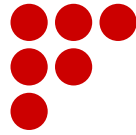
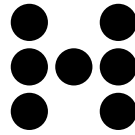


CARINTHIA  
UNIVERSITY  
OF APPLIED  
SCIENCES



FACHHOCHSCHULE  
KÄRNTEN



CARINTHIA UNIVERSITY OF APPLIED SCIENCES

DEGREE PROGRAM: SYSTEMS DESIGN

---

# LAB REPORT

”Robot Control”

---

Student:	Patrick HOFER
ID:	1410528006
Lecturer:	Dr. Andreja ROJKO
Submitted:	March 18, 2016
Submitted:	
Grade:	

# Contents

<b>List of Figures .....</b>	<b>3</b>
<b>List of Tables .....</b>	<b>4</b>
<b>List of Acronyms .....</b>	<b>5</b>
<b>1 Properties of robot .....</b>	<b>6</b>
1.1 Planar robot with 2 degrees of freedom .....	6
1.2 Simulation of robot dynamics .....	8
<b>2 Computed torque control .....</b>	<b>10</b>
2.1 PD controller for outer loop .....	11
2.2 PID outer loop controller .....	15
<b>3 Computed torque like controllers .....</b>	<b>18</b>
3.1 PD controller with gravity compensation .....	18
3.2 Classical joint control .....	20
<b>4 Practical issues at the realisation of robot control .....</b>	<b>27</b>
<b>5 CT like controller with disturbance estimator .....</b>	<b>30</b>
<b>6 Sliding mode control .....</b>	<b>34</b>
6.1 Ideal sliding mode controller .....	34
6.2 Boundary layer .....	38
<b>7 Adaptive control .....</b>	<b>40</b>

## List of Figures

1.1	Schematic of a planar robot with 2 degrees of freedom . . . . .	6
1.2	Verification of the model with $\mathbf{x}_0 = \left( \frac{-\pi}{2} \ 0 \ 0 \ 0 \right)^T$ . . . . .	8
1.3	Verification of the model with $\mathbf{x}_0 = \left( \frac{\pi}{2} \ 0 \ 0 \ 0 \right)^T$ . . . . .	9
1.4	Verification of the model with $\mathbf{x}_0 = \left( 0 \ 0 \ 0 \ 0 \right)^T$ . . . . .	9
2.1	Control loop with implement computed torque controller . . . . .	10
2.2	Simulation results of closed loop with PD controller with $\zeta = 1$ . . . . .	12
2.3	Simulation results of closed loop with PD controller with $\zeta = 0.1$ . . . . .	13
2.4	Simulation results of closed loop with PD controller with $\zeta = 10$ . . . . .	14
2.5	Simulation results of closed loop with PD controller with $\zeta = 1$ and added disturbance $\tau_D = 1 \text{ N m}$ . . . . .	14
2.6	Simulation results of closed loop with PID controller and disturbance . . . . .	16
2.7	Simulation results of closed loop with PID controller and unknown payload . . . . .	17
3.1	Control loop for the PD controller with gravity compensation . . . . .	18
3.2	Simulation results of PD controller with gravity compensation and $\omega_n = 100 \frac{\text{rad}}{\text{s}}$ . . . . .	19
3.3	Simulation results of PD controller with gravity compensation and $\omega_n = 10 \frac{\text{rad}}{\text{s}}$ . . . . .	20
3.4	Simulation results of classical PD controller with $\omega_n = 10 \frac{\text{rad}}{\text{s}}$ . . . . .	21
3.5	Simulation results of classical PD controller with $\omega_n = 25 \frac{\text{rad}}{\text{s}}$ . . . . .	22
3.6	Simulation results of classical PD controller with $\omega_n = 50 \frac{\text{rad}}{\text{s}}$ . . . . .	23
3.7	Simulation results of classical PID controller with $\omega_n = 10 \frac{\text{rad}}{\text{s}}$ . . . . .	25
3.8	Simulation results of classical PID controller with $\omega_n = 50 \frac{\text{rad}}{\text{s}}$ . . . . .	26
3.9	Simulation results of classical PID controller with $\omega_n = 50 \frac{\text{rad}}{\text{s}}$ and torque saturation . . . . .	26
4.1	Simulation results of CT PD controller with a sampling time $T_s = 1 \text{ ms}$ . . . . .	27
4.2	Simulation results of CT PD controller with a sampling time $T_s = 50 \text{ ms}$ . . . . .	28
4.3	Simulation results of CT PD controller with a sampling time $T_s = 100 \text{ ms}$ . . . . .	29
5.1	Schematic of the controller structure for $i^{\text{th}}$ joint . . . . .	30
5.2	Simulation results of CT like controller with disturbance estimator with $l_1 = l_2 = 20$ and $\frac{\omega_{\text{estim}}}{\omega_n} = 2$ . . . . .	31
5.3	Simulation results of CT like controller with disturbance estimator with $l_1 = l_2 = 100$ and $\frac{\omega_{\text{estim}}}{\omega_n} = 2$ . . . . .	32
5.4	Simulation results of CT like controller with disturbance estimator with $l_1 = l_2 = 100$ and $\frac{\omega_{\text{estim}}}{\omega_n} = 10$ . . . . .	33
6.1	Simulation results of sliding mode controller with $\lambda_i = 10$ and $k_i = 20$ . . . . .	35
6.2	Simulation results of sliding mode controller with $\lambda_i = 10$ and $k_i = 10$ . . . . .	36
6.3	Simulation results of sliding mode controller with $\lambda_i = 25$ and $k_i = 10$ . . . . .	36
6.4	Simulation results of sliding mode controller with $\lambda_i = 10$ , $k_i = 20$ and $\hat{\mathbf{g}} = 0$ . . . . .	37
6.5	Simulation results of sliding mode controller with $\lambda_i = 10$ , $k_i = 30$ and $\hat{\mathbf{g}} = 0$ . . . . .	38
6.6	Simulation results of sliding mode controller with $\lambda_i = 10$ , $k_i = 20$ and $\tanh(\mathbf{s})$ . . . . .	39
6.7	Simulation results of sliding mode controller with $\lambda_i = 10$ , $k_i = 40$ and $\tanh(5\mathbf{s})$ . . . . .	39
7.1	Simulation results of adaptive controller with $\lambda_i = 5$ , $\gamma_i = 10$ and $k_{v,i} = 100$ . . . . .	41
7.2	Simulation results of adaptive controller with $\lambda_i = 5$ , $\gamma_i = 10$ and $k_{v,i} = 10$ . . . . .	42

## List of Tables

2.1	Controller parameters for simulations witch PD outer loop controller . . . . .	11
3.1	Controller parameters for simulations witch classical PD controller . . . . .	21
3.2	Controller parameters for simulations witch classical PID controller . . . . .	24
5.1	Controller parameters for simulations witch CT like controller with disturbance estimator . . . . .	31

## List of Acronyms

**CT** Computed torque

**DOF** degrees of freedom

**SMC** Sliding Mode Control

# 1 Properties of robot

## 1.1 Planar robot with 2 degrees of freedom

The first task is to derive the equation of motion for a 2 degrees of freedom (DOF) planar robot using Lagrange formalism. The schematic of such a robot is shown in Figure 1.1. This robot consists of two drives, connected by links which have the mass at the tip of them.

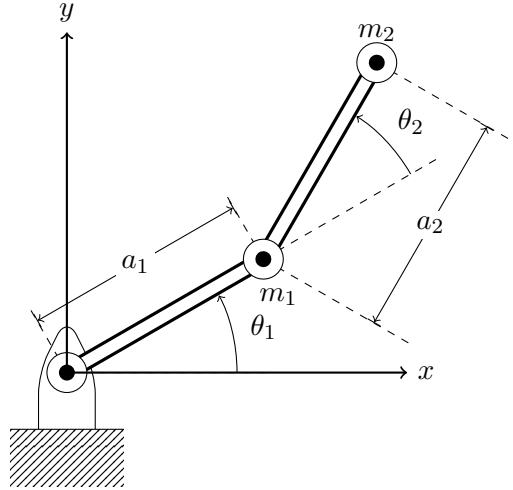


Figure 1.1: Schematic of a planar robot with 2 degrees of freedom

The generalised coordinates of the system are the angles  $\theta_1$  and  $\theta_2$ , afterwards denoted by  $q_1$  and  $q_2$  respectively. As the masses are assumed to be point masses, the according position vectors can be used to derive the kinetic and the potential energy. The position and velocity vectors are given by:

$$\begin{aligned} \mathbf{r}_1 &= \begin{pmatrix} a_1 \cos(q_1) \\ a_1 \sin(q_1) \end{pmatrix} & \mathbf{r}_2 &= \begin{pmatrix} a_1 \cos(q_1) + a_2 \cos(q_1 + q_2) \\ a_1 \sin(q_1) + a_2 \sin(q_1 + q_2) \end{pmatrix} \\ \dot{\mathbf{r}}_1 &= \begin{pmatrix} -a_1 \sin(q_1) \dot{q}_1 \\ a_1 \cos(q_1) \dot{q}_1 \end{pmatrix} & \dot{\mathbf{r}}_2 &= \begin{pmatrix} -a_1 \sin(q_1) \dot{q}_1 - a_2 \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \\ a_1 \cos(q_1) \dot{q}_1 + a_2 \cos(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \end{pmatrix} \end{aligned}$$

The kinetic energy  $T$  of the robot arm is described by the following equation:

$$\begin{aligned} T &= \frac{m_1}{2} \dot{\mathbf{r}}_1^T \dot{\mathbf{r}}_1 + \frac{m_2}{2} \dot{\mathbf{r}}_2^T \dot{\mathbf{r}}_2 \\ &= \frac{a_1^2(m_1 + m_2) + a_2^2 m_2}{2} \dot{q}_1^2 + \frac{a_2^2 m_2}{2} \dot{q}_2^2 + a_2 m_2 (a_2 \dot{q}_2 + a_1 \cos(q_2) \dot{q}_1 + a_1 \cos(q_2) \dot{q}_2) \dot{q}_1 \end{aligned} \quad (1.1)$$

The potential energy  $V$  is given by:

$$V = m_1 g y_1 + m_2 g y_2 = a_1 m_1 g \sin(q_1) + m_2 g (a_1 \sin(q_1) + a_2 \sin(q_1 + q_2)) \quad (1.2)$$

The Lagrangian  $L$ , which is needed for the Lagrange formalism, is defined as the difference of the kinetic energy  $T$  and the potential energy  $V$ :

$$L = T - V \quad (1.3)$$

The Lagrange formalism is given by the following equation:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} = Q_k \quad (1.4)$$

where:

$$\begin{aligned} q_k &: k^{\text{th}} \text{ generalised coordinate} \\ \dot{q}_k &: k^{\text{th}} \text{ generalised velocity} \\ Q_k &: k^{\text{th}} \text{ generalised force} \end{aligned}$$

Inserting Equations 1.1 and 1.2 into 1.3 and applying the result on 1.4, results in the differential equations, which describe the movement of the planar robot. The resulting system description has the following form:

$$\tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \tau_D \quad (1.5)$$

where:

$$\begin{aligned} \tau &= \text{input torque vector} \\ \mathbf{M}(\mathbf{q}) &= \text{inertia matrix} \\ \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) &= \text{centrifugal/corriolis vector} \\ \mathbf{g}(\mathbf{q}) &= \text{gravitation vector} \\ \tau_D &= \text{disturbance vector} \end{aligned}$$

Often, the term  $\mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$  is gathered and denoted by  $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$ . The result of the calculation is given by:

$$\begin{aligned} \tau = & \begin{pmatrix} a_1^2 m_1 + a_1^2 m_2 + a_2^2 m_2 + 2a_1 a_2 m_2 \cos(q_2) & a_2 m_2 (a_2 + a_1 \cos(q_2)) \\ a_2 m_2 (a_2 + a_1 \cos(q_2)) & a_2^2 m_2 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} \\ & + \begin{pmatrix} -a_1 a_2 m_2 \dot{q}_2 \sin(q_2) (2\dot{q}_1 + \dot{q}_2) \\ a_1 a_2 m_2 \dot{q}_1^2 \sin(q_2) \end{pmatrix} \\ & + \begin{pmatrix} g m_2 (a_2 \cos(q_1 + q_2) + a_1 \cos(q_1)) + a_1 g m_1 \cos(q_1) \\ a_2 g m_2 \cos(q_1 + q_2) \end{pmatrix} \end{aligned}$$

In this result the disturbance vector  $\tau_D$  was set to zero, because it is unknown.

As a next step, the direct kinematics and the Jacobian matrix have to be found. The direct kinematic was already calculated in the beginning and is denoted by  $\mathbf{r}_2$ :

$$\mathbf{r}_2 = \begin{pmatrix} a_1 \cos(q_1) + a_2 \cos(q_1 + q_2) \\ a_1 \sin(q_1) + a_2 \sin(q_1 + q_2) \end{pmatrix}$$

The Jacobian is the partial derivative of the position vector with respect to the generalised coordinates:

$$\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{r}_2}{\partial \mathbf{q}} = \begin{pmatrix} -a_2 \sin(q_1 + q_2) - a_1 \sin(q_1) & -a_2 \sin(q_1 + q_2) \\ a_2 \cos(q_1 + q_2) + a_1 \cos(q_1) & a_2 \cos(q_1 + q_2) \end{pmatrix}$$

For all further considerations, the following state vector is introduced:

$$\mathbf{x} = \begin{pmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

The next step is to check the plausibility of the equations of motion by simulation.

## 1.2 Simulation of robot dynamics

To check, if the previously derived dynamic equations are plausible, some simulations are done. For these simulations, the following values are used for the parameters:

$$\begin{aligned} m_1 &= m_2 = 1 \text{ kg} \\ a_1 &= a_2 = 1 \text{ m} \\ g &= 9.81 \frac{\text{m}}{\text{s}^2} \end{aligned}$$

As a first simulation, the stable, downright equilibrium point is used as initial condition. The expected result is, that no movement occurs, as the system always stays in its equilibrium. For the next simulation, the unstable, upper equilibrium is used. Again, no movement is to be expected, as the initial condition is again an equilibrium point and no external force is applied. As a last simulation, an initial condition of  $\mathbf{x}_0 = (0 \ 0 \ 0 \ 0)^T$  is used. The expected result is some kind of chaotic oscillation of both joints.

Figure 1.2 shows the results of the simulation with the stable equilibrium point as initial condition. As expected, the position remains zero for the whole simulation. The small values of the velocities, somewhere around  $10^{-14}$ , are caused by numerical errors.

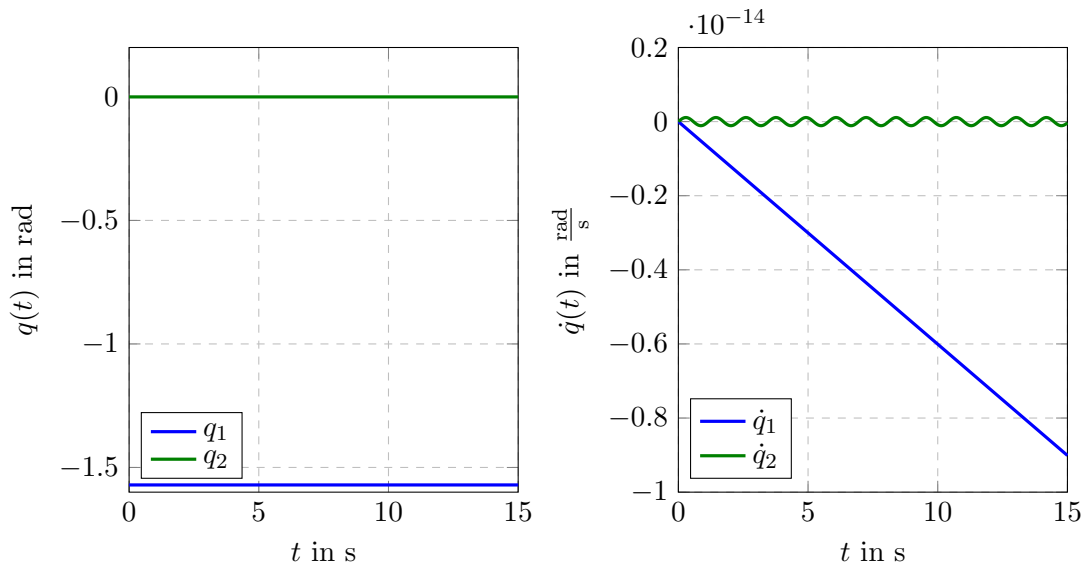


Figure 1.2: Verification of the model with  $\mathbf{x}_0 = \left( -\frac{\pi}{2} \ 0 \ 0 \ 0 \right)^T$

Figure 1.3 shows the simulation results where both joints start at the upper equilibrium point. Until approximately 7s, no movement can be seen. Then suddenly, both joints fall and start to oscillate chaotic afterwards. This result is also caused by numerical errors. As the error is integrated, it increases with every calculation step and eventually results in the seen behaviour.



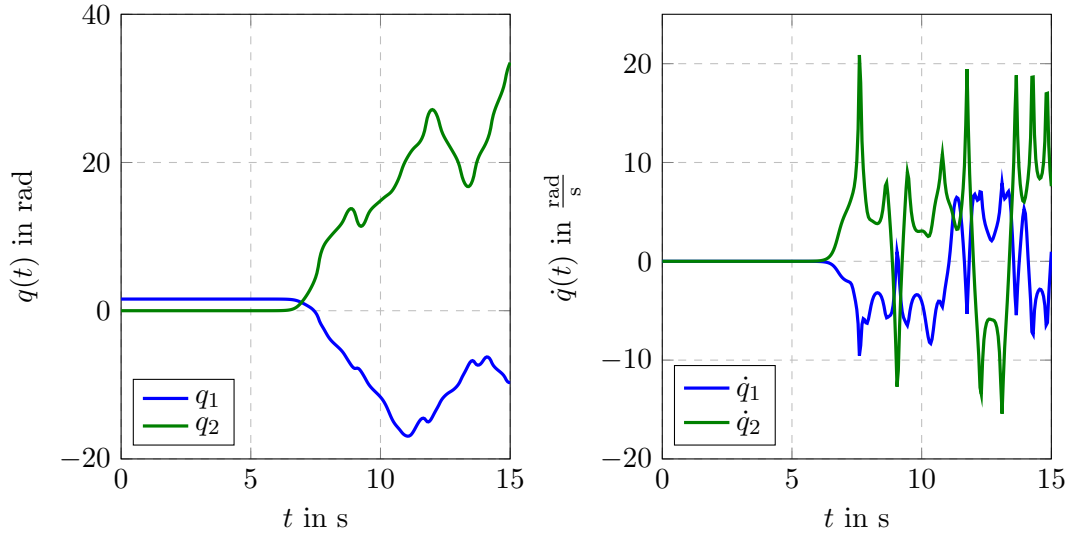


Figure 1.3: Verification of the model with  $\mathbf{x}_0 = \left( \frac{\pi}{2} \ 0 \ 0 \ 0 \right)^T$

The result for the simulation with all initial conditions set to zero is seen in Figure 1.4. As expected, both joints oscillate in a chaotic behaviour. This oscillation continues, as no friction was implemented in the derived model.

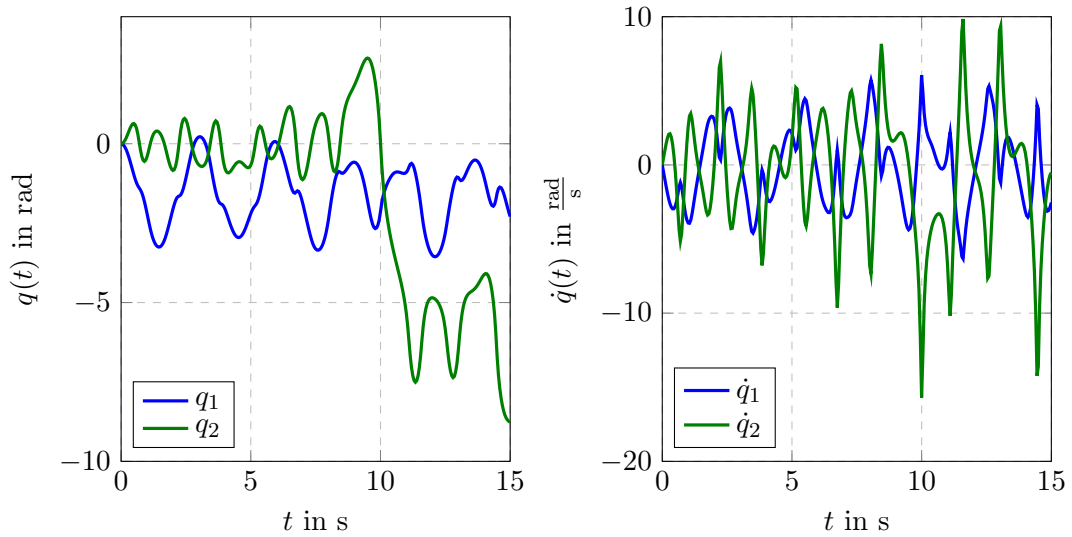


Figure 1.4: Verification of the model with  $\mathbf{x}_0 = \left( 0 \ 0 \ 0 \ 0 \right)^T$

As the simulation results fit with the expectations, the derived dynamic model can be seen as correct. The next step is to design a controller for the motion of the robot.

## 2 Computed torque control

A Computed torque (CT) controller is a nonlinear controller, which linearises the error system and applies then a linear control law for the stabilisation of the closed loop system. As a first step, the tracking error system need to be defined:

$$\begin{aligned} \mathbf{e} &= \mathbf{q}_D - \mathbf{q} \\ \dot{\mathbf{e}} &= \dot{\mathbf{q}}_D - \dot{\mathbf{q}} \\ \ddot{\mathbf{e}} &= \ddot{\mathbf{q}}_D - \ddot{\mathbf{q}} \end{aligned}$$

where:

**e** : error vector

$\mathbf{q}_D$  : vector with reference trajectories for each joint

Substituting the acceleration vector  $\ddot{\mathbf{q}}$  with the robot dynamics (Equation 1.5) results in

$$\ddot{\mathbf{e}} = \ddot{\mathbf{q}}_D - \mathbf{M}^{-1} (\boldsymbol{\tau} - \mathbf{v} - \mathbf{g}) + \mathbf{M}^{-1} \boldsymbol{\tau}_D$$

Defining  $\mathbf{u} = \ddot{\mathbf{q}}_D - \mathbf{M}^{-1}(\boldsymbol{\tau} - \mathbf{v} - \mathbf{g})$  and  $\mathbf{w} = \mathbf{M}^{-1}\boldsymbol{\tau}_D$  results in the following linear system

$$\begin{pmatrix} \dot{\mathbf{e}} \\ \ddot{\mathbf{e}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{w}$$

As the actual input of the robot arm is not the input  $\mathbf{u}$ , but the motor torque  $\tau$ ,  $\mathbf{u} = \ddot{\mathbf{q}}_D - \mathbf{M}^{-1}(\tau - \mathbf{v} - \mathbf{g})$  needs to be solved for  $\tau$ :

$$\tau = \mathbf{M}(\ddot{\mathbf{q}}_D - \mathbf{u}) + \mathbf{v} + \mathbf{g} \quad (2.1)$$

Equation 2.1 shows the CT control law for the inner feedforward loop. This is the linearising control law, which nullifies the nonlinear parts of the robot dynamics and reduces the system to a linear double integrator system. The dynamics of the closed loop can be defined by choosing a specific linear controller for  $\mathbf{u}$ . Figure 2.1 shows a schematic of the implementation of a CT controller, with the linearising feedforward controller and the linear control law in the outer loop.

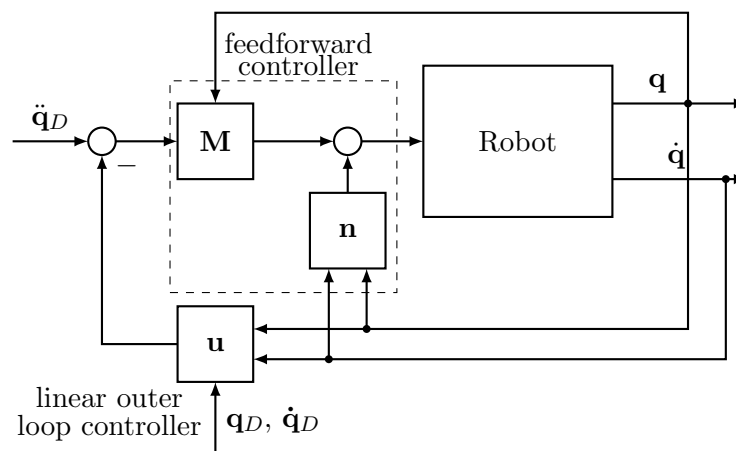


Figure 2.1: Control loop with implement computed torque controller

In the following two sections, two different control laws for  $\mathbf{u}$  are designed and the behaviour of the closed loop system is simulated. For the simulations, the following two reference trajectories are used:

$$\begin{aligned} q_{1,D} &= 0.1 \sin(\pi t) \\ q_{2,D} &= 0.1 \cos(\pi t) \end{aligned}$$

## 2.1 PD controller for outer loop

As first step, a PD control law of the form

$$\mathbf{u} = -\mathbf{K}_d \dot{\mathbf{e}} - \mathbf{K}_p \mathbf{e}$$

where:

$\mathbf{K}_d$  : diagonal matrix with derivative gains  
 $\mathbf{K}_p$  : diagonal matrix with proportional gains

To find the stability range for the two gains, the linear system for the  $i^{th}$  joint is considered.

$$\begin{pmatrix} \dot{e}_i \\ \ddot{e}_i \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -k_d & -k_p \end{pmatrix} \begin{pmatrix} e_i \\ \dot{e}_i \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} w_i$$

The characteristic polynomial of the closed loop is then given by:

$$\Delta(s) = s^2 + k_{d,i}s + k_{p,i} \stackrel{!}{=} 0$$

The roots of this polynomial are negative if all coefficients have the same sign. Therefore, the controlled system is stable if  $k_{d,i}$  and  $k_{p,i}$  are both chosen positive. To set the two parameters, the characteristic polynomial is compared with a general second order polynomial:

$$p(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$$

By comparing the two polynomials, the parameters of the controller can be set with  $k_{d,i} = 2\zeta\omega_n$  and  $k_{p,i} = \omega_n^2$ .  $\zeta$  describes the damping factor and  $\omega_n$  the undamped, natural frequency. For the first simulation of the controller, the damping factor is set to  $\zeta = 1$ . This results in a faster system response than by choosing  $\zeta > 1$  and has no overshoot, compared with  $\zeta < 1$ . To show the effects of over- and under-damping, two additional simulations are done. For the frequency  $\omega_n = 10 \frac{rad}{s}$  is used. A last simulation shows the influence of disturbances on the control behaviour. Table 2.1 shows the controller parameters and the according Figure numbers, which show the simulation results.

Table 2.1: Controller parameters for simulations with PD outer loop controller

	$k_{p,i}$	$k_{d,i}$	$\tau_D$	According Figure
$\zeta = 1$ :	100	20	0 N m	2.2
$\zeta = 0.1$ :	100	2	0 N m	2.3
$\zeta = 10$ :	100	200	0 N m	2.4
$\zeta = 1$ :	100	20	1 N m	2.5

Figure 2.2 shows the simulation results for the CT controller with PD controller and critical damping. In the upper two plots, the angular positions  $\mathbf{q}$  and the applied torques of both joints can be seen. Joint 1 has an average input torque of 30 N m and a peak torque of 80 N m at the beginning. The other joint has gets an average torque of 10 N m and a peak torque of 30 N m at the beginning. These values are very high and a very strong motor would be needed to be able to apply such torques. The bottom left sub-figure shows the tracking error of the two joints. The error of the first joint starts at zero, has a small peak of approximately 0.001 rad, and then converges to zero at 0.5 s. The second joint starts with an error of 0.1 rad, which is caused by the cosine input signal. The error also converges to zero at approximately 0.5 s. The last plot shows the velocity error, which also converges to zero for both joints.

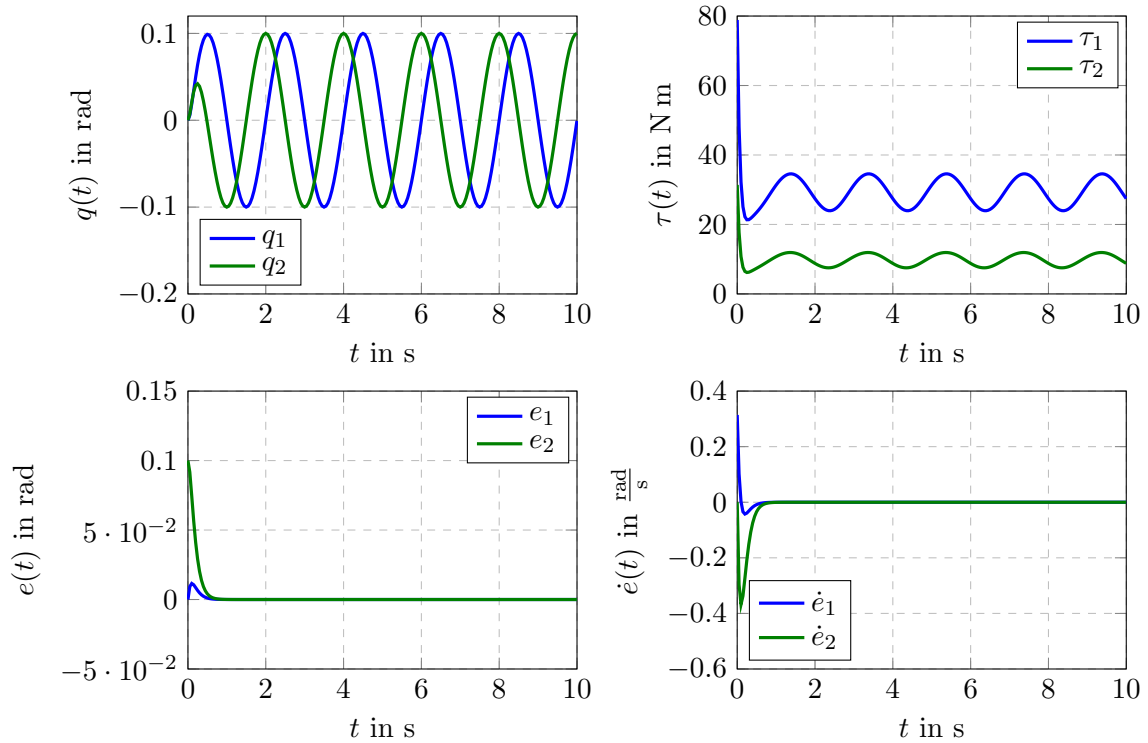


Figure 2.2: Simulation results of closed loop with PD controller with  $\zeta = 1$

Figure 2.3 shows the results of the simulation with reduced damping factor. The upper left plot also shows the angular position of both joints. At the beginning of the simulation, some oscillations can be seen. In the sub plot, which shows the tracking error of both joints, the oscillating behaviour of the weak damped second order polynomial can be seen. This also shows, that it takes approximately 4 s for the errors to settle. Even though the transient behaviour is slower, the applied peak torques are lower than in the previous simulation. The first joint has a peak torque of 50 N m and the second joint needs a peak torque of 20 N m.

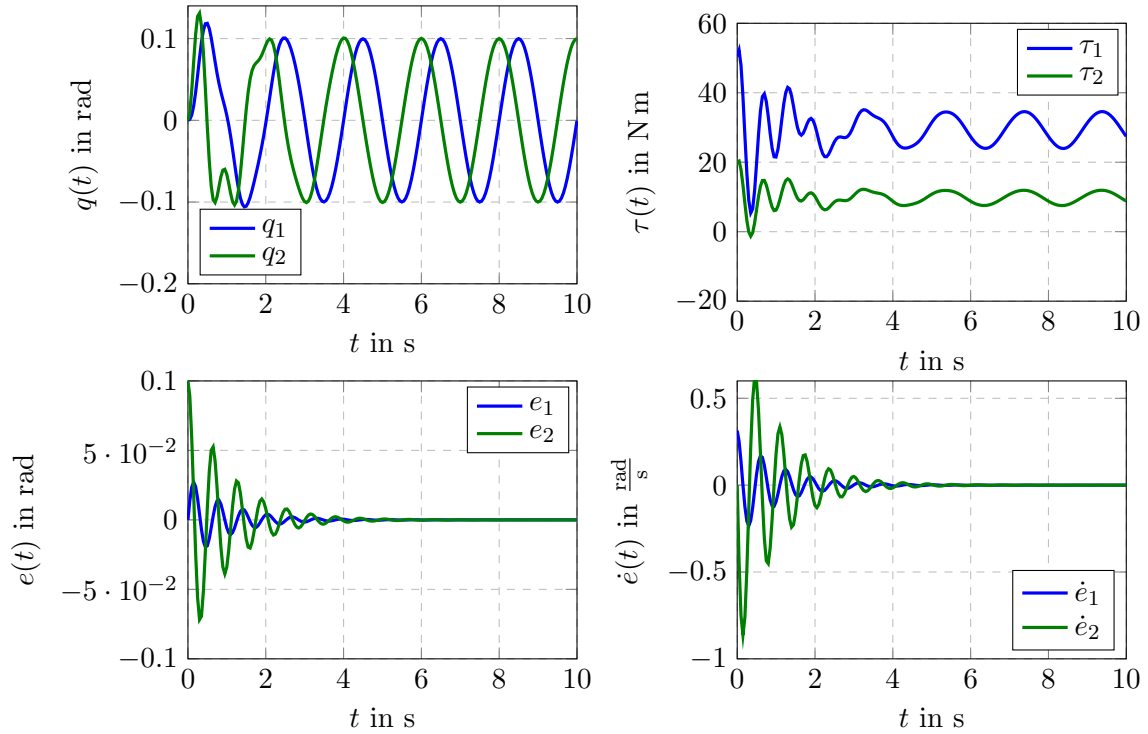


Figure 2.3: Simulation results of closed loop with PD controller with  $\zeta = 0.1$

In Figure 2.4, the results of the simulation with a damping factor of  $\zeta = 10$  are shown. It can be seen, that the tracking error of the second joint takes over 8 s to converge towards zero, which is much slower, than with critical damping. Also, the initial torque has a much higher peak with over 350 N m for the first joint and approximately 150 N m for the second joint. This means, that the transient response of the system is much slower and the input torque is much higher, than when critical damping is used.

Figure 2.5 shows the simulation results for the PD controller with critical damping and a disturbance, which is applied on each joint, of  $\tau_D = 1$  N m. The transient response of the system is the same, as in the first simulation. On the other hand, the disturbance causes a steady-state error. In order to remove this steady-state error, an integrating part needs to be added to the open loop system.

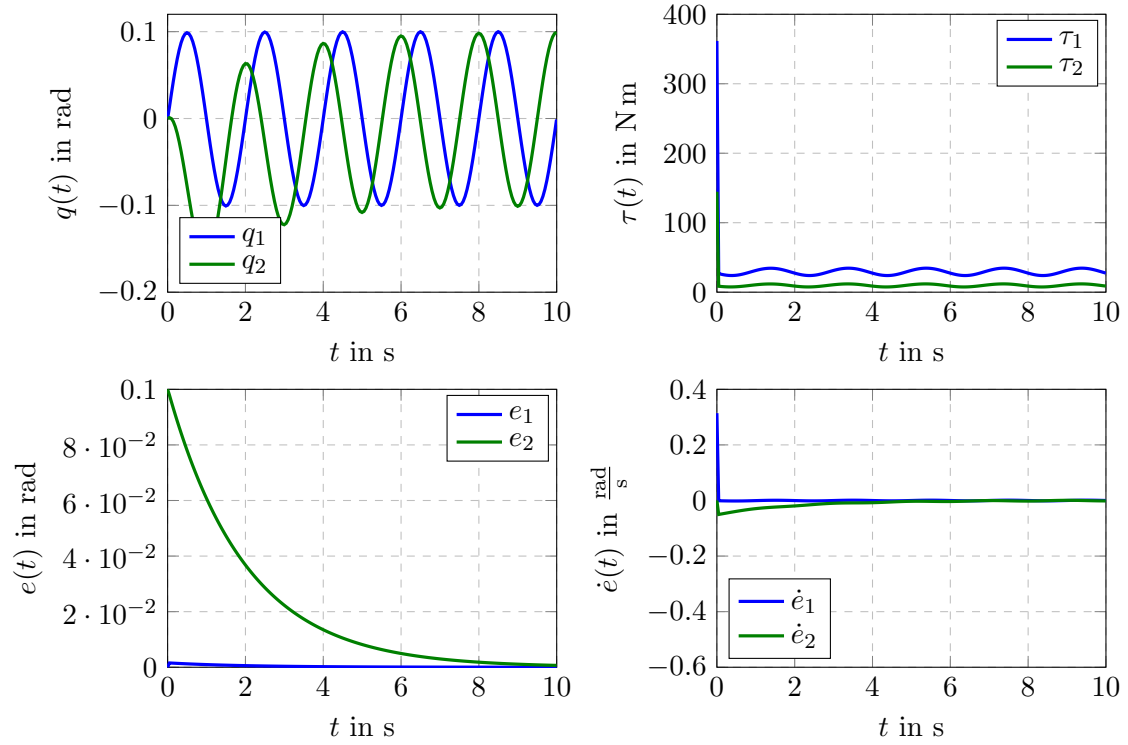


Figure 2.4: Simulation results of closed loop with PD controller with  $\zeta = 10$

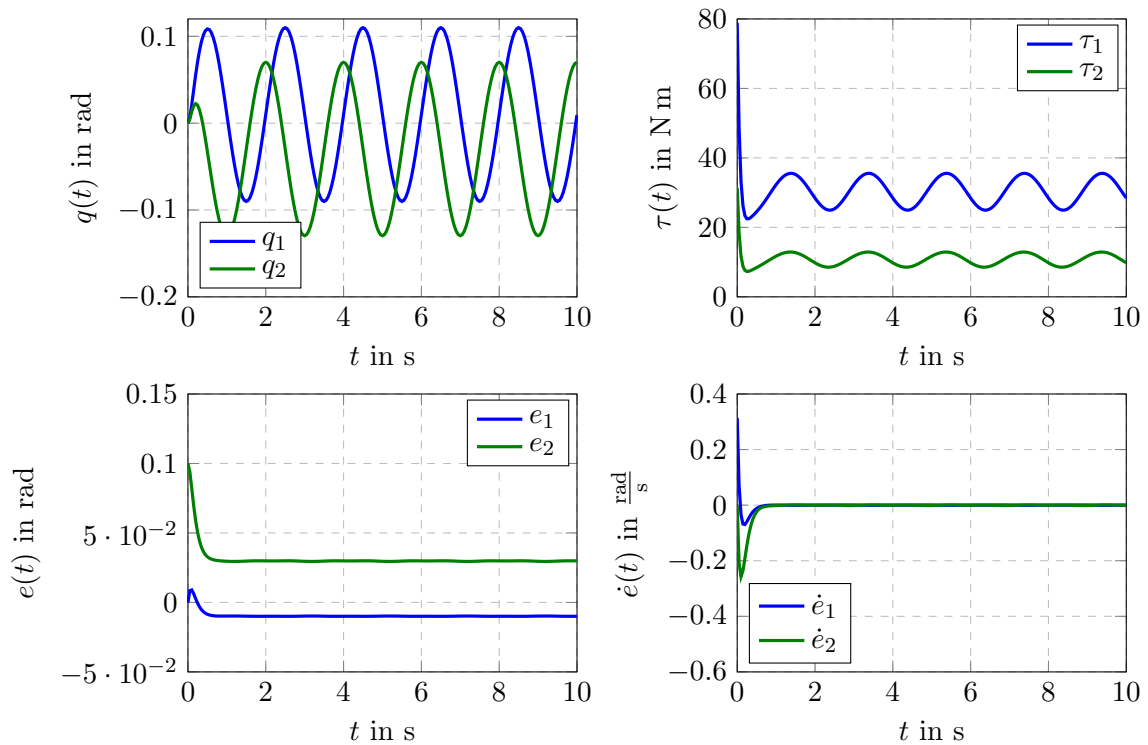


Figure 2.5: Simulation results of closed loop with PD controller with  $\zeta = 1$  and added disturbance  $\tau_D = 1 \text{ Nm}$

In the next section, a PID controller is used as an outer loop controller, to overcome the steady-

state error which is caused by the external disturbance.

## 2.2 PID outer loop controller

In order to apply an integrator to the open loop system, a new state is introduced in the state space description of the error dynamics:

$$\begin{pmatrix} \dot{\varepsilon} \\ \dot{\mathbf{e}} \\ \ddot{\mathbf{e}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \varepsilon \\ \mathbf{e} \\ \dot{\mathbf{e}} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u} + \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{w}$$

where:

$$\varepsilon : \int \mathbf{e}(\tau) d\tau$$

This leads to a new outer loop control law, which includes the new state variable  $\varepsilon$ :

$$\mathbf{u} = -\mathbf{K}_d \dot{\mathbf{e}} - \mathbf{K}_p \mathbf{e} - \mathbf{K}_i \varepsilon$$

Before implementing a controller with PID structure, the stability condition for the closed loop system need to be defined. Therefore, again the system of the  $i^{th}$  link is used. The characteristic polynomial of this link is given by:

$$\Delta(s) = s^3 + k_{d,i}s^2 + k_{p,i}s + k_{i,i} \stackrel{!}{=} 0$$

To calculate the conditions, for which this polynomial only contains negative roots, the Routh-Hurwitz stability criterion is used.

$$\begin{array}{ccc} 1 & k_{p,i} & 0 \\ k_{d,i} & k_{i,i} & 0 \\ \frac{k_{d,i}k_{p,i} - k_{i,i}}{k_{d,i}} & 0 & 0 \\ k_{i,i} & 0 & 0 \end{array}$$

In order, that the first row has no changes in sign, the following conditions need to be fulfilled:

$$\begin{aligned} k_{d,i} &> 0 \\ k_{i,i} &> 0 \\ k_{p,i} &> \frac{k_{i,i}}{k_{d,i}} \end{aligned}$$

For the controller design, an integral gain  $k_{i,i} = 500$  is given. To define the other two parameter,  $k_{p,i}$  and  $k_{d,i}$ , the general second order polynomial is used and a third, stable pole  $a$  is added:

$$\begin{aligned} p(s) &= (s^2 + 2\zeta\omega_n s + \omega_n^2)(s + a) \\ &= s^3 + (2\zeta\omega_n + a)s^2 + (2\zeta\omega_n a + \omega_n^2)s + \omega_n^2 a \stackrel{!}{=} s^3 + k_{d,i}s^2 + k_{p,i}s + k_{i,i} \end{aligned}$$

By comparing the coefficients of the two polynomials, the parameters can be defined.  $\zeta$  is again set to 1, for critical damping, and  $\omega_n$  is set to  $8 \frac{\text{rad}}{\text{s}}$ . This results in  $k_{p,i} = 2520$  and  $k_{d,i} = 100.2$ . For the simulation of the controller behaviour, first a disturbance torque  $\tau_D = 1 \text{ N m}$  is applied. Also, a second simulation shows the behaviour of the controller if an unknown payload of  $0.5 \text{ kg}$  is added to the robot.

Figure 2.6 shows the simulation results of the CT controller with PID outer loop controller. Similar to the last simulation with the PD outer loop controller, a disturbance torque of 1 N m is added to both joints. Due to the integrating part of the controller, the steady state error of the closed loop system is reduced to 0 rad. The applied torque for the first joint ranges from 25, N m up to 35, N m and for the second joint a torque in the range of 8, N m up to 13, N m is used.

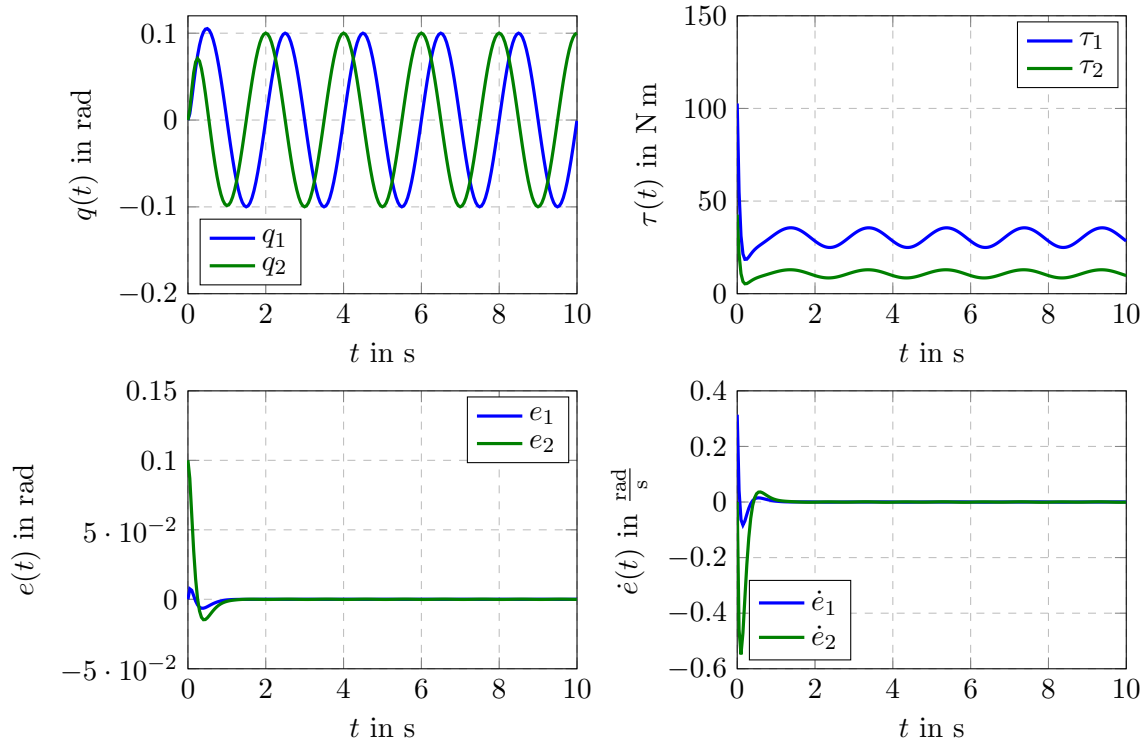


Figure 2.6: Simulation results of closed loop with PID controller and disturbance

Figure 2.7 shows the simulation results with an unknown payload of 0.5 kg added. Compared to the previous simulation, the position error of the joints does not converge to 0 rad anymore. The reason for this is, that the inner loop of the CT controller does not match the nonlinearities of the system ideally and therefore, a nonlinear behaviour remains in the closed loop system. The error, which results from these nonlinearities can also not be compensated by the integrating part of the outer loop controller. As the mass of the second joint is higher, because of the unknown payload, the needed torque is also increased. The torque for the first joint now ranges from 32, N m up to 48, N m and for the second joint it ranges from 12, N m up to 19, N m. This simulation shows, that for small steady state error a precise model of the system is needed.



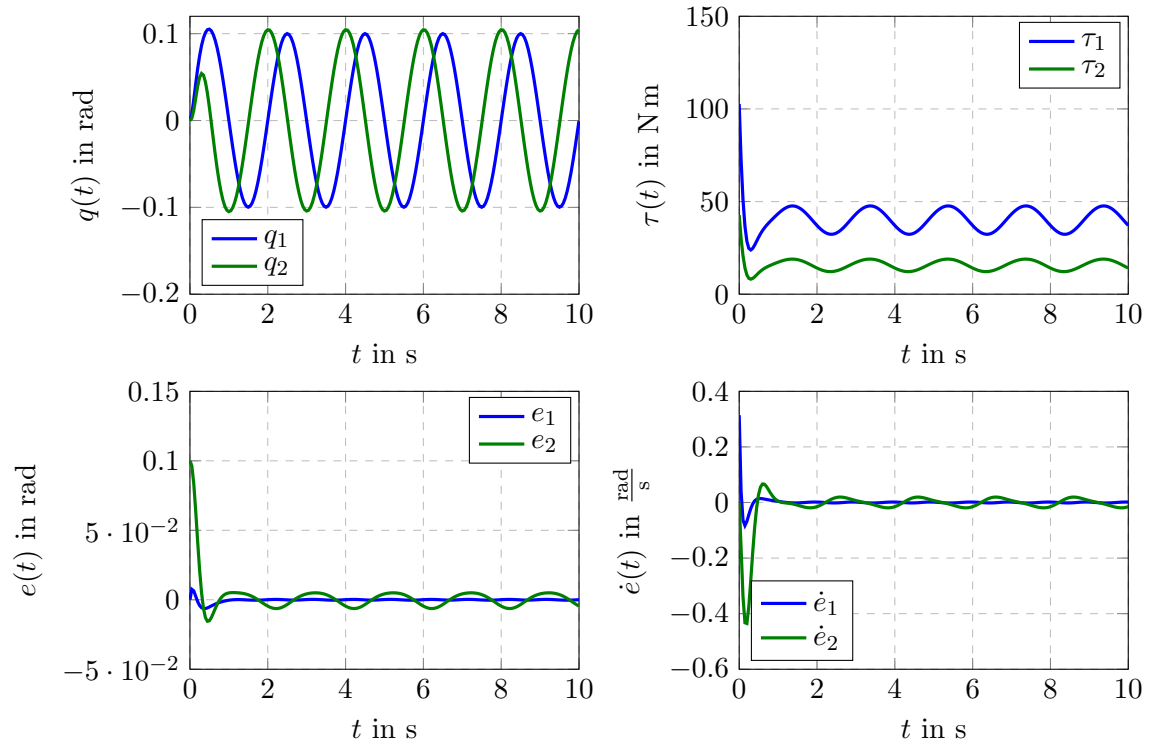


Figure 2.7: Simulation results of closed loop with PID controller and unknown payload

In the next chapter, a derivation of the CT controllers is described. Also, some simulations regarding this derivations will be done.

### 3 Computed torque like controllers

The CT like controllers are a derivation of the original CT controller. For the CT control law, it is assumed that the dynamic model of the system is completely known. This is almost never the case on a real robot, because of some dynamic components, which are hard to model, like friction. Therefore, the CT control law is changed from the perfect robot model to the estimated model. In this chapter, first a PD controller with gravity compensation will be explained and afterwards two different types of classical, decoupled joint control are shown.

#### 3.1 PD controller with gravity compensation

For this CT like controller, the moment of inertia and the centrifugal/coriolis terms are neglected and only the gravitational part is considered. This results in a much simpler control law, which still offers good results for  $\dot{\mathbf{q}}(t) = 0$ . This is, because in steady state only the gravitation has an influence on the behaviour of the system. For the stabilisation of the system, the outer loop still contains a PD controller. The overall control law for the PD controller with gravity compensation is given by:

$$\tau_c = -\mathbf{u} + \mathbf{g}(\mathbf{q})$$

The schematic for this control structure is shown in Figure 3.1. The inner feedforward loop only contains the gravitation compensation, while the outer loop still implements the linear control law  $\mathbf{u}$ .

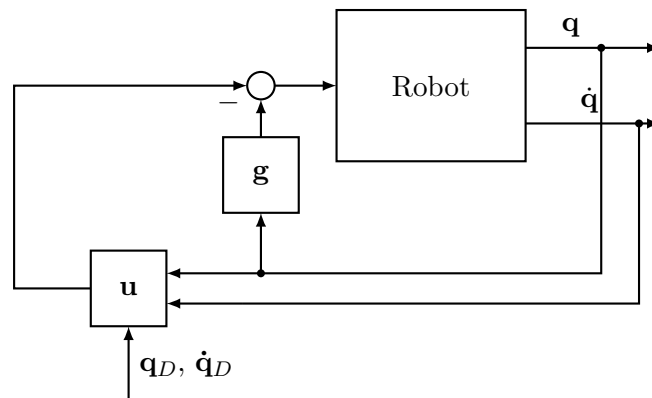


Figure 3.1: Control loop for the PD controller with gravity compensation

For the design of the PD controller, again the general second order polynomial is used. This means, the controller parameters can be chosen by setting  $\zeta$  and  $\omega_n$ . Again, critical damping is used, so  $\zeta = 1$ . To show the influence of the parameter  $\omega_n$ , two different values are used. The first simulation is done with  $\omega_n = 100 \frac{\text{rad}}{\text{s}}$ , while for the second  $\omega_n = 10 \frac{\text{rad}}{\text{s}}$  is used.

Figure 3.2 shows the simulation with  $\omega_n = 100 \frac{\text{rad}}{\text{s}}$ . It can be seen, that the position error of the closed loop system is somewhere around  $0.1 \cdot 10^{-3} \text{ rad}$ . Also, the system is already settled after approximately 0.2s. On the other hand, the initial torque of the second joint is approximately 1 kNm. After the system is settled, the torque for the first joint oscillates from 24 Nm up to 35 Nm and for the second joint it ranges from 7.5 Nm to 12 Nm.

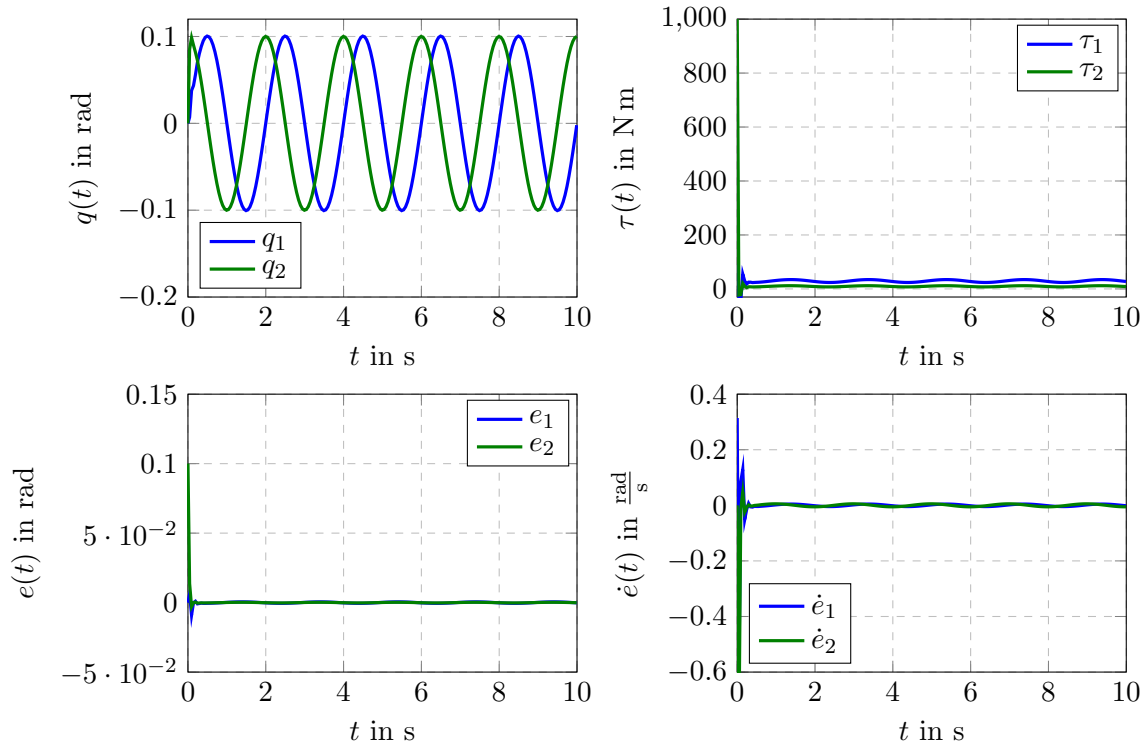


Figure 3.2: Simulation results of PD controller with gravity compensation and  $\omega_n = 100 \frac{\text{rad}}{\text{s}}$

Figure 3.3 show the results of the simulation with the PD controller with gravity compensation with  $\omega_n = 10 \frac{\text{rad}}{\text{s}}$ . Compared to the previous simulation, the position error is much higher, as it has an oscillation with an amplitude of  $60 \cdot 10^{-3} \text{ rad}$  at the first joint and an oscillation with an amplitude of  $25 \cdot 10^{-3} \text{ rad}$  at the second joint. Also, the system takes approximately 1 s to settle. The initial torque, which is applied to the joints, is much lower than in the previous simulation. The second joint has a short peak of approximately 20 N m, which is much more realistic. After the system is settled, the oscillation of the first joint ranges from 21 N m to 38 N m and for the second joint it ranges from 6.5 N m up to 13 N m.

These simulations show, that even if  $\dot{\mathbf{q}} \neq 0$  the results of the controller are quite good, if the parameters of the PD controller are high enough. On the other hand, increasing the gains can lead to actuator saturation, which is also not wanted. Therefore, on a real robot, a compromise which fulfils the needed requirements has to be made.

The next section describes the decoupled control of the two joints with a standard PD/PID controller.

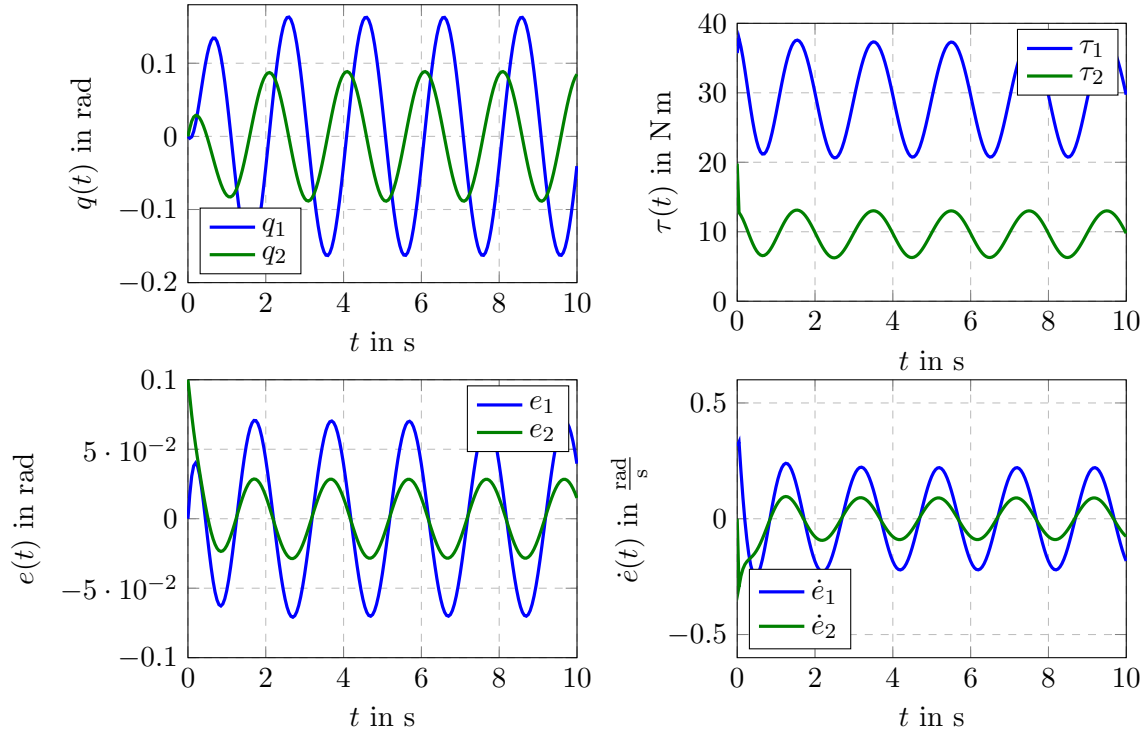


Figure 3.3: Simulation results of PD controller with gravity compensation and  $\omega_n = 10 \frac{\text{rad}}{\text{s}}$

### 3.2 Classical joint control

In the classical joint control, even more simplifications are made. While the PD controller with gravity compensation still included the gravitational part, this controller does not cover any nonlinearities of the system and simply implements the linear control law. This means, the control law only includes the linear part  $\mathbf{u}$ :

$$\tau_c = -\mathbf{u}$$

Therefore, this controller results in one decoupled control law per joint. This method works pretty good for motors with very high gear ratio, as higher gear ratio decrease the influence of the nonlinear effects and reduces the control problem to the control of the drives. The model of the joints is then simplified to:

$$J_i \ddot{q}_i + B_i \dot{q}_i = u_i - \frac{1}{N_i} \tau_{D,i} \quad (3.1)$$

where:

- $J_i$  : moment of inertia of  $i^{th}$  joint
- $B_i$  : viscous friction coefficient
- $N_i$  : gear ratio

In this section, first a PD controller is applied for both joints and afterwards a PID controller is used.

## PD control

For the design of the PD controller, the general second order polynomial is used. Again, critical damping is used for the controller design. The control law for the  $i^{th}$  joint is:

$$u_i = k_{d,i}\dot{e}_i + k_{p,i}e_i$$

To show the influence of the frequency  $\omega_n$ , three simulations with different values are performed. Table 3.1 summarises the controller parameters and the according figures. Both joints use the same controller parameters.

Table 3.1: Controller parameters for simulations with classical PD controller

	$k_{p,i}$	$k_{d,i}$	According Figure
$\omega_n = 10 \frac{\text{rad}}{\text{s}}$ :	100	20	3.4
$\omega_n = 25 \frac{\text{rad}}{\text{s}}$ :	625	50	3.5
$\omega_n = 50 \frac{\text{rad}}{\text{s}}$ :	2500	100	3.6

In Figure 3.4, the results of the simulation with the classical PD controller with  $\omega_n = 10 \frac{\text{rad}}{\text{s}}$  can be seen. The position error, when using this controller, is very high, with an approximate average of 0.3 rad for the first joint and 0.1 rad for the second joint. This high offset is caused by the gravitation, which was neglected in the controller design. The applied torque has no high peak at the beginning and an oscillation from 21 N m up to 34 N m for the first joint and ranges from 7 N m to 11 N m for the second joint.

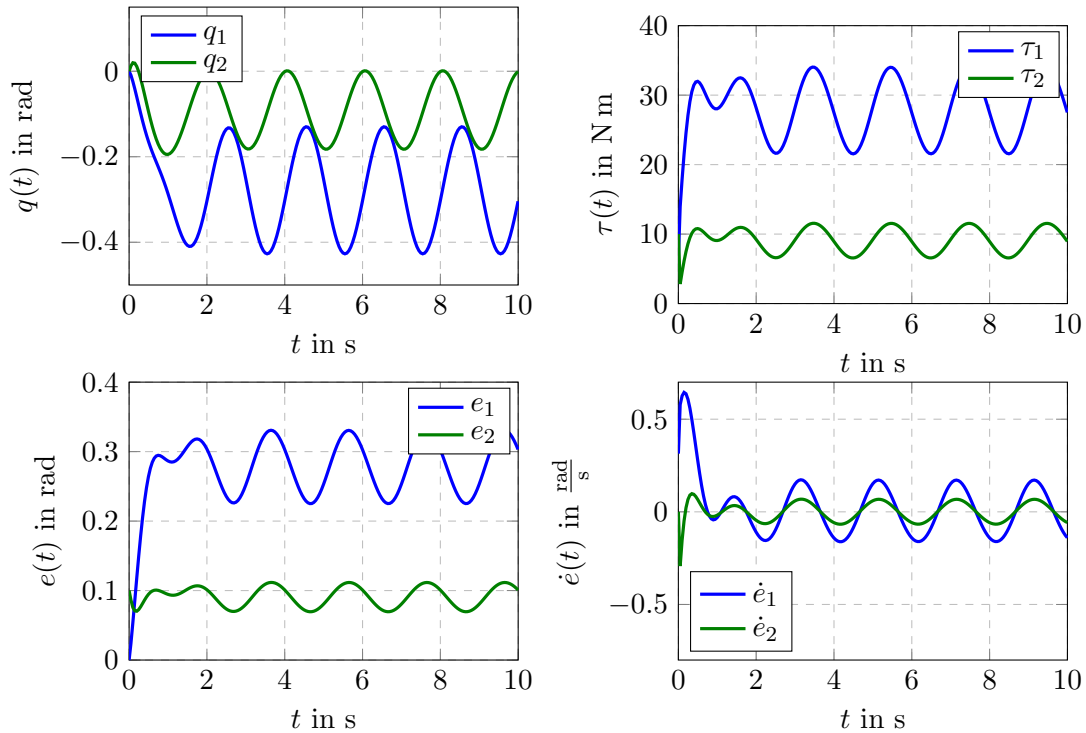


Figure 3.4: Simulation results of classical PD controller with  $\omega_n = 10 \frac{\text{rad}}{\text{s}}$

Figure 3.5 shows the simulation results for the decoupled PD controller with  $\omega_n = 25 \frac{\text{rad}}{\text{s}}$ . Compared to the previous results, with the lower gains, the position error is smaller with an average value of 0.05 rad for the first joint and 0.19 rad for the second joint. The amplitude of the velocity error is also smaller than in the previous simulation. On the other hand, the initial torque already shows a high peak of approximately 60 N m. After the system is settled, the amplitudes of the torque are similar to the previous simulation.

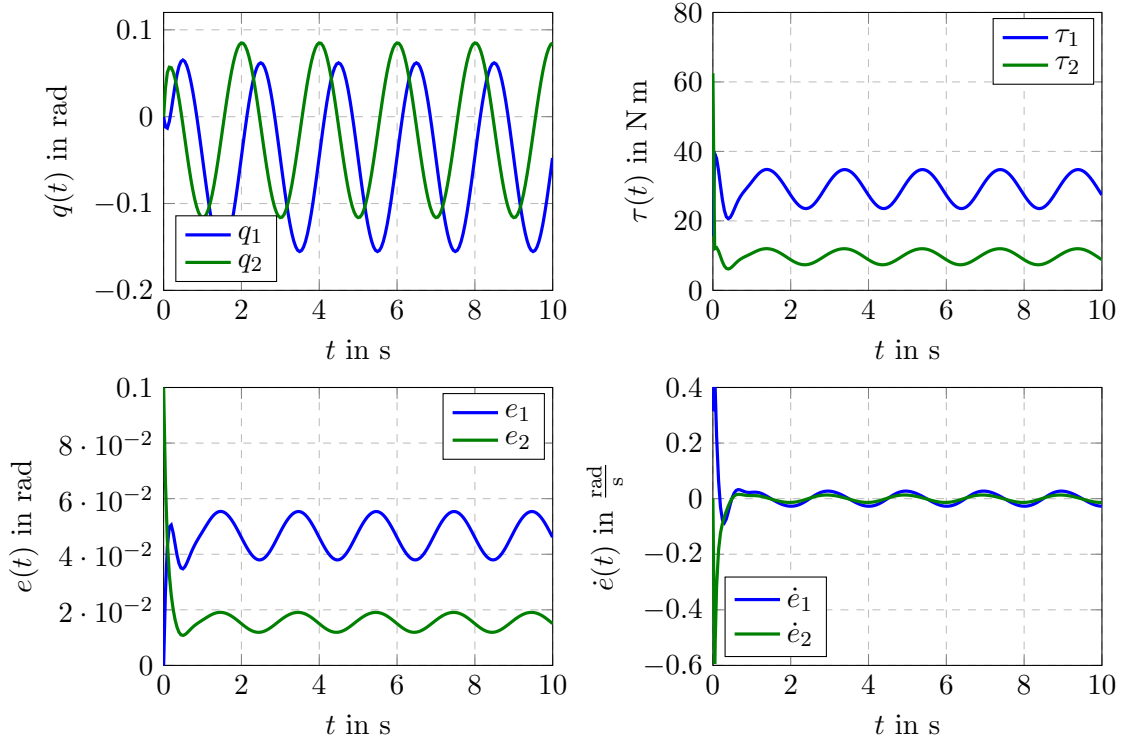


Figure 3.5: Simulation results of classical PD controller with  $\omega_n = 25 \frac{\text{rad}}{\text{s}}$

The simulation results for the high gains, with  $\omega_n = 50 \frac{\text{rad}}{\text{s}}$ , can be seen in Figure 3.6. The position and velocity errors are even smaller than in the simulation with  $\omega_n = 25 \frac{\text{rad}}{\text{s}}$ . The average position error is decreased to 0.01 rad for the first joint and to 0.003 rad for the second joint. On the other hand, the initial torque peak is much higher with an approximate value of 250 N m.

The next step is to implement an integrating part into this classical joint controller, to remove the steady state error, which is caused by the gravitation.

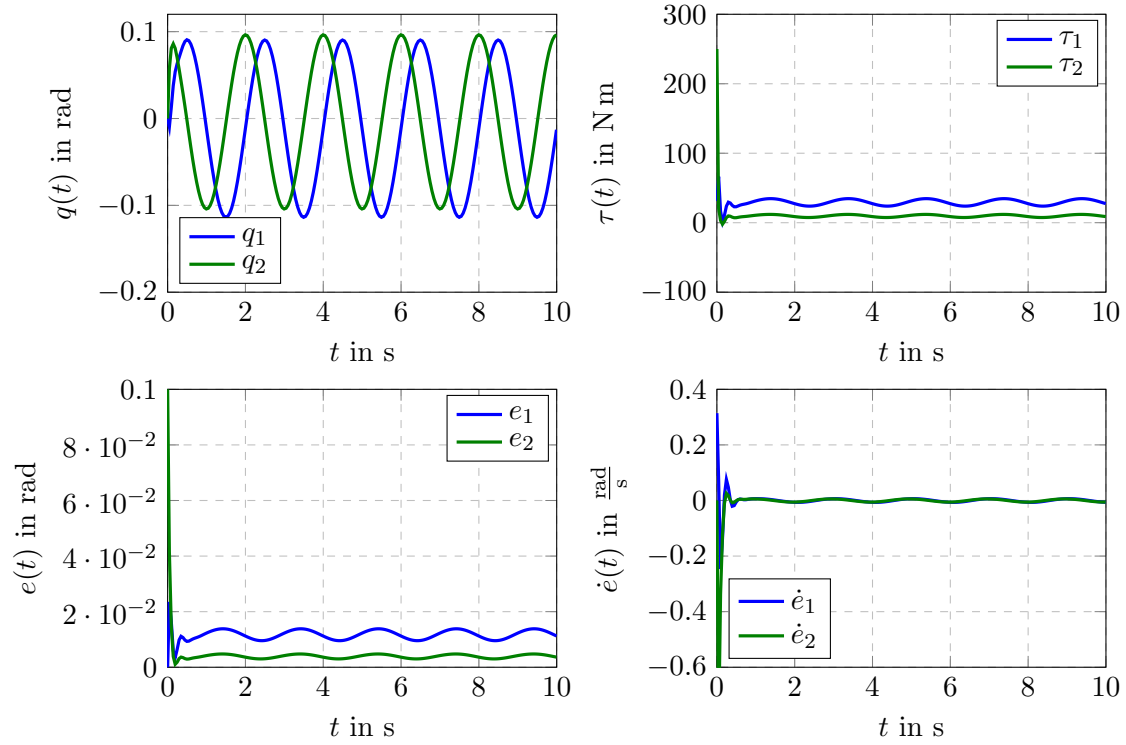


Figure 3.6: Simulation results of classical PD controller with  $\omega_n = 50 \frac{\text{rad}}{\text{s}}$

## PID control

To remove the steady state error, which existed when using the PD controller, an integrating part is included into the controller. The control law for the  $i^{\text{th}}$  joint is now:

$$u_i = k_{d,i}\dot{e}_i + k_{p,i}e_i + k_{i,i}\varepsilon_i$$

Inserting this controller in the equation of the linear system, as seen in Equation 3.1, results in:

$$\begin{aligned} J_i\ddot{q}_i + B_i\dot{q}_i &= k_{d,i}\dot{e}_i + k_{p,i}e_i + k_{i,i}\varepsilon_i - \frac{1}{N}\tau_{D,i} \\ &= k_{d,i}(\dot{q}_{D,i} - \dot{q}_i) + k_{p,i}(q_{D,i} - q_i) + k_{i,i} \int (q_{D,i} - q_i)dt - \frac{1}{N}\tau_{D,i} \\ \Rightarrow J_i\ddot{q}_i + (B_i + k_{d,i})\dot{q}_i + k_{p,i}q_i + k_{i,i} \int q_i dt &= k_{d,i}\dot{q}_{D,i} + k_{p,i}q_{D,i} + k_{i,i} \int q_{D,i} dt - \frac{1}{N}\tau_{D,i} \end{aligned}$$

The characteristic polynomial of this differential equation is:

$$\Delta(s) = J_i s^3 + (B_i + k_{d,i})s^2 + k_{p,i}s + k_{i,i}$$

To find the stability region for the controller parameters, the Routh-Hurwitz stability criterion is applied.

$J_i$	$k_{p,i}$	0
$B_i + k_{d,i}$	$k_{i,i}$	0
$\frac{(B_i + k_{d,i})k_{p,i} - J_i k_{i,i}}{B_i + k_{d,i}}$	0	0
$k_{i,i}$	0	0

As a moment of inertia is always positive,  $J_i > 0$ , the following conditions have to be fulfilled such that there is no change in sign of the first column, which implies that the closed loop system is stable:

$$\begin{aligned} B_i + k_{d,i} &> 0 \\ k_{i,i} &> 0 \\ (B_i + k_{d,i}) \frac{k_{p,i}}{J_i} &> k_{i,i} \end{aligned}$$

Again, the controller parameters for both joints are chosen to be the same. The parameter  $k_{i,i}$  is set to 1000. For the calculation of the other two controller parameters, the general second order polynomial is enhanced by a third, stable pole. Two simulations are made to show the influence of the frequency  $\omega_n$  and in a third simulation, the influence of an actuator saturation is analysed. In Table 3.2, the controller parameters and the used saturation for the different simulations are summarised.

Table 3.2: Controller parameters for simulations with classical PID controller

	$k_{p,i}$	$k_{d,i}$	saturation	According Figure
$\omega_n = 10 \frac{\text{rad}}{\text{s}}$ :	300	30	$\pm\infty \text{ N m}$	3.7
$\omega_n = 50 \frac{\text{rad}}{\text{s}}$ :	2540	100.4	$\pm\infty \text{ N m}$	3.8
$\omega_n = 50 \frac{\text{rad}}{\text{s}}$ :	2540	100.4	$\pm 35 \text{ N m}$	3.9

Figure 3.7 shows the simulation results for the classic PID controller with  $\omega_n = 10 \frac{\text{rad}}{\text{s}}$ . It can be seen, that the position error has no more offset and oscillates around zero for both joints. The amplitude of the oscillations is 0.02 rad for the first joint and 0.01 rad for the second joint. The peak torque of the first joint is 44 N m and for the second joint it is 30 N m.



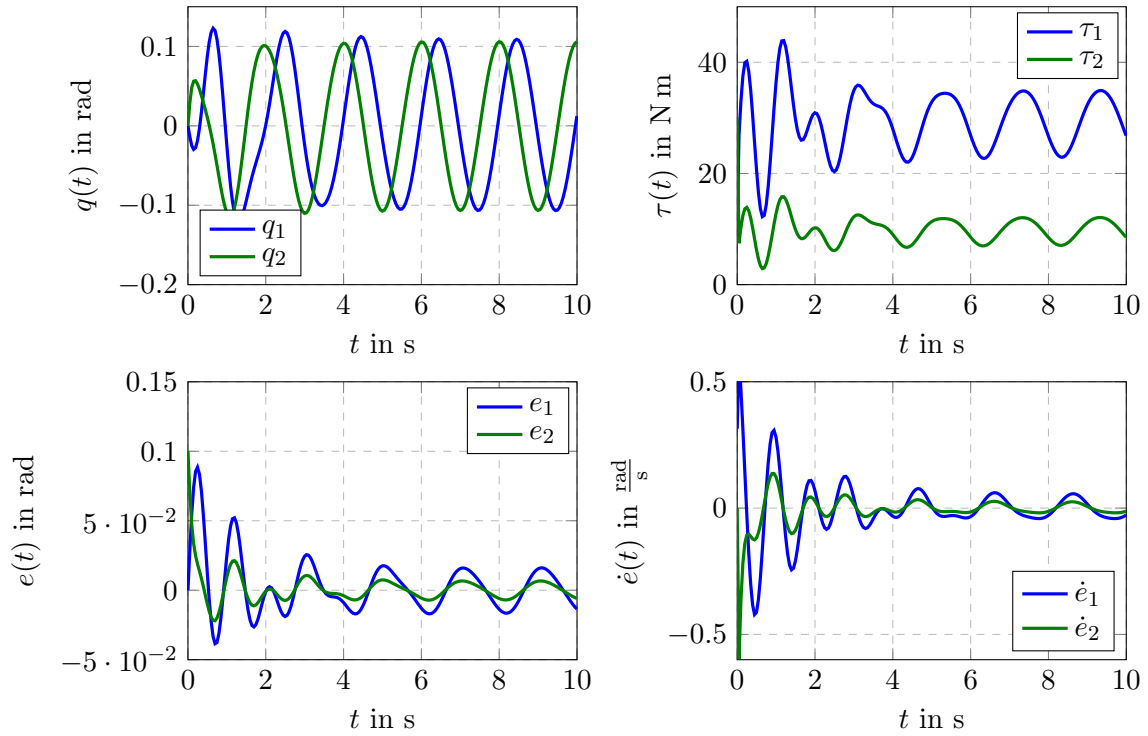


Figure 3.7: Simulation results of classical PID controller with  $\omega_n = 10 \frac{\text{rad}}{\text{s}}$

Figure 3.8 shows the simulation results for PID controller with  $\omega_n = 50 \frac{\text{rad}}{\text{s}}$ . With the higher parameters, the position error has a much smaller amplitude with 0.002 rad for the first joint and less than 0.001 rad for the second joint. On the other hand, the initial torque is very high, with a peak of approximately 250 N m.

In Figure 3.9, the simulation with the actuator saturation of  $\pm 35 \text{ N m}$  can be seen. The influence of the saturation can be seen during the first 0.7 s. The torque, which is applied to the first joint, is saturated during this time slot. Afterwards, the torque does not reach the saturation limit anymore. The position error during the settling process is also higher, when compared to the simulation with no saturation. After the system is settled, it behaves similar to the system with no saturation.

These simulation conclude the CT like controllers. The next chapter describes one of the problem that arises, when implementing a CT controller on a real system.

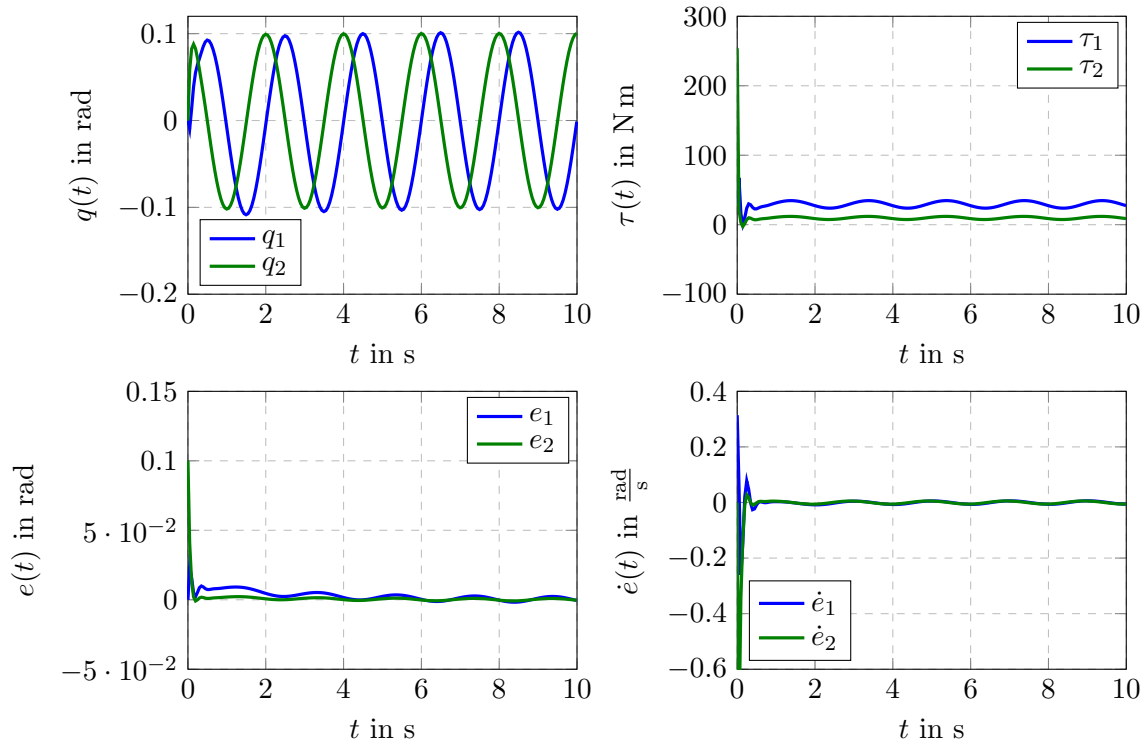


Figure 3.8: Simulation results of classical PID controller with  $\omega_n = 50 \frac{\text{rad}}{\text{s}}$

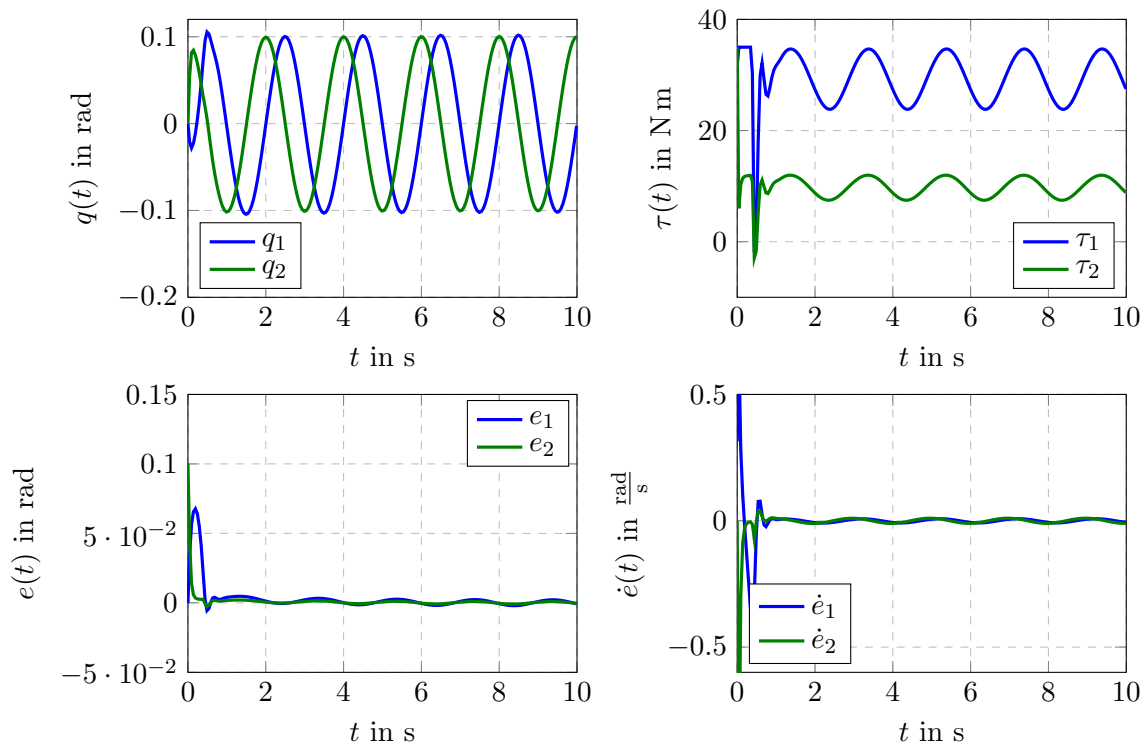


Figure 3.9: Simulation results of classical PID controller with  $\omega_n = 50 \frac{\text{rad}}{\text{s}}$  and torque saturation

## 4 Practical issues at the realisation of robot control

When implementing a CT controller, often a digital realisation of the controller is needed. For the design of a digital controller, the robot dynamics need to be discretised. This results in very complex dynamic equations for which the design of a controller is complicated. Also the discretisation of an existing controller cannot be applied to a CT controller, as this type of controller is a nonlinear one. Therefore, the transformation is done by approximation. This means, the continuous control law of the CT controller is used as its discrete equivalent. To show the influence of the used sampling time, some simulations are made. As controller, the CT controller with the PD controller in the outer loop is used, as already shown in section 2.1. The controller parameters are set to  $k_{p,i} = 100$  and  $k_{d,i} = 20$ . For the simulations, only the sampling time of the controller is changed, so that the simulated robot still used variable step size. In this way, it can still be seen as a continuous system. If the controller would be implemented on a real robot, this system would also be a continuous one. Therefore, the results obtained in this way of simulation are more realistic.

For the implementation of the discrete CT controller, different sampling times are used. The first simulation uses a sampling time of  $T_s = 1$  ms, the second a sampling time of  $T_s = 50$  ms and the third uses  $T_s = 100$  ms.

Figure 4.1 shows the results of the simulation with a settling time of  $T_s = 1$  ms. With this sampling time, the results are very similar to the original result, with the continuous control law, as it can be seen in Figure 2.2.

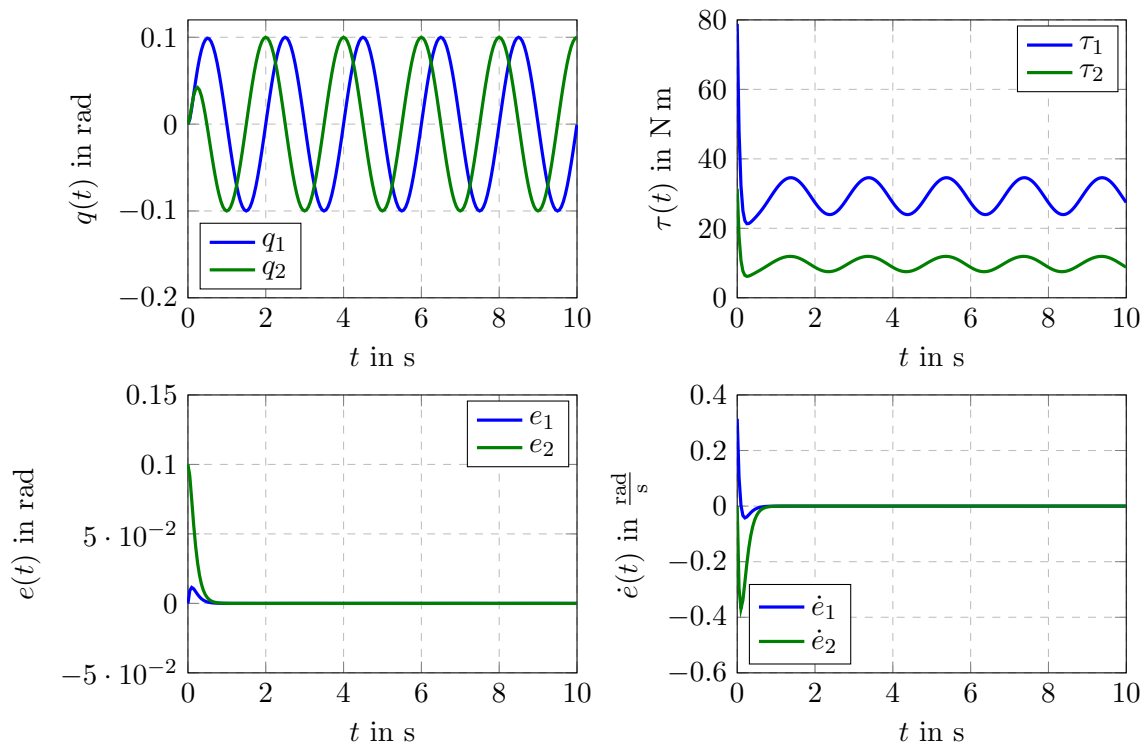


Figure 4.1: Simulation results of CT PD controller with a sampling time  $T_s = 1$  ms

Figure 4.2 shows the simulation results with a controller sampling time of  $T_s = 50$  ms. In the bottom left plot, which shows the position error, a small oscillation can be seen. This is caused by the quite high sampling time. For higher sampling times, this effect increases.

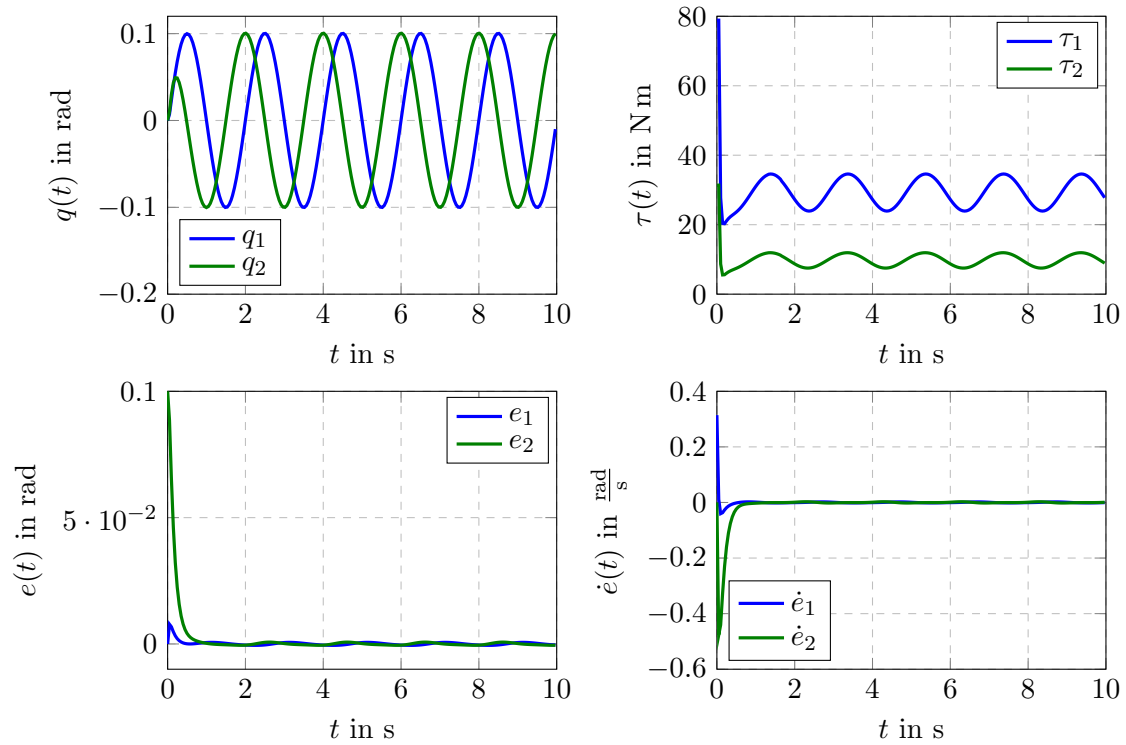


Figure 4.2: Simulation results of CT PD controller with a sampling time  $T_s = 50$  ms

In Figure 4.3, the results of the simulation with a sampling time of  $T_s = 100$  ms is shown. With this sampling time, the oscillation of the position error has already an increasing amplitude, which means, the closed loop system is already unstable. The problem is, that the controller only calculates the needed torque every 0.1 s. In this time, the robot has already moved further, as it would need to, therefore, the controller changes the sign of the torque. This means, the controller is too slow to control the robot, which causes unstable behaviour.

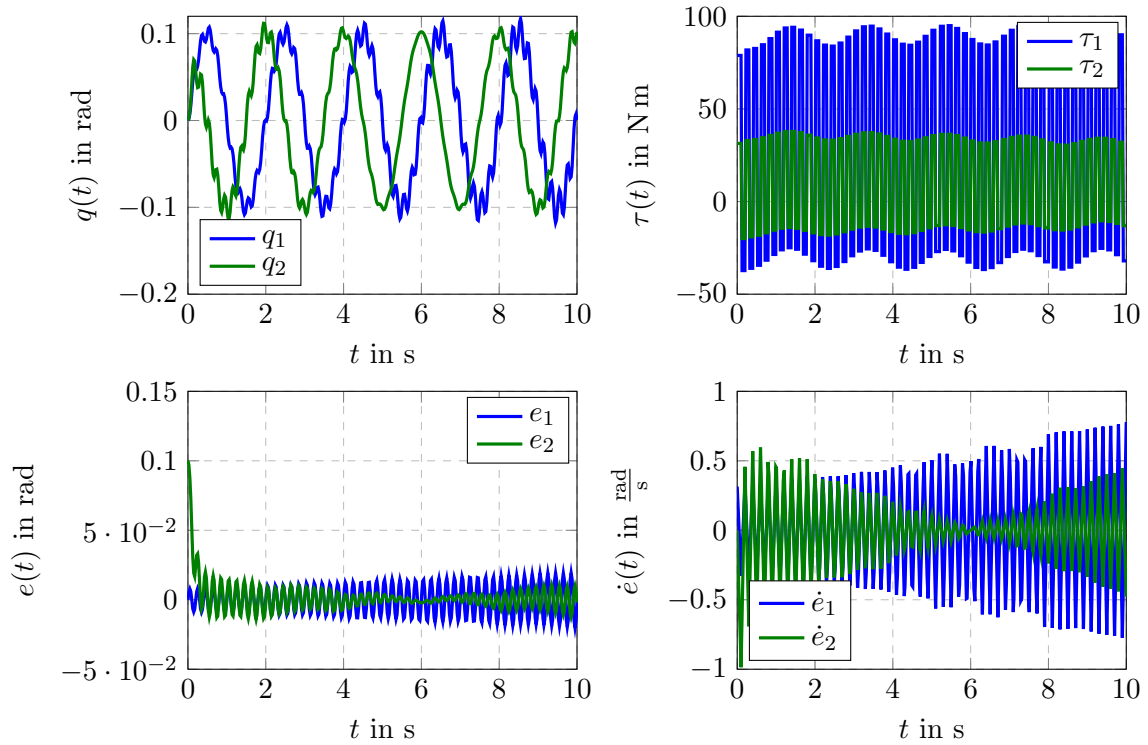


Figure 4.3: Simulation results of CT PD controller with a sampling time  $T_s = 100$  ms

In this chapter, the effects of the discretisation of a CT controller and the influence of the sampling time on the controller behaviour are shown. Optimally, the sampling time should be 1 ms or even less. If the sampling time is too high, the closed loop system starts to oscillate and if the sampling time is increased even further, it can result in unstable behaviour.

In the next chapter, another CT like controller is analysed for its behaviour when applied to the 2 DOF planar robot.

## 5 CT like controller with disturbance estimator

Similar to the classical joint controller is this a decoupled control law. This means, every joint can be controlled separately. This is an advantage, as every joint can be controlled by a separate processing unit and does not need the information of the other joints. To weaken the effects of the nonlinearities, which were completely neglected in the classical joint control approach, a disturbance estimator is included in this controller. The control law consists of the average moment of inertia of the corresponding joint, a PD controller for the stabilisation and the disturbance estimator to compensate the unknown or unmodelled dynamics. The control law for the  $i^{th}$  joint has the following form:

$$\tau_{c,i} = \bar{M}_{ii}\ddot{q}_{calc,i} + \hat{w}_i(q_i, \dot{q}_i)$$

where:

$$\begin{aligned} \bar{M}_{ii} &: \text{average moment of inertia of the joint} \\ \ddot{q}_{calc,i} &: \ddot{q}_{D,i} + k_{d,i}\dot{e}_i + k_{p,i}e_i \\ \hat{w}_i(q_i, \dot{q}_i) &: \text{estimated disturbances} \end{aligned}$$

For the estimation of the disturbances, a PI estimator of the following form is used:

$$\hat{w}_i(q_i, \dot{q}_i) = l_i(\dot{q}_{calc,i} - \dot{q}_i)$$

where:

$$l_i : \text{estimator parameter}$$

Figure 5.1 shows the block diagram of the controller structure for the  $i^{th}$  joint.

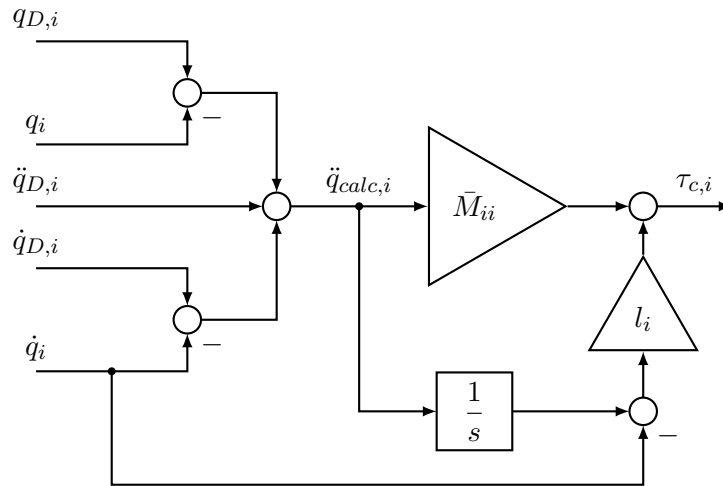


Figure 5.1: Schematic of the controller structure for  $i^{th}$  joint

When applying this controller, the error dynamics contain two frequencies. The first frequency is  $\omega_n$  which is defined by the PD controller. The second frequency  $\omega_{estim}$  is defined as  $\frac{l_i}{\bar{M}_{ii}}$ . Through the controller parameter  $l_i$ , this frequency can be set to different values. The peak position error depends on the ratio between the two frequencies.

To analyse the influence of the controller parameters, three simulations using different parameters were realised. In Table 5.1, the applied values for the three controllers are summarised.

Table 5.1: Controller parameters for simulations with CT like controller with disturbance estimator

$l_i$	$\frac{\omega_{estim}}{\omega_n}$	According Figure
20	2	5.2
100	2	5.3
100	10	5.4

Figure 5.2 shows the simulation results for  $l_1 = l_2 = 20$  and a ratio of 2. The position error of the first joint does not exceed a maximum value of 0.02 rad but takes approximately 20 s to settle. For the second joint, the error settles much faster, at approximately 5 s. The applied torque has a high initial peak and oscillates afterwards in the range of 24 N m and 35 N m for the first joint and from 7.5 N m up to 12 N m for the second joint.

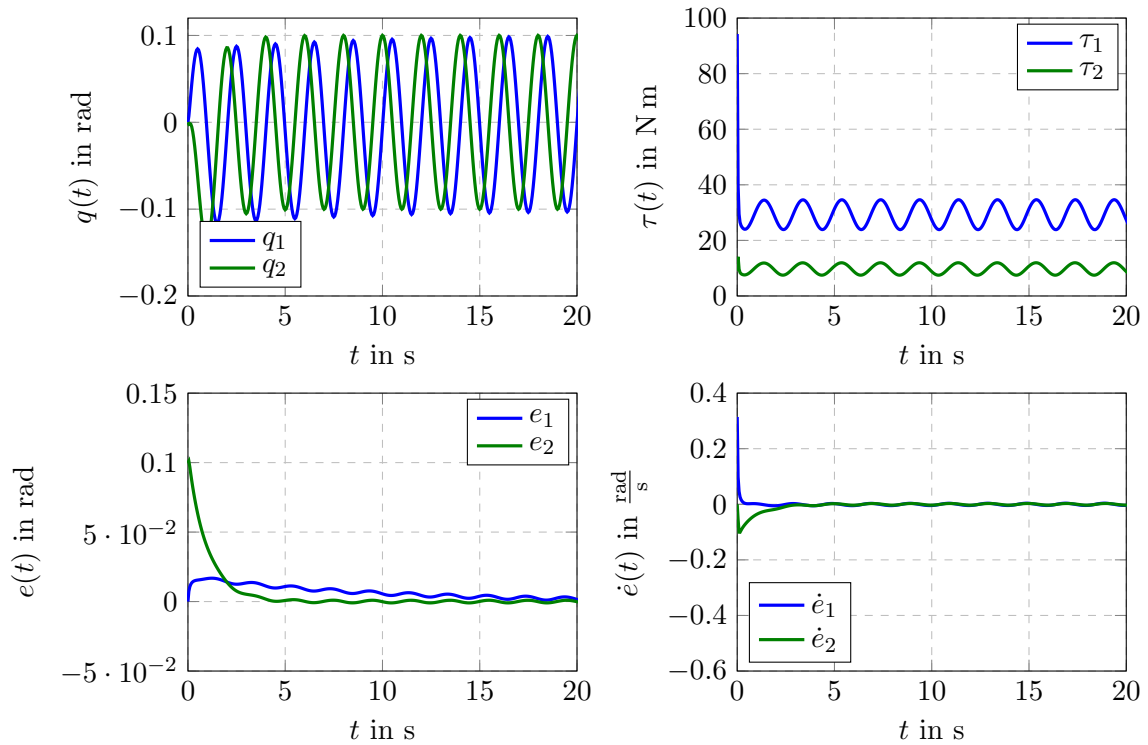


Figure 5.2: Simulation results of CT like controller with disturbance estimator with  $l_1 = l_2 = 20$  and  $\frac{\omega_{estim}}{\omega_n} = 2$

Figure 5.3 shows the results of the simulation with  $l_1 = l_2 = 100$  and a ratio of 2. Compared to the previous simulation, the system settles much faster. The first joint takes approximately 2 s to settle and the second is already settled after 1 s. Also, the oscillation of the error has a lower amplitude than before. On the other hand, the initial torque peak is higher, with a value of approximately 200 N m.

In Figure 5.4, the simulation results with the controller parameters  $l_1 = l_2 = 100$  and  $\frac{\omega_{estim}}{\omega_n} = 10$

can be seen. Through the higher ratio, the frequency of the controller is reduced, which causes a higher settling time as in the previous simulation. The peak error is still much lower than in the first simulation with  $l_i = 20$ . The amplitude of the oscillation of the error is similar to the previous results. Increasing the ratio resulted in a lower torque peak at the beginning, which is now around 100 N m.

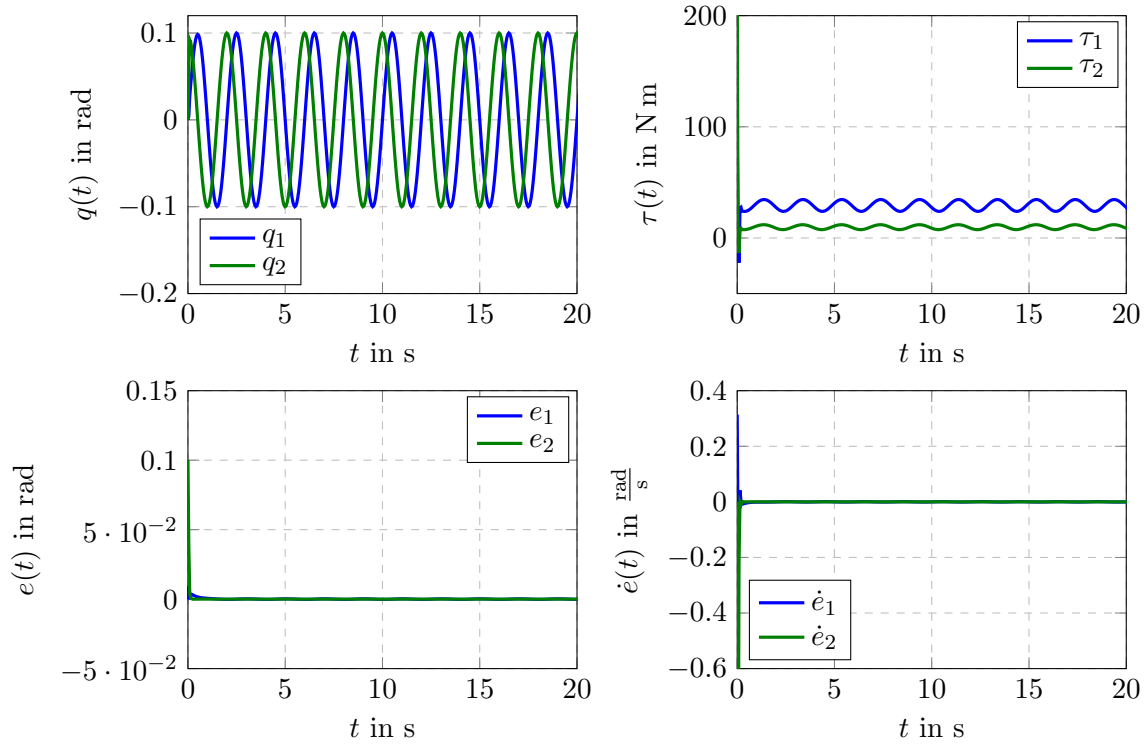


Figure 5.3: Simulation results of CT like controller with disturbance estimator with  $l_1 = l_2 = 100$  and  $\frac{\omega_{estim}}{\omega_n} = 2$

In this chapter, a CT like controller with PI disturbance estimator was shown and its efficiency was analysed. The behaviour of the controller depends mainly on the estimator parameter  $l_i$  and the ratio between the frequencies  $\frac{\omega_{estim}}{\omega_n}$ . Increasing the parameter  $l_i$  results in a faster settling of the error system. Changing the ratio  $\frac{\omega_{estim}}{\omega_n}$  results in a change of the maximum position error and also influences the settling time.

The next chapter describes the design and the simulation of a sliding mode controller for the 2 DOF planar robot.



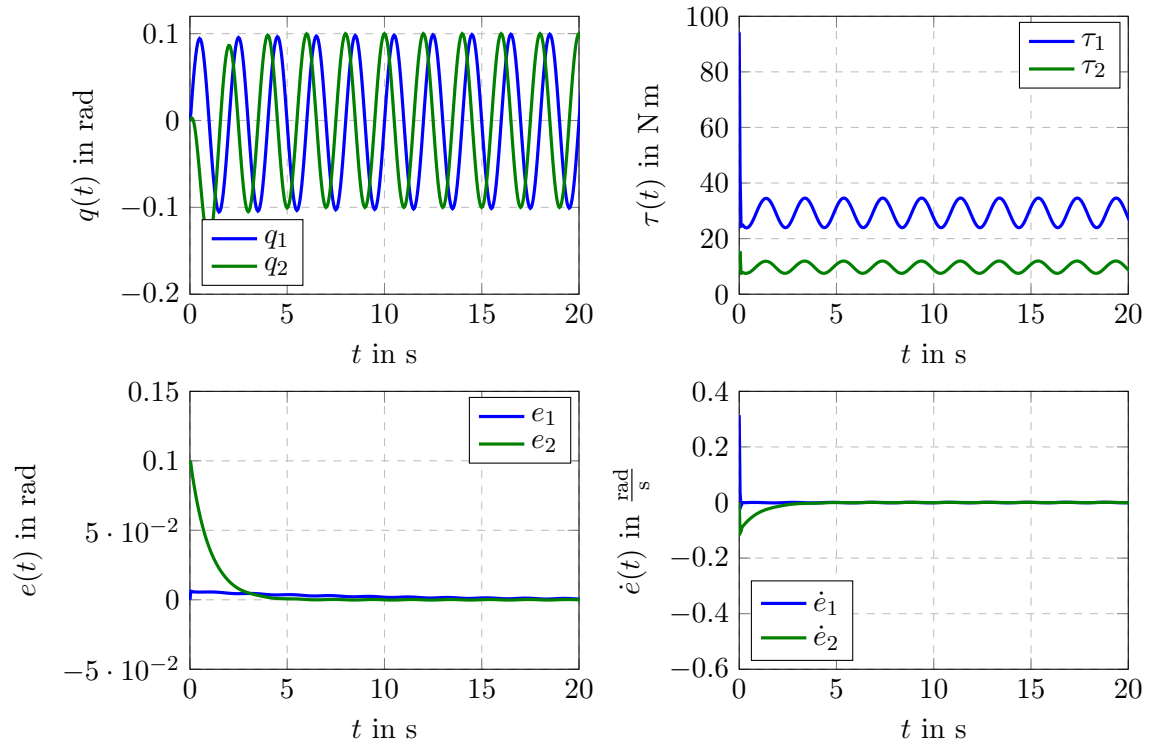


Figure 5.4: Simulation results of CT like controller with disturbance estimator with  $l_1 = l_2 = 100$  and  $\frac{\omega_{estim}}{\omega_n} = 10$

## 6 Sliding mode control

The main idea behind Sliding Mode Control (SMC) is to force a system to a so called sliding manifold and let it slide along this surface to the origin. When on the sliding manifold, the system behaves like a system with an order of one less than the original. To keep the system on the sliding manifold, a rapid switching actuating signal is applied. This is obtained by the *sign* function. As this would need infinite fast switching, which cannot be realised, often a boundary layer is introduced, where the *sign* function is replaced with another function, like the *tanh*. Each joint of the robot is a second order system. This means, that the sliding surface has to be of first order. In order to stabilise the error system of the robot arm, the following sliding manifold is used for the  $i^{th}$  joint:

$$\mathbf{s} = \dot{\mathbf{e}} + \Lambda \mathbf{e}$$

where:

$\Lambda$  : diagonal matrix with time constants of the systems, when in sliding mode

This corresponds to a differential equation of first order. In order, that the system moves towards the origin,  $\lambda > 0$  must be fulfilled. To control the robot, the following SMC law is applied:

$$\tau_c = \hat{\mathbf{M}}\ddot{\mathbf{q}}_s + \hat{\mathbf{V}}_m\dot{\mathbf{q}}_s + \hat{\mathbf{g}} + \mathbf{K}sign(\mathbf{s})$$

where:

- $\hat{\mathbf{M}}$  : estimated inertia matrix
- $\hat{\mathbf{V}}_m$  : estimated centrifugal/coriolis matrix
- $\dot{\mathbf{q}}_s$  :  $\Lambda \mathbf{e} + \dot{\mathbf{q}}_D$
- $\hat{\mathbf{g}}$  : estimated gravitation vector
- $\mathbf{K}$  : diagonal matrix with gains

The parameter matrix  $\mathbf{K}$  is used to suppress the unknown or unmodelled dynamics of the system. This means, the better the dynamics are known or modelled, the lower  $\mathbf{K}$  can be chosen. To test the performance of the SMC law, some simulation were done. For the first few simulations, the ideal sliding mode controller with the *sign* function was used. For the last simulations, the *sign* was replaced with the *tanh* function, to show the effects of the boundary layer. For the simulations, the following reference signals were used:

$$\begin{aligned} q_{D,1} &= \sin(t) \\ q_{D,2} &= \cos(t) \end{aligned}$$

### 6.1 Ideal sliding mode controller

As estimated parameters, a portion of the actual parameters is used. The first three simulations use  $\hat{\mathbf{M}} = 0.25\mathbf{M}$ ,  $\hat{\mathbf{V}}_m = 0.75\mathbf{V}_m$  and  $\hat{\mathbf{g}} = 0.75\mathbf{g}$ . To show the influence of the controller parameters, the first simulation uses  $\lambda_i = 10$  and  $k_i = 20$ . In the second simulation  $\lambda_i = 10$  and  $k_i = 10$  were applied. For the last simulation,  $\lambda_i = 25$  and  $k_i = 10$  was used.

Figure 6.1 shows the simulation results for the parameters  $\lambda_i = 10$  and  $k_i = 20$ . The position error of the first joint converges to zero after approximately 1s and for the second joint after 0.5s. The torques of both joints show the high frequency switching of the SMC law.

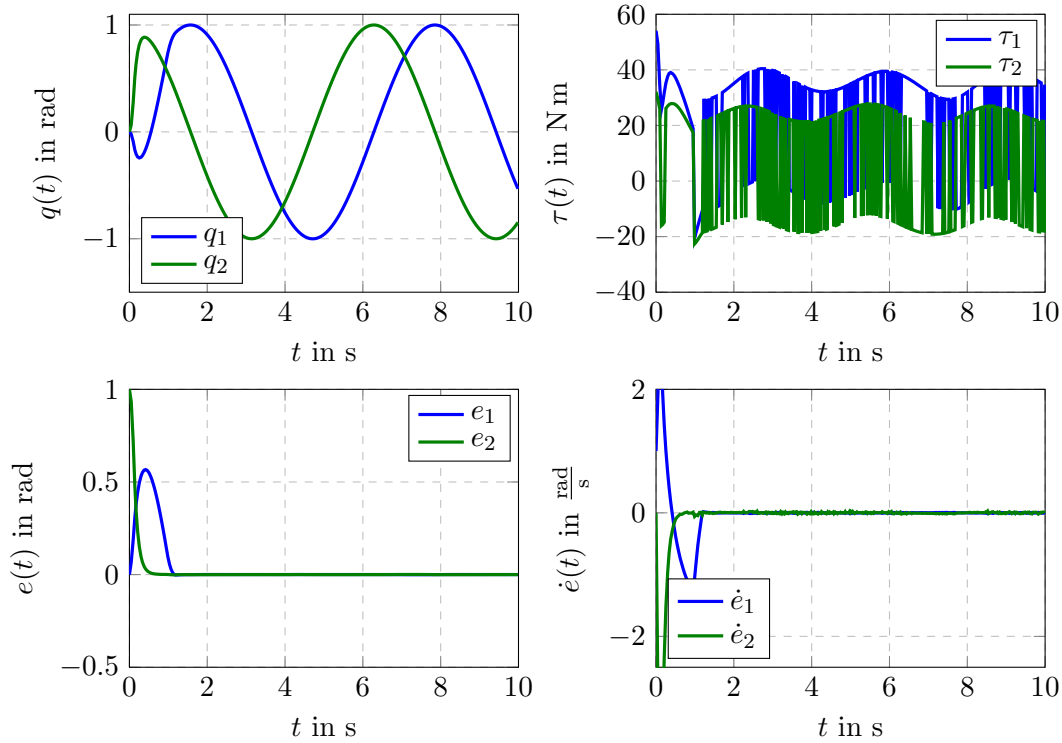


Figure 6.1: Simulation results of sliding mode controller with  $\lambda_i = 10$  and  $k_i = 20$

In Figure 6.2, the simulation results for the parameters  $\lambda_i = 10$  and  $k_i = 10$  are shown. Compared to the previous simulation, the first joint takes approximately twice as long to converge. Also, the maximum position error is higher than in the previous simulation. As the gain for the *sign* function was reduced, the height of the torque jumps is also reduced. Furthermore, the maximum and minimum torque for the two joints is also reduced.

Figure 6.3 shows the results of the simulation with the increased  $\lambda$ . It can be seen that the position error takes even longer to converge towards zero. Also, the torque has a high peak of 130 N m in the settling process.

For the next two simulations, the estimated gravitation is reduced to zero. For the first simulation, the controller parameters of the first simulation were used,  $\lambda_i = 10$  and  $k_i = 20$ , and for the second one the parameters were adapted, to obtain better results.

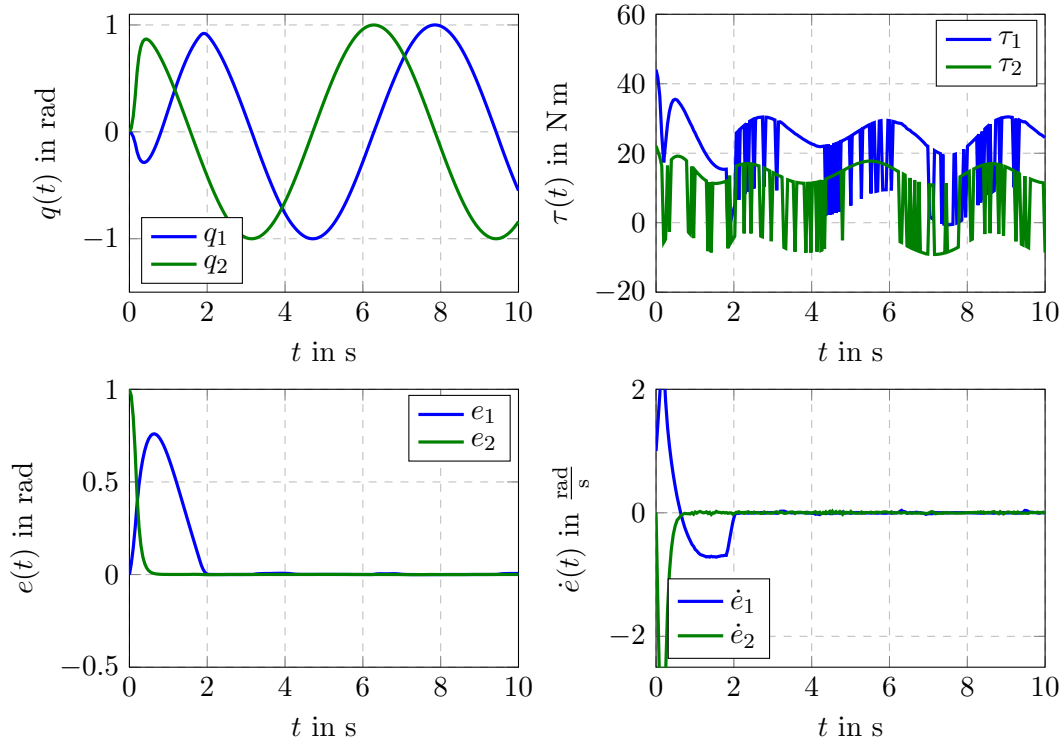


Figure 6.2: Simulation results of sliding mode controller with  $\lambda_i = 10$  and  $k_i = 10$

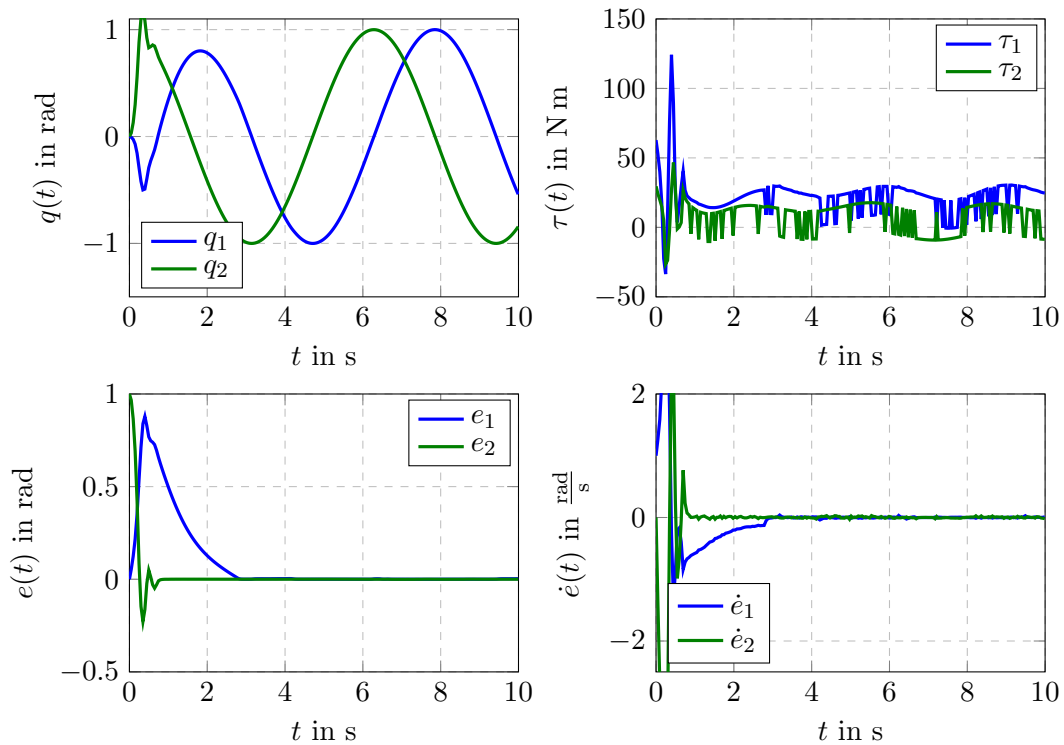


Figure 6.3: Simulation results of sliding mode controller with  $\lambda_i = 25$  and  $k_i = 10$

Figure 6.4 shows the simulation results for the controller with no gravity estimation and the original controller parameters. Because of the neglected gravitation, the position error for the

first joint does not converge anymore. The applied switching signal is not high enough to keep the joint on the sliding mode.

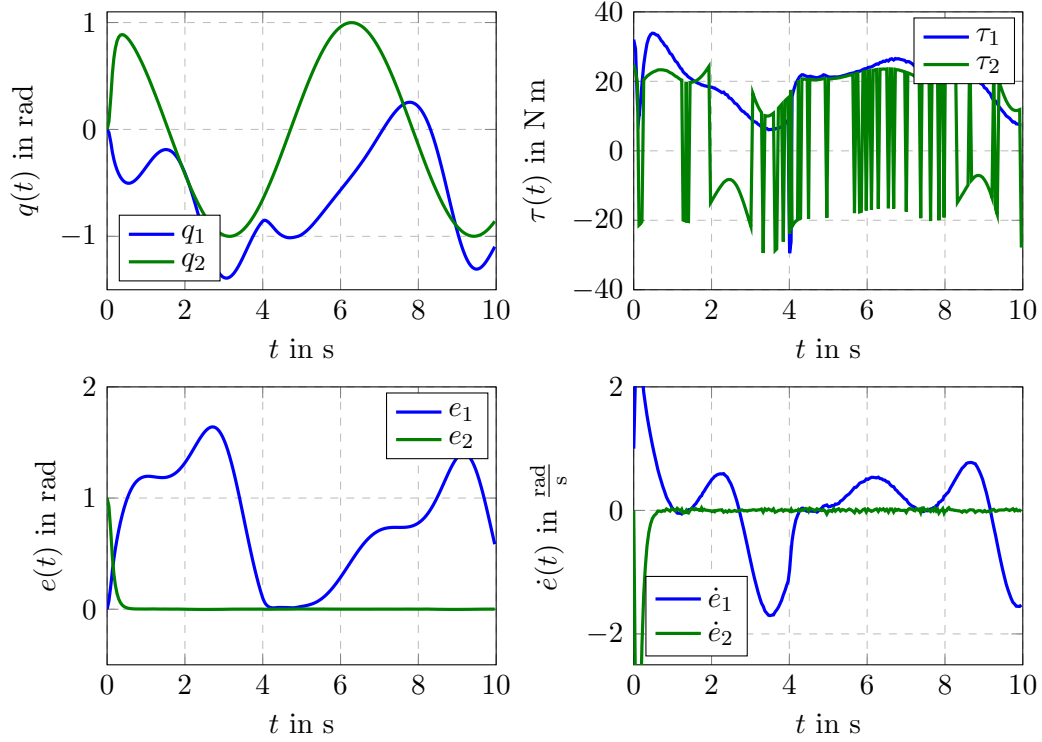


Figure 6.4: Simulation results of sliding mode controller with  $\lambda_i = 10$ ,  $k_i = 20$  and  $\hat{\mathbf{g}} = 0$

In Figure 6.5 the results of the simulation with no gravity estimation and the adapted SMC law parameters can be seen. The gain of the *sign* function was increased to 30. By using these parameters, the error system of the first joint is reduced to an oscillation with small amplitude. In order to settle the first joint, 2s are needed. The change of the performance of the second joint is not clearly visible.

For the next two simulations, a boundary layer is introduced.

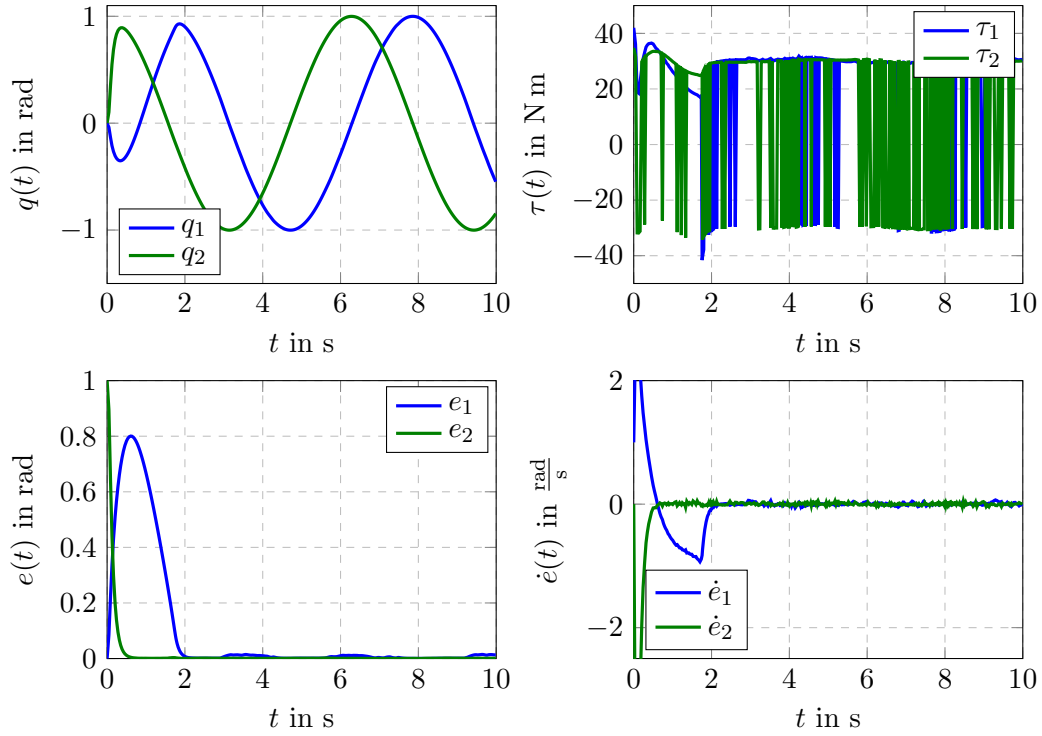


Figure 6.5: Simulation results of sliding mode controller with  $\lambda_i = 10$ ,  $k_i = 30$  and  $\hat{\mathbf{g}} = 0$

## 6.2 Boundary layer

To check the influence of the change from the ideal sliding mode, which uses the *sign* function, to a boundary layer with a *tanh* function another two simulations were realised. For the first simulation, the controller parameters  $\lambda_i = 10$  and  $k_i = 20$  were used along with *tanh(s)*. For the second simulation, the parameters were adapted, to result in a more suitable behaviour.

Figure 6.6 shows the results of the simulation with the boundary layer *tanh(s)* with the initial controller parameters. The high frequency switching of the applied torque has disappeared. On the other hand, the error dynamics has lost its asymptotic stability and shows some oscillating behaviour around the origin.

For the simulation results, which can be seen in Figure 6.7, the parameters  $\lambda_i = 10$  and  $k_i = 40$  along with the boundary layer *tanh(5s)* were used. Using this set of parameters, the amplitude of the oscillation of the error dynamics of the first joint is greatly reduced. The initial peak is approximately 0.4 rad. On the other hand, the applied torque is also increased, when compared to the previous simulation.

In this chapter, a SMC law was used for the control of the 2 DOF planar robot. This control concept uses a high frequency switching to force the system behaviour to converge. The robustness of this concept was analysed, by only using a portion of the system dynamics for the controller design and by even neglecting a part of it. The final simulations showed the influence of the boundary layer.

In the next chapter, an adaptive control law is presented and its behaviour is analysed.

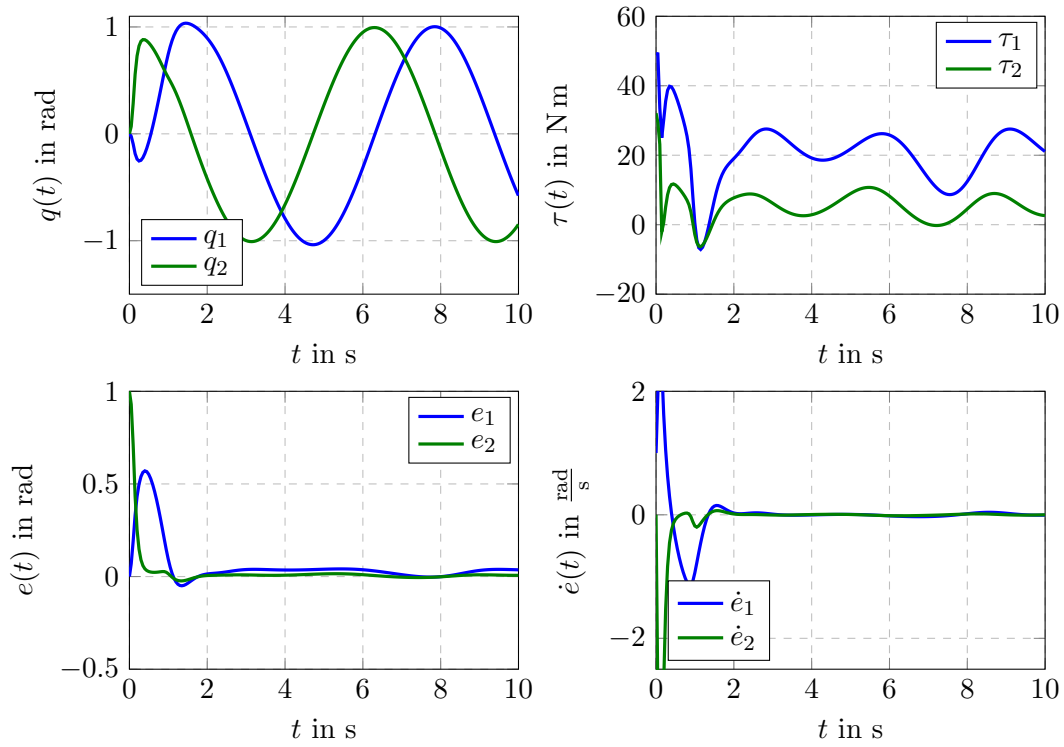


Figure 6.6: Simulation results of sliding mode controller with  $\lambda_i = 10$ ,  $k_i = 20$  and  $\tanh(s)$

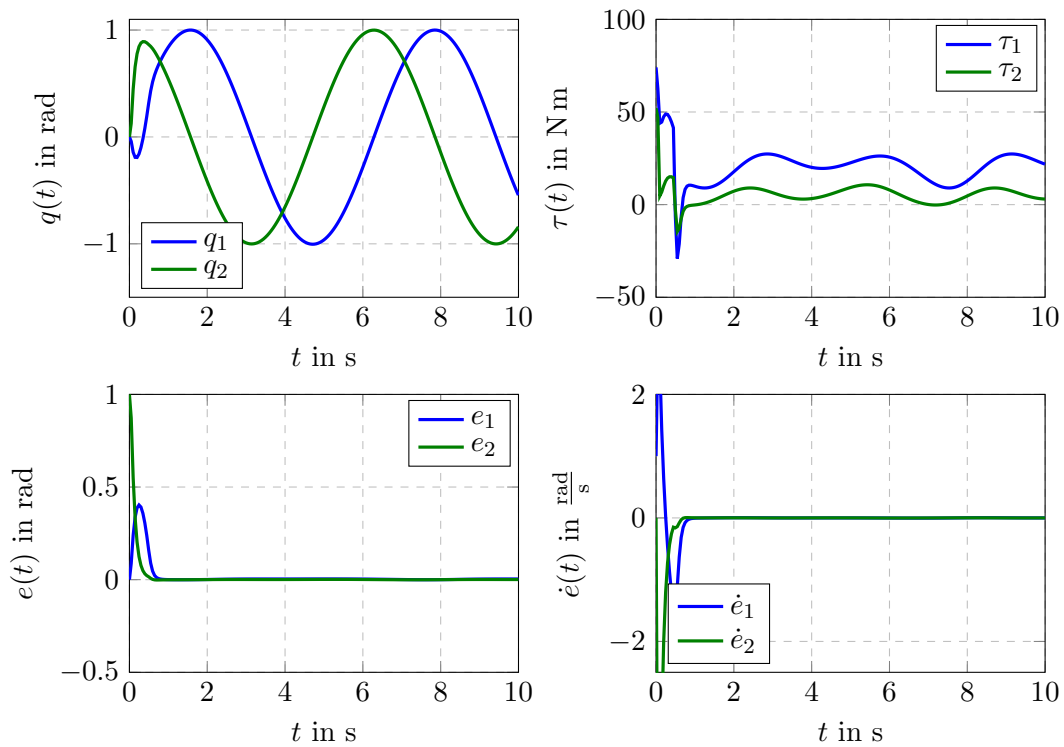


Figure 6.7: Simulation results of sliding mode controller with  $\lambda_i = 10$ ,  $k_i = 40$  and  $\tanh(5s)$

## 7 Adaptive control

An adaptive controller uses an algorithm to change the parameters of the control law during the control process in order to minimize the error of the system. For the control of the planar robot, an inertia related adaptive controller is used. For this approach, the dynamic equation of the robot arm is reordered. All known parts of the dynamic equation are collected into a matrix denoted by  $\mathbf{Y}(\cdot)$  and the parameters, which have to be adapted, are gathered in a vector denoted by  $\varphi$ . In this case, the masses of the system are supposed to be unknown. With the following equation, the matrix  $\mathbf{Y}(\cdot)$  can be calculated:

$$\mathbf{Y}(\cdot)\varphi = \mathbf{M}(\ddot{\mathbf{q}}_D + \Lambda\dot{\mathbf{e}}) + \mathbf{V}_m(\dot{\mathbf{q}}_D + \Lambda\mathbf{e}) + \mathbf{g}$$

As already mentioned, the masses are unknown. Therefore, the parameter vector  $\varphi$  consists of the two masses. All other parts of the equation are put into  $\mathbf{Y}(\cdot)$ . This results in the following elements of the matrix:

$$\begin{aligned} Y_{11} &= (\ddot{q}_{D,1} - \lambda_1 (\dot{q}_1 - \dot{q}_{D,1})) a_1^2 + g \cos(q_1) a_1 \\ Y_{12} &= g (a_2 \cos(q_1 + q_2) + a_1 \cos(q_1)) + (\ddot{q}_{D,1} - \lambda_1 (\dot{q}_1 - \dot{q}_{D,1})) (a_1^2 + 2 \cos(q_2) a_1 a_2 + a_2^2) \\ &\quad + a_2 (\ddot{q}_{D,2} - \lambda_2 (\dot{q}_2 - \dot{q}_{D,2})) (a_2 + a_1 \cos(q_2)) - 2 a_1 a_2 \dot{q}_2 \sin(q_2) (\dot{q}_{D,1} - \lambda_1 (q_1 - q_{D,1})) \\ &\quad - a_1 a_2 \dot{q}_2 \sin(q_2) (\dot{q}_{D,2} - \lambda_2 (q_2 - q_{D,2})) \\ Y_{21} &= 0 \\ Y_{22} &= a_2^2 (\ddot{q}_{D,2} - \lambda_2 (\dot{q}_2 - \dot{q}_{D,2})) + a_2 (\ddot{q}_{D,1} - \lambda_1 (\dot{q}_1 - \dot{q}_{D,1})) (a_2 + a_1 \cos(q_2)) + a_2 g \cos(q_1 + q_2) \\ &\quad + a_1 a_2 \dot{q}_1 \sin(q_2) (\dot{q}_{D,1} - \lambda_1 (q_1 - q_{D,1})) \end{aligned}$$

Furthermore, the tracking error of the system needs to be defined. This is done in a similar way to the sliding surface of the sliding mode controller:

$$\mathbf{r} = \Lambda\mathbf{e} + \dot{\mathbf{e}}$$

$\Lambda$  is again a diagonal matrix which is positive definite. With these components, the control law can be built:

$$\tau_c = \mathbf{Y}(\cdot)\hat{\varphi} + \mathbf{K}_v\mathbf{r}$$

where:

$$\begin{aligned} \hat{\varphi} &: \text{approximated system parameters} \\ \mathbf{K}_v &: \text{diagonal gain matrix, positive definite} \end{aligned}$$

As a last component, the adaptation rule for the unknown parameters is needed:

$$\dot{\hat{\varphi}} = \Gamma\mathbf{Y}^T(\cdot)\mathbf{r}$$

where:

$$\Gamma : \text{diagonal matrix which defines adaptation speed}$$

For the simulation of the adaptive controller, the masses of the planar robot are changed to  $m_1 = 0.8 \text{ kg}$  and  $m_2 = 2.3 \text{ kg}$ . The initial value for the parameter vector is supposed to be  $\hat{m}_1 = \hat{m}_2 = 0 \text{ kg}$ . The following two reference signals are used for the two joints:

$$q_{D,1} = q_{D,2} = \sin(t)$$



As controller parameters,  $\lambda_i = 5$ ,  $\gamma_i = 10$  and  $k_{v,i} = 100$  were used. In Figure 7.1, the simulation results can be seen. The upper left plot shows the adaptation of the parameters. After approximately 2s, the parameters are adapted to the real values. The plot at the upper right side shows the applied control torque. The plots at the bottom show the position and velocity error. It can be seen, that the position error does not exceed 0.05 rad for both joints and converges towards zero after approximately 2s.

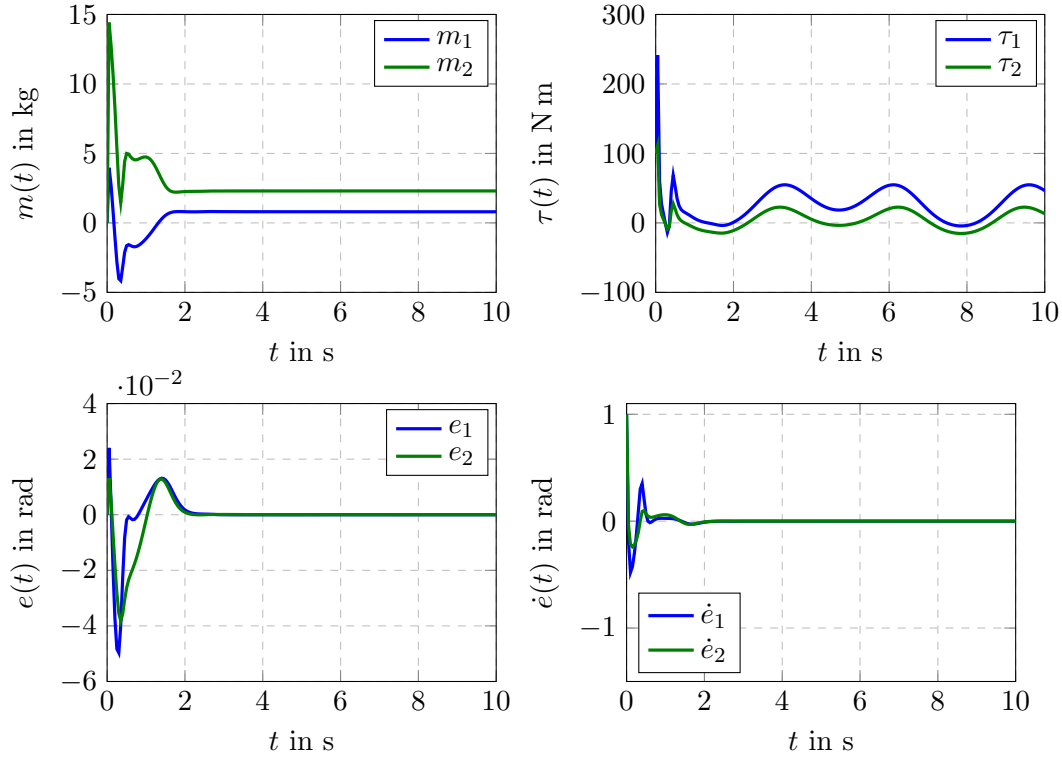


Figure 7.1: Simulation results of adaptive controller with  $\lambda_i = 5$ ,  $\gamma_i = 10$  and  $k_{v,i} = 100$

Another simulation of the adaptive controller with  $k_{v,i} = 10$  was realised. All other parameters are unchanged. The results can be seen in Figure 7.2. Compared to the previous simulation, the adaptation process takes a longer time, as still some ripples are present at 3s. Also the initial torque peak is approximately 50 N m higher than in the previous simulation. The position error of the second joint has a high peak with almost 0.3 rad. After approximately 4s, the parameters are adapted and the system is settled.

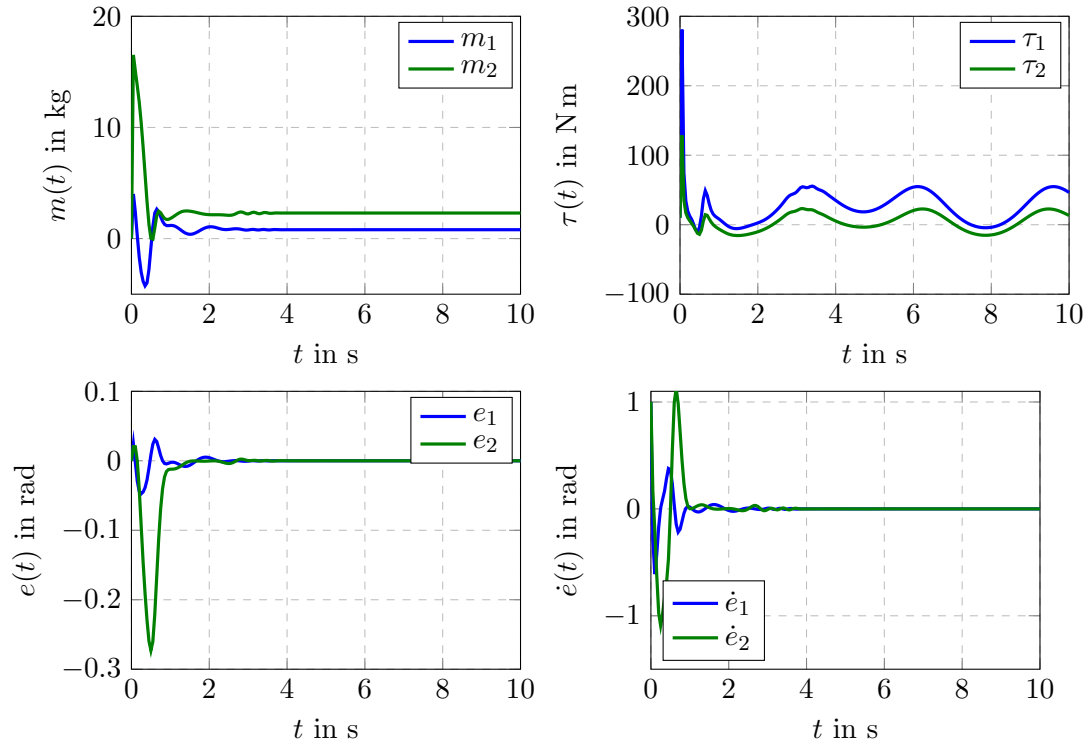


Figure 7.2: Simulation results of adaptive controller with  $\lambda_i = 5$ ,  $\gamma_i = 10$  and  $k_{v,i} = 10$

In this chapter, an adaptive controller was shown. Although the masses were supposed to be unknown at the beginning, the controller managed to adapt to the system in a fast manner.