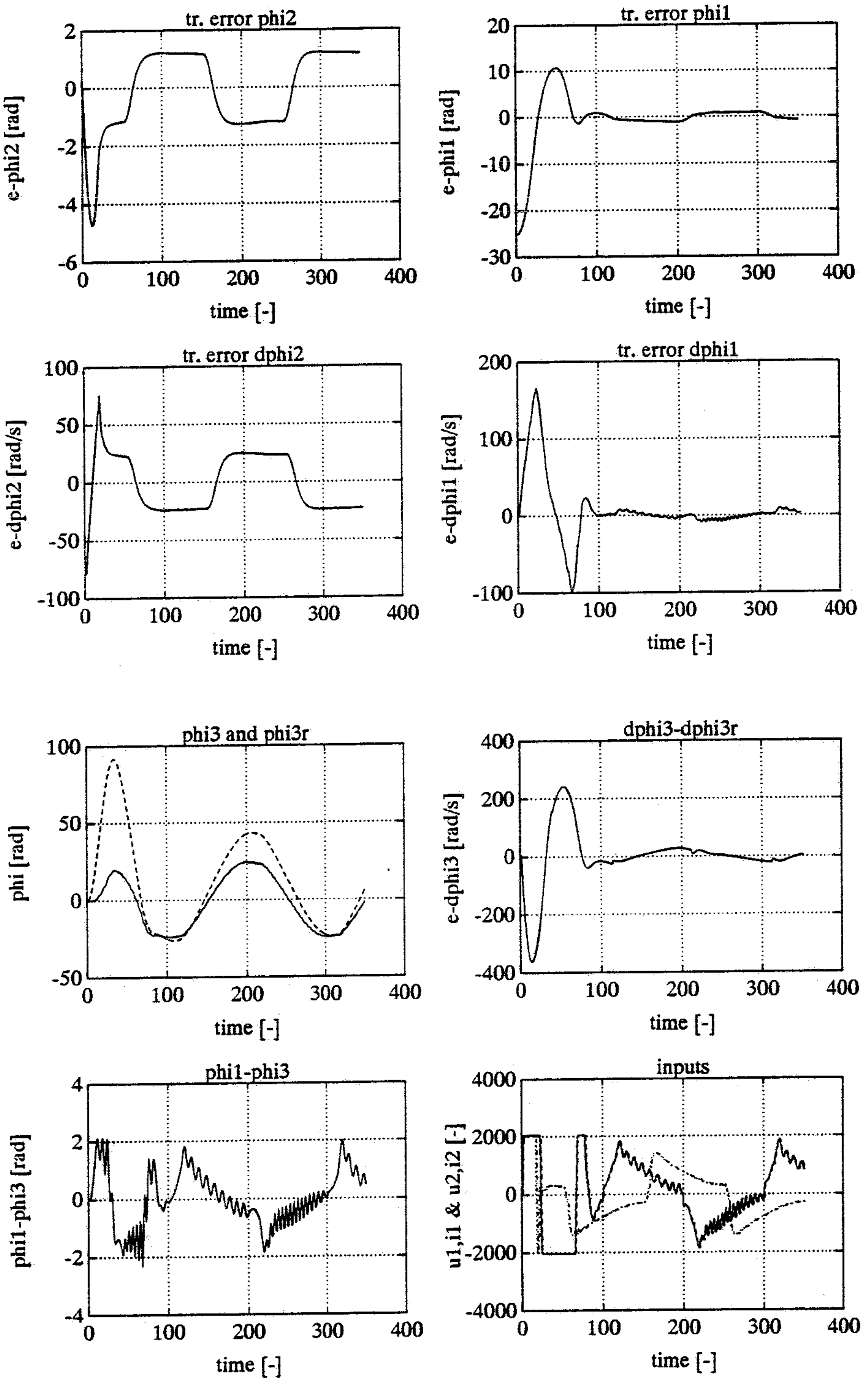
simulation, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : (k= 0.5 Nm/rad k_e= 0.1 Nm/rad)

situation 14



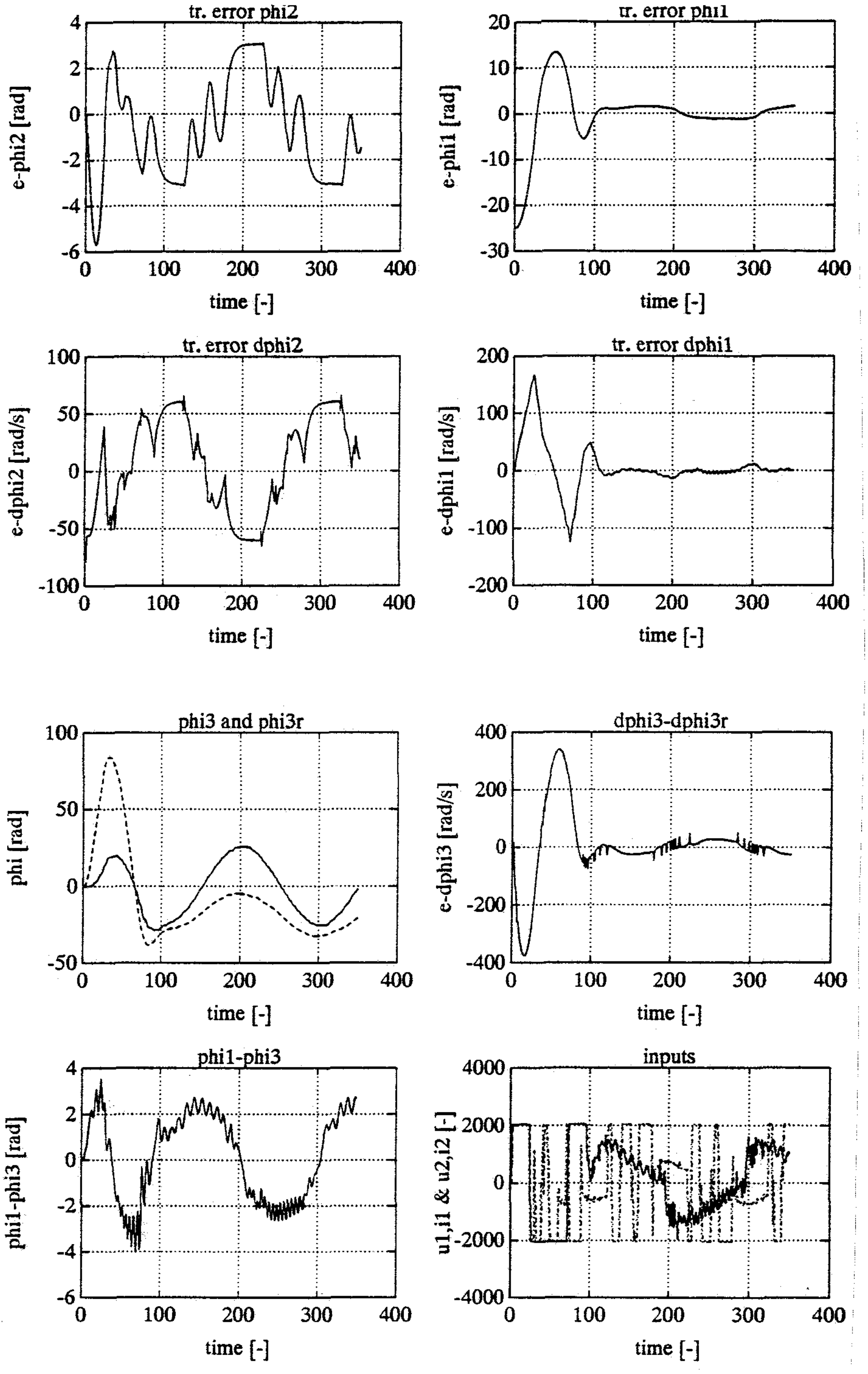
simulation, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : (k= 0.5 Nm/rad k_e= 1 Nm/rad)

situation 15



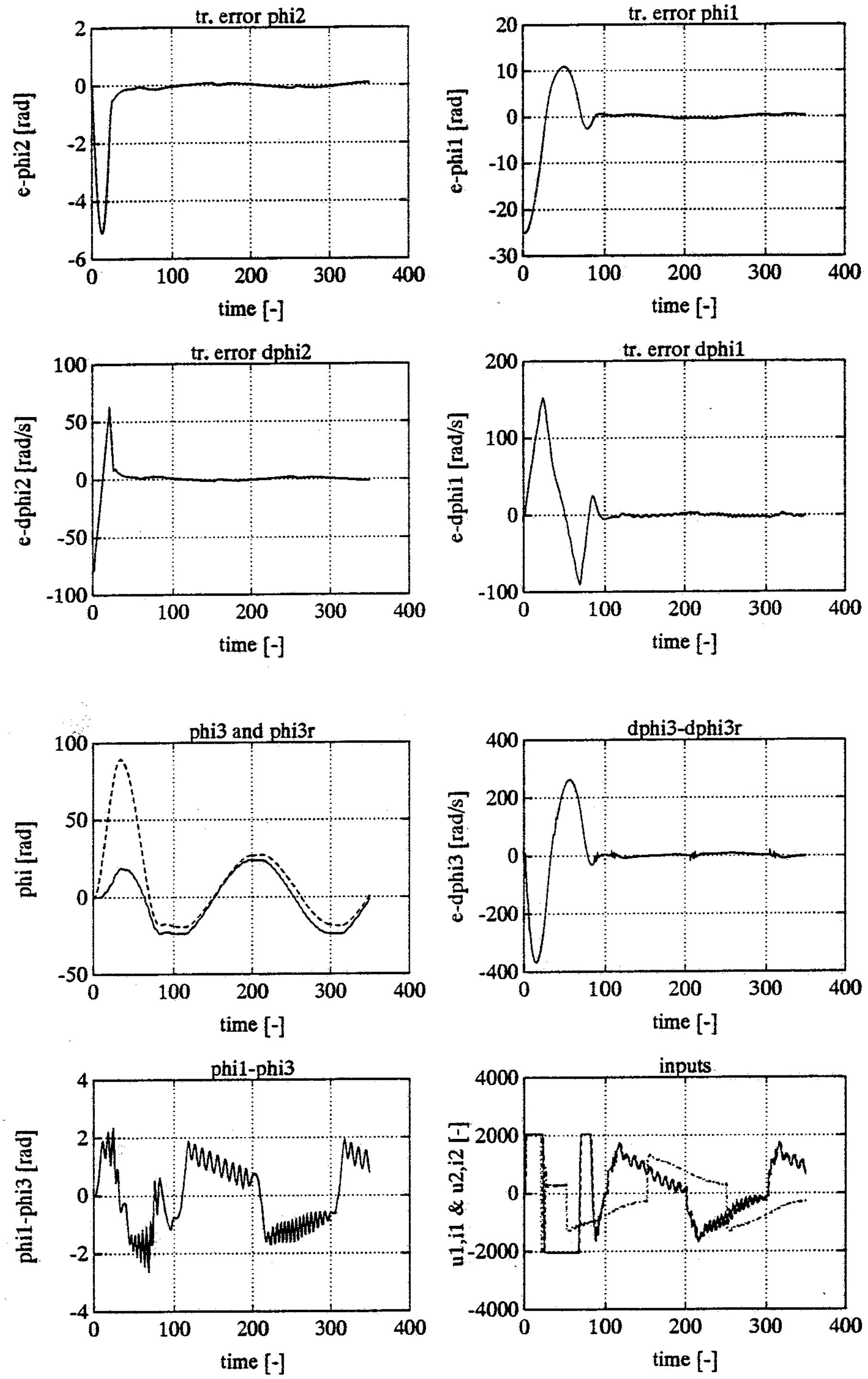
simulation, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : (w_1= w_3= 0.25 Nm w_2= 0.15 Nm w_{1e}= w_{3e}= 0.1 Nm w_{2e}= 0.05 Nm)

situation 16



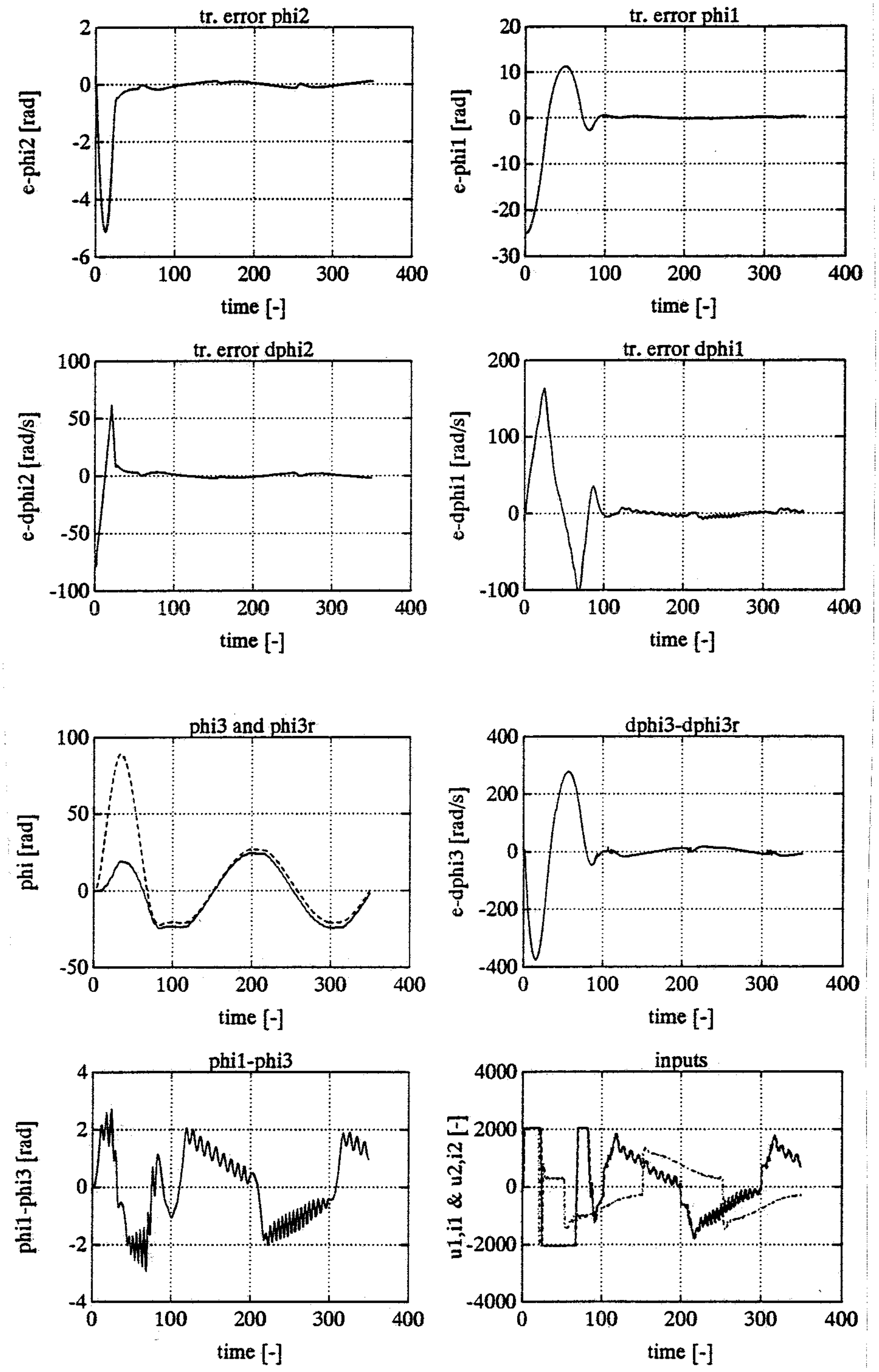
simulation, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : (w_1= w_3= 0.25 Nm w_2= 0.15 Nm w_{1e}= w_{3e}= 0.5 Nm w_{2e}= 0.3 Nm)

situation 17



simulation, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : (J_1= 2.15e-3, J_2= 1.45e-4 J_{1e}= 3e-3 J_{2e}= 2e-4)

situation 18



simulation, desired trajectory: B , k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0
wrong estimated par. : (m_s= 2.3 kg m_t= 8.5 kg m_e= 2.3 kg m_{se}= 3 kg m_{te}= 10 kg m_{ee}= 3 kg)

APPENDIX F: OUTPUT CONTROL

In this appendix I will give s_x and s_y of equation (6.2) and (6.5). Further I will show some simulation - and experimental results.

s_x and s_y

From fig 6.1 it follows:

$$s_x r = \sin(atan(\frac{\varphi_3 r - \varphi_1 r}{l}))(\frac{0.5l + \varphi_2 r}{\cos(atan(\frac{\varphi_3 r - \varphi_1 r}{l}))}) \qquad (F.1)$$

Because of the fact that φ_3r-φ_1r is small, we can simplify equation (F.1):

$$s_x r \approx (\frac{\varphi_3 r - \varphi_1 r}{l})(0.5l + \varphi_2 r) \qquad (F.2)$$

Now I will consider the s_y. From fig. 6.2 it follows:

$$y = \varphi_2 r cos\beta \qquad (F.3)$$

$$\cos\beta = \cos(atan(\frac{\varphi_1 r - \varphi_3 r}{l})) \qquad (F.4)$$

Because of the fact that φ_3r-φ_1r is small, we can simplify equation (F.4):

$$\cos\beta \approx 1-\frac{1}{2}(\frac{\varphi_1 r-\varphi_3 r}{l})^2 \qquad (F.5)$$

This leads to:

$$y \approx \varphi_2 r-\frac{1}{2}\varphi_2 r(\frac{\varphi_1 r-\varphi_3 r}{l})^2 = \varphi_2 r-s_y \qquad (F.6)$$

In the on-line generation routine of φ_{1d} and φ_{2d} (6.2, 6.5 and 6.6 and 6.7) I have used s_x and s_y of equation (F.2) and (F.6).
We have to derivate s_x and s_y twice (to calculate $\dot{x}_d$, $\ddot{x}_d$ and $\dot{y}_d$, $\ddot{y}_d$). If we had use F.1 anf F.3 we get long terms, this can lead to problems during the execution of experiments (computing time can grow too big).

Simulationresults

I have considered the next situations:

1)
- simulation
- desired trajectory: x_d= αsin(ωt) y_d= αsin(ωt) α= 25 rad ω= 3.14 rad/s
- springconstant: k = 0.5 Nm/rad
- control gain parameters: k_1= k_2= k_3= 0.05, lab_1= lab_2= 25
- observer-poles (used by the pole-placement routine): -25+0.2i
-25+0.3i
-25+0.4i
- measure frequency: 0.01 s
- execution time: 3.5 s
- position t=0: φ_1= φ_2= φ_3= 0 (middle of the xy table)
- velocity t=0: φ_1= φ_2= φ_3= 0
- particularies: no on-line generation of φ_{1d} and φ_{2d}

Note: This is the same situation as situation 1 of appendix E.

2) As situation 1) but now:

- : $a_x= a_y= 1$

Note: This is the same as the on-line generation routine of φ_{1d} and φ_{2d} with (6.2) and (6.5)

3) As situation 2) but now:

- scaled on-line generation of φ_{1d} and φ_{2d}: $a_x= a_y= 0.75$ (see 6.6 and 6.7)

4) As situation 2) but now:

- scaled on-line generation of φ_{1d} and φ_{2d}: $a_x= a_y= 0.5$

5) As situation 2) but now:

- experiment
- control gain parameters $k_1= k_2= k_3= 0.05$, $lab_1= 50$, $lab_2= 25$
- observer-poles: -50+0.2i, -50+0.3i, -50+0.4i
- measure frequency: 0.007 s
- scaled on-line generation of φ_{1d} and φ_{2d}: $a_x= a_y= 0.25$

6) As situation 5) but now:

- desired trajectory: $x_d= \alpha\cos(\omega t)$ $y_d= \alpha\sin(\omega t)$, $\alpha= 25$ rad $\omega= 3.14$ rad/s

7) As situation 6) but now:

- desired trajectory: $x_d=\alpha\cos(\omega_1 t)$ $y_d=\alpha\sin(\omega_2 t)$, $\alpha=25$ rad $\omega_1=3.14$ rad/s, $\omega_2=1.57$ rad/s

The simulation - and experimental results

Of every situation I have made the following plots:

1 tracking error x-direction: x_d-x
2 tracking error y-direction: y_d-y
3 input x-direction: u_e
4 input y-direction: us
5 desired and real trajectory of the end-effector

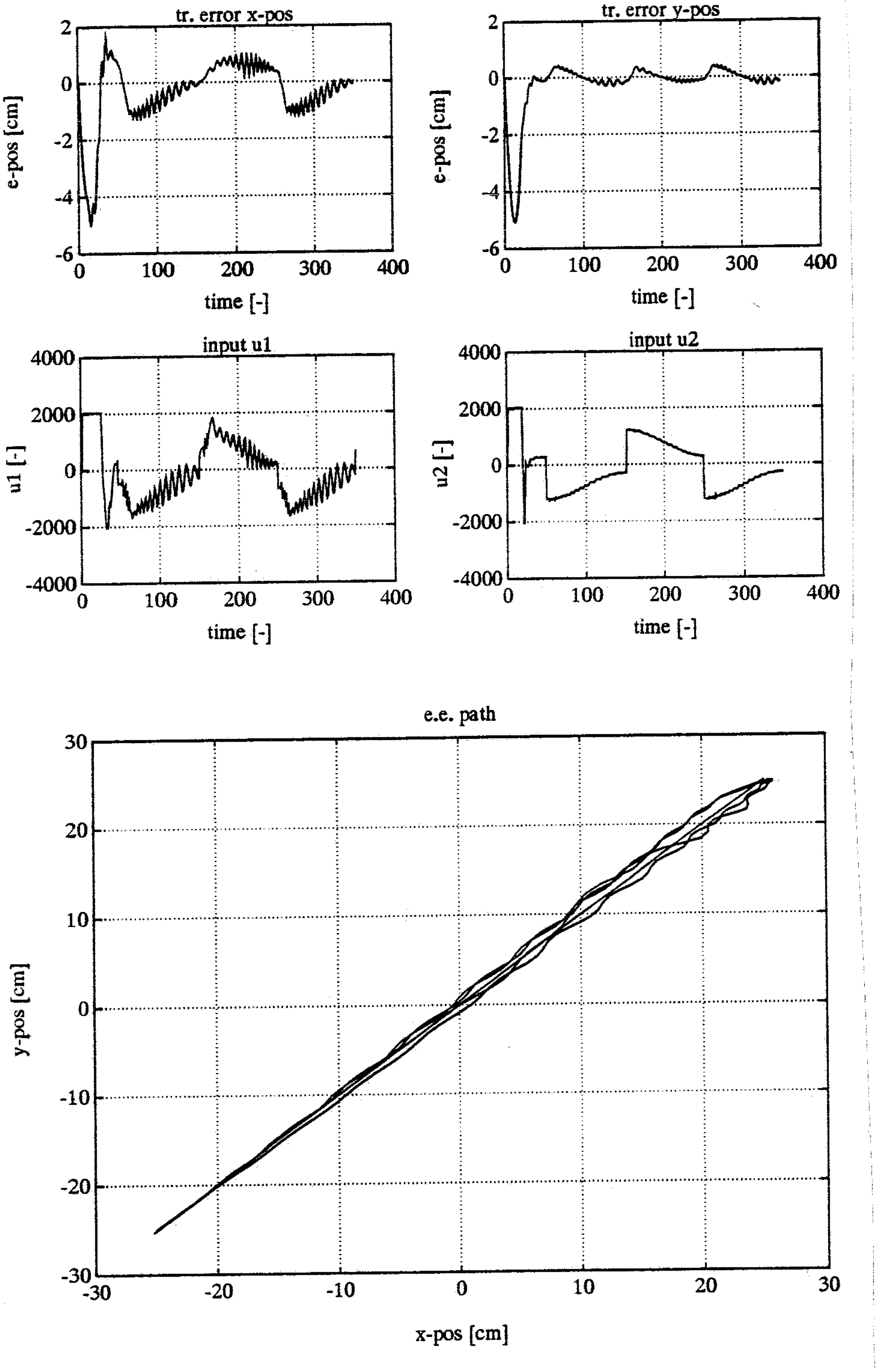
The structure of a page with plots is:

$$\begin{bmatrix} (1) & (2) \\ (3) & (4) \\ \multicolumn{2}{c}{(5)} \end{bmatrix}$$

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

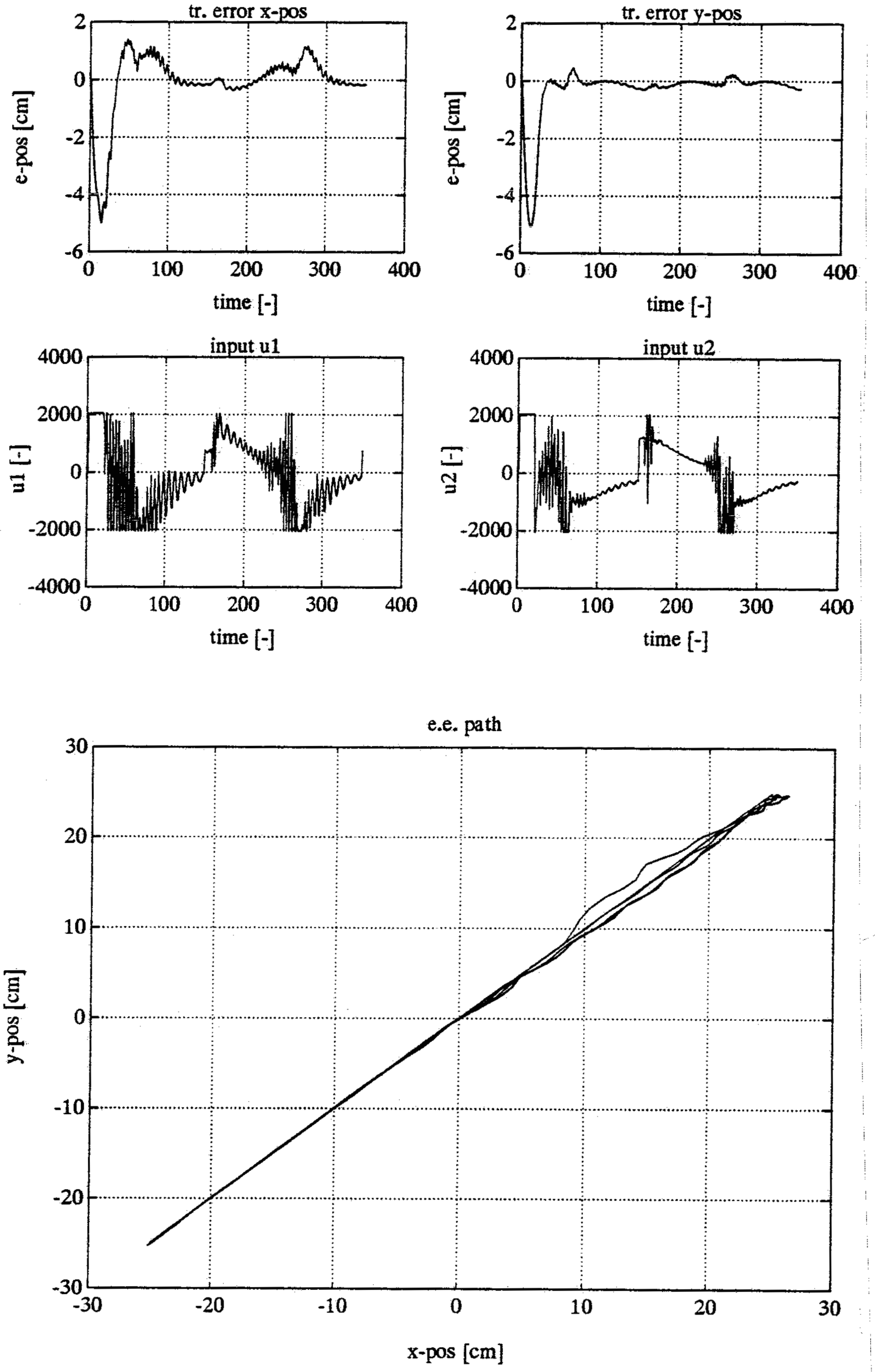
F: Results of the controlling of the end-effector of the xy table

situation 1



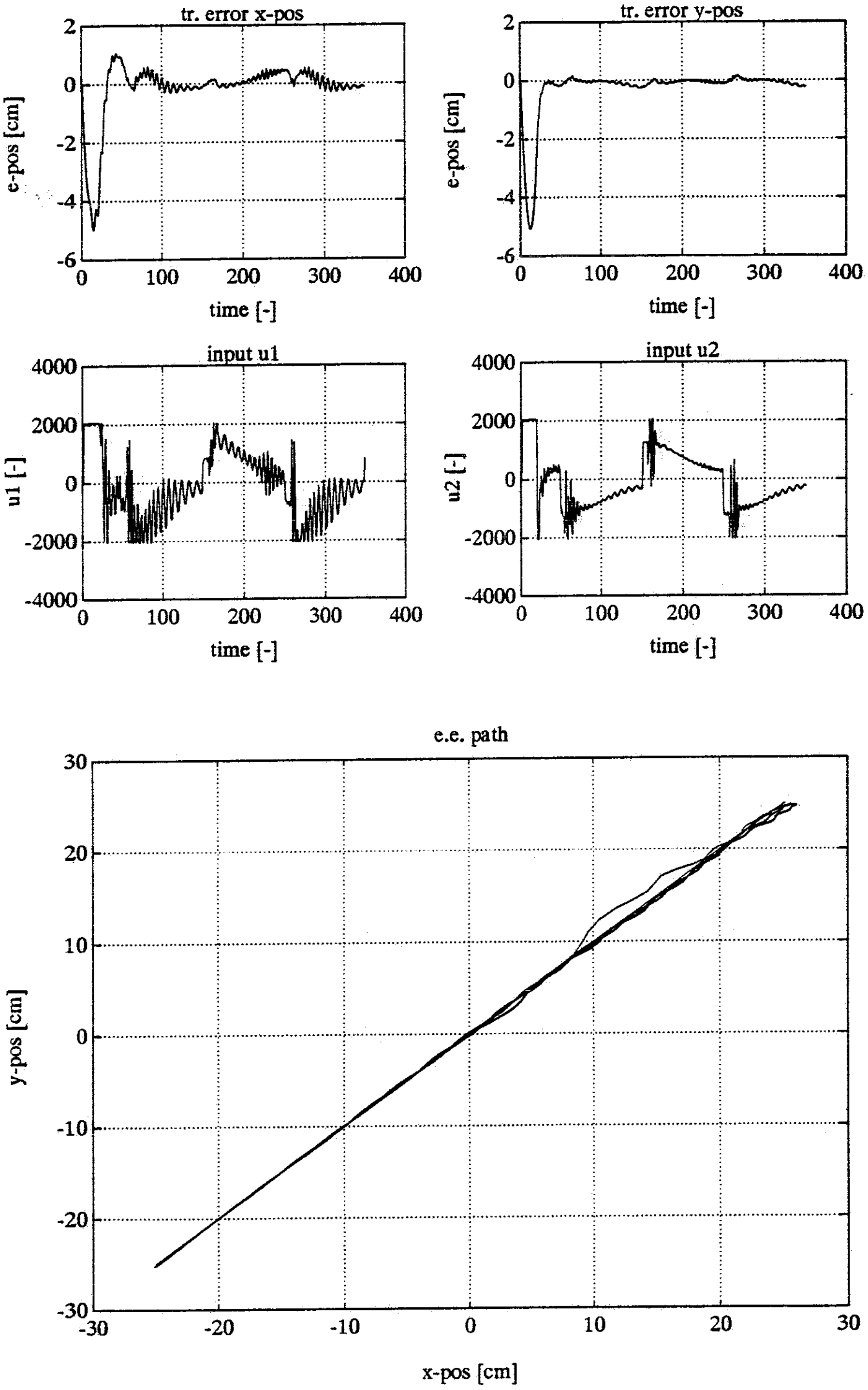
simulation, desired trajectory: x_d= 25sin(3.14t) y_d= 25sin(3.14t),
k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0, a_x= a_y= 0

situation 2



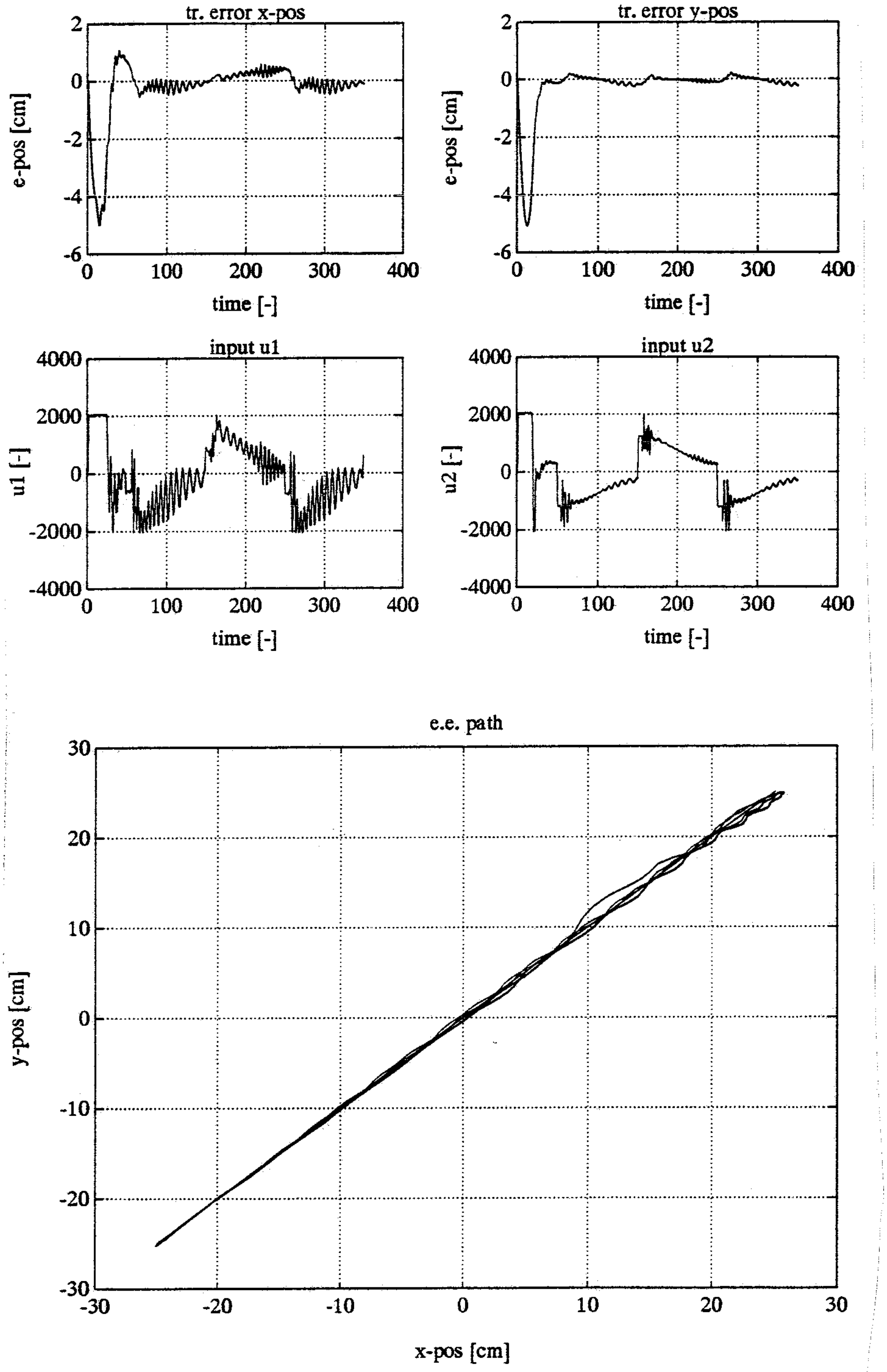
simulation, desired trajectory: x_d= 25sin(3.14t) y_d= 25sin(3.14t),
k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0, a_x= a_y= 1

situation 3



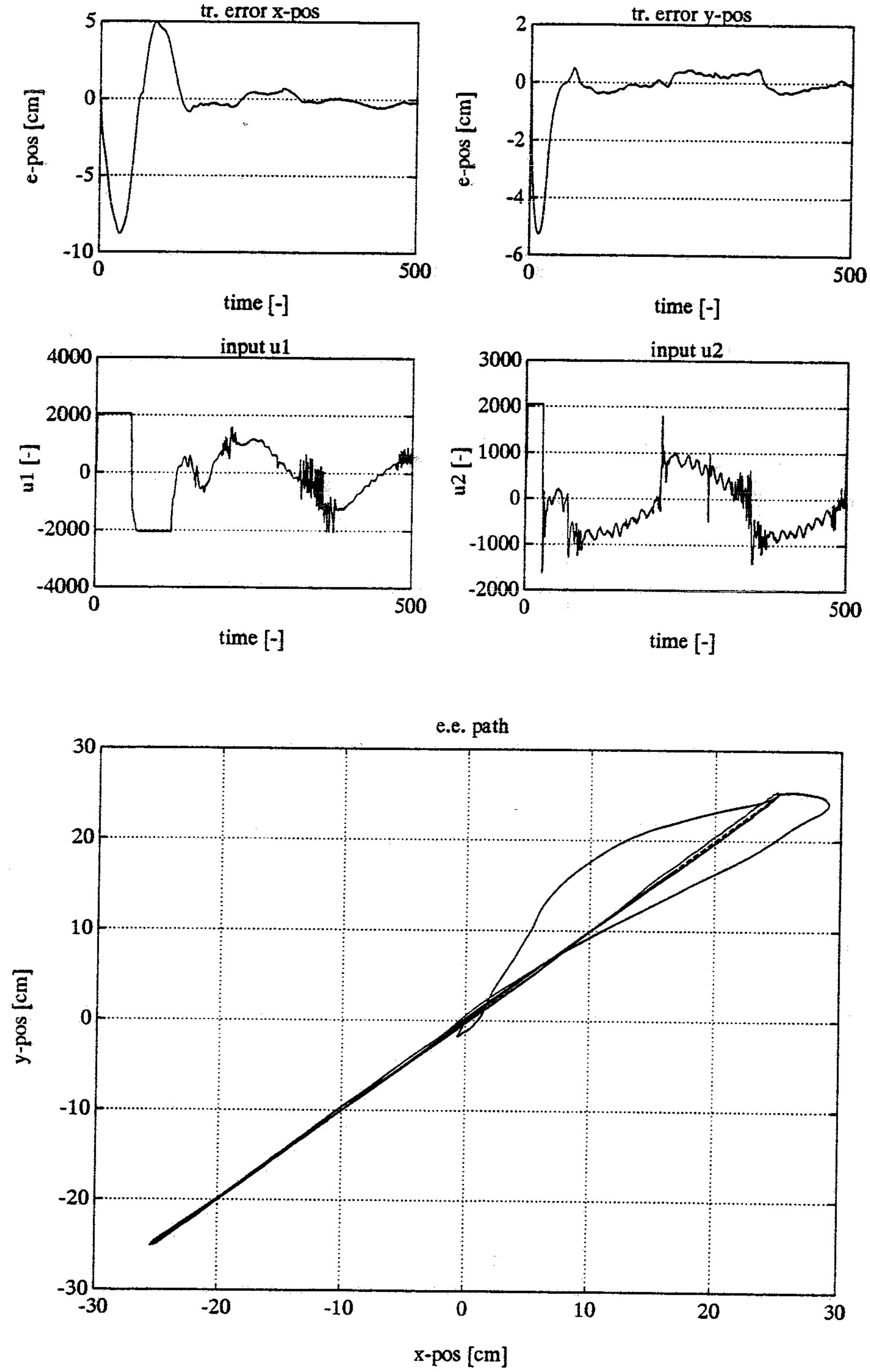
simulation, desired trajectory: x_d= 25sin(3.14t) y_d= 25sin(3.14t),
k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0, a_x= a_y= 0.75

situation 4



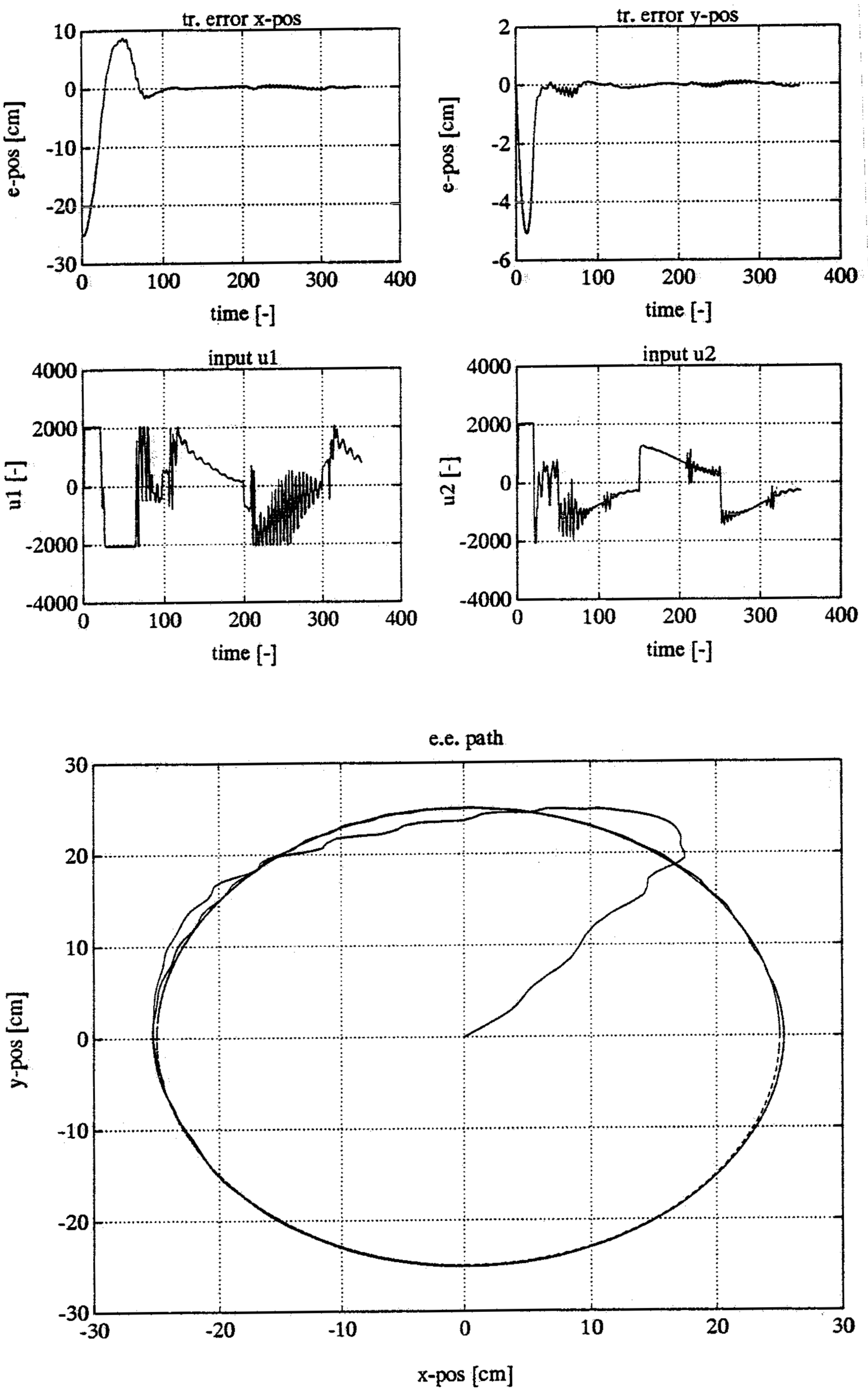
simulation, desired trajectory: x_d= 25sin(3.14t) y_d= 25sin(3.14t),
k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 25 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0, a_x= a_y= 0.5

situation 5



experiment, desired trajectory: x_d= 25sin(3.14t) y_d= 25sin(3.14t),
k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 50 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -50, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi1(0)$= 0, $\Delta\varphi_3(0)$= 0, a_x= a_y= 0.25

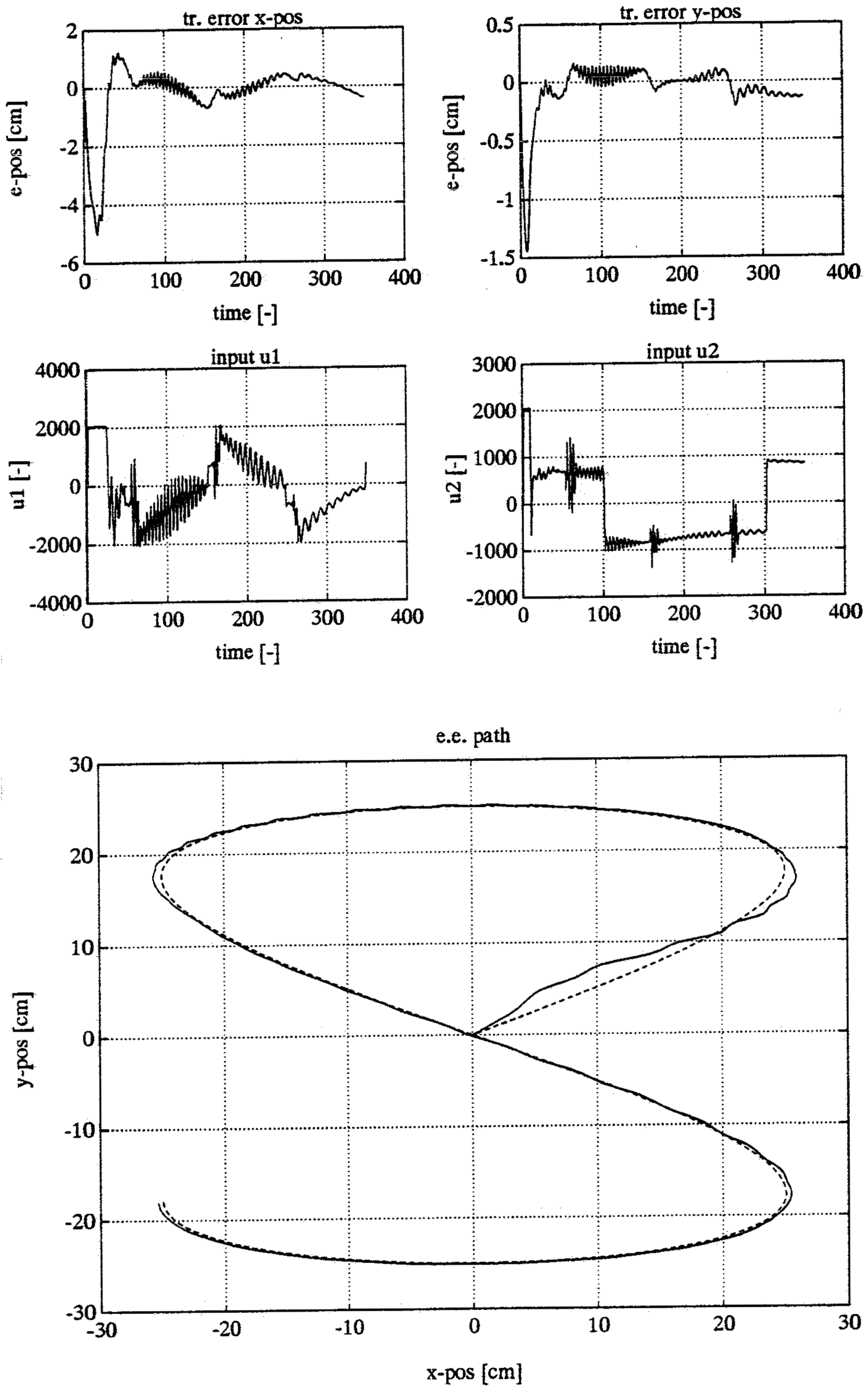
situation 6



simulation, desired trajectory: x_d= 25cos(3.14t) y_d= 25sin(3.14t),
k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 50 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -50, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0, a_x= a_y= 0.25

F: Results of the controlling of the end-effector of the xy table

situation



simulation, desired trajectory: x_d= 25cos(3.14t) y_d= 25sin(1.57t),
k= 0.5 Nm/rad, control par.: k_1= 0.05, k_2= 0.05, k_3= 0.05, λ_1= 50 and λ_2= 25
meas. freq.: 1/0.01 Herz, observ. poles: -25, position t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi_1(0)$= 0, $\Delta\varphi_3(0)$= 0
velocity t=0: $\Delta\varphi_2(0)$= 0, $\Delta\varphi 1(0)$= 0, $\Delta\varphi_3(0)$= 0, a_x= a_y= 0.25

APPENDIX G: REFERENCES

Asada, H. and J. J. E. Slotine (1986)
Robot analysis and control
John Wiley Sons, New York

Popov, V. M. (1969)
Hyperstability of control systems
Springer, New York

Slotine, J. J. E. and S. Hong (1987)
Two-time scale sliding control of manipulators with flexible joints
American Control Reference, pp. 805-810

Lammerts, I. M. M. (1991)
Adaptive composite computed torque control of robots with elastic motor transmissions
Article, Eindhoven University of Technology

Molengraft, M. J. G. van de (1990)
Identification of non-linear mechanical systems
ISBN 90-9003596-6, Eindhoven University of Technology

Heeren, T. A. G. (1989)
On control of manipulators
ISBN 90-9002789-0, Eindhoven University of Technology

Kok, J. J. (1990)
Regelen 2, Regelen 3, Regelen 4
Lecture notes, Eindhoven University of Technology

Kammer, R (1987)
Numerieke methoden voor technici
ISBN 90 6674 801 X