

ISSN 0280-5316
ISRN LUTFD2/TFRT--5657--SE

Observer-based Impedance Control in Robotics

Dan Henriksson

Department of Automatic Control
Lund Institute of Technology
November 2000

Department of Automatic Control Lund Institute of Technology Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER THESIS		
		<i>Date of issue</i> November 2000		
		<i>Document Number</i> ISRN LUTFD2/TFRT-5657--SE		
<i>Author(s)</i> Dan Henriksson	<i>Supervisor</i> Rolf Johansson and Anders Robertsson			
	<i>Sponsoring organisation</i>			
<i>Title and subtitle</i> Observer-based Impedance Control in Robotics (Observatörsbaserad impedansreglering inom robotik)				
<i>Abstract</i> <p>In situations where robots are performing manipulation tasks, e.g. welding, grinding, and drilling, it is necessary to avoid excessive forces in the contact with the environment. Because of geometric uncertainty and finite positioning accuracy this usually cannot be solved by position control alone. The subject of this thesis is robot force control, with particular interest devoted to impedance control. Two other force control strategies are also treated to some extent. These are compliance control and direct force control.</p> <p>Impedance control is a technique aimed at controlling the dynamic relation between position error and force error in interaction similar to Newton's second law of motion. The velocity is used to modify the damping in the impedance relation. A problem, however, is that the velocity is not available for measurement. For this reason an observer is designed to reconstruct the velocity signal. A good model of the robot joint used is obtained by system identification.</p> <p>Another important issue is stability, and two different approaches are used for the stability analysis. The first one uses theory for piecewise linear control systems, motivated by the fact that the contact force non-linearity can be modeled as piecewise linear. The other approach is based on results on observer-based SPR feedback and stability is proved by the Popov criterion.</p> <p>Experiments are carried out on an ABB industrial robot 2000 to verify the theoretical results.</p>				
<i>Key words</i>				
<i>Classification system and/ or index terms (if any)</i>				
<i>Supplementary bibliographical information</i>				
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>		
<i>Language</i> English	<i>Number of pages</i> 56	<i>Recipient's notes</i>		
<i>Security classification</i>				

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, SE-221 00 Lund, Sweden
Fax +46 46 222 44 22 E-mail ub2@ub2.lu.se

Contents

1. Introduction	3
1.1 Background	3
1.2 Thesis outline	3
2. Robot force control schemes	5
2.1 Compliance control	5
2.2 Impedance control	5
2.3 Direct force control	6
3. The robot system	7
3.1 The ABB industrial robot 2000	7
3.2 The real-time environment	8
3.3 The force sensor	9
4. Identification of joint one	10
4.1 Physical modeling	10
4.2 Experiments	11
4.3 Estimation of ARMAX model	12
4.4 Estimation of continuous state-space model	13
4.5 Validation	14
5. Preliminary experiments	16
5.1 Problem formulation	16
5.2 Experimental setup	16
5.3 The control law	17
5.4 Stability analysis	18
5.5 Case study	22
5.6 Results and discussion	26
6. Observer-based force control	27
6.1 Problem formulation	27
6.2 Theory	27
6.3 Impedance control	28
6.4 Direct force control	34
6.5 Results and discussion	34
7. Implementational aspects	36
7.1 Resolver nonlinearity	36
7.2 Friction	37
8. Conclusions and future work	38
8.1 Conclusions	38
8.2 Future work	39
9. Bibliography	40
A. Stability for piecewise linear control systems	41
B. Matlab code	43
B.1 The ARMAX estimation	43
B.2 The estimation of the continuous state-space model	44
B.3 Computation of piecewise quadratic Lyapunov function using PWLTOOL	45
B.4 Computation of state feedback with observer	46
C. Simulink models	47

1. Introduction

1.1 Background

Today robots are used frequently in industry in different applications. Many of these applications will naturally require the robot to come into contact with a physical environment. This interaction with the environment will set constraints on the geometric paths that can be followed, a situation referred to as *constrained motion*.

To avoid excessive contact forces the robot trajectory must be planned with high accuracy. This is, however, often impossible because of geometric uncertainty and finite positioning accuracy. A way to solve this conflict is to let the robot manipulator be force controlled.

Of the many different schemes for robot force control proposed in the literature [2, 4, 5, 13], three will be considered in this thesis. These are *compliance control*, *impedance control*, and *direct force control*. The emphasis in this thesis will be on impedance control.

An important issue in robot force control is stability. In this thesis an approach is taken to verify stability of one-dimensional impedance control using theory of piecewise linear control systems [6].

A problem with industrial robots is that they often only have sensors to measure position. In impedance control this is of particular concern, as the velocity is used explicitly in the control law. A second approach of this thesis is therefore to study observer-based force control, where a Kalman-filter is used to estimate velocity signals.

1.2 Thesis outline

The thesis is organized as follows:

The second chapter introduces the three force control techniques used in the thesis. These are compliance control, impedance control, and direct force control. Examples are given to illustrate their use.

The third chapter gives an overview of the robot system used in the experiments. The physical robot is presented in the first section, and the second section treats the computer system used to control it. The force sensor is also described.

Chapter four is devoted to identification experiments of joint one of the robot. The identified model is used both in simulations and in the design of the observer dynamics.

In chapter five some preliminary experiments are presented. The goal is to study the properties of impedance control, and to see if stability can be verified by using theory for piecewise linear systems. Simulations as well as experimental results are presented.

Chapter six deals with observer-based force control, with the emphasis on impedance control. A design procedure proposed in [10] is used, which results in an optimal state feedback controller with certain robustness properties. This procedure is extended to the case of reconstructed states in [7]. Stability is proved using the Popov criterion. Simulations and experiments are done to verify the theoretical results.

Some aspects of the implementation are discussed in chapter seven.

The last chapter summarizes the results and some suggestions on extensions and future work are given as well.

2. Robot force control schemes

2.1 Compliance control

Compliance control [2, 4, 13], or stiffness control, can be either passive or active. In passive compliance control the robot end effector is equipped with a mechanical device of springs and dampers to guarantee a certain compliant behavior. Active compliance control, on the other hand, is achieved by tuning the stiffness of the controller with the stiffness of the environment. This is done by using an ordinary PD control action on the position error, and just changing the gain of the controller. The smaller gain that is used, the more compliant the robot will be with regard to the environment. The contact force is treated as a load disturbance. No force measurements are required for compliance control.

2.2 Impedance control

The purpose of impedance control [5, 8] is to obtain a dynamic relation between position and force, rather than controlling either of these variables alone. The simplest form of impedance is the static relation between force and displacement

$$F = K(x_r - x) \quad (2.1)$$

where F is the force at the robot end effector, x is the position, and x_r is a reference position. K is a constant with the interpretation of stiffness.

Another important term in the impedance relation is the relation between force and velocity. A desired relation can be written as

$$F = D \cdot \frac{d}{dt}(x_r - x) \quad (2.2)$$

with the constant D representing damping.

The impedance relation used in this thesis is obtained by combining (2.1) and (2.2), i.e.

$$F = K(x_r - x) + D \cdot \frac{d}{dt}(x_r - x) \quad (2.3)$$

A third term sometimes used in the impedance relation is the acceleration. This term will get the interpretation of inertia.

Equation (2.3) implies the use of an output variable defined as

$$y_i = K \cdot \Delta x + D \cdot \frac{d\Delta x}{dt} - F \quad (2.4)$$

with $\Delta x = (x_r - x)$

Control which keeps $y_i = 0$ will accomplish the desired impedance behavior, i.e. a trade-off between trajectory error and force error. If the motion is unconstrained F will be zero and the robot will move to the reference position.

Impedance control is often used in situations where it is necessary with simultaneous control of motion and force. Consider for example a robot polishing a window. To avoid breaking the glass the robot must be compliant, while on the same time position control must be assured to get the window polished properly. Another area in which impedance control has been applied is cooperation between robots.

2.3 Direct force control

Direct force control [2, 13] is aimed at obtaining a specified force in the contact between the robot and the environment. This can be achieved in a number of ways, but a common technique is the so called outer force-inner position control. As the name implies, an outer force loop is designed around an inner position control loop. The force loop is controlled by a PI-controller, whereas the position loop is controlled by a PID- or PD-controller. The situation is shown in Figure 2.1.

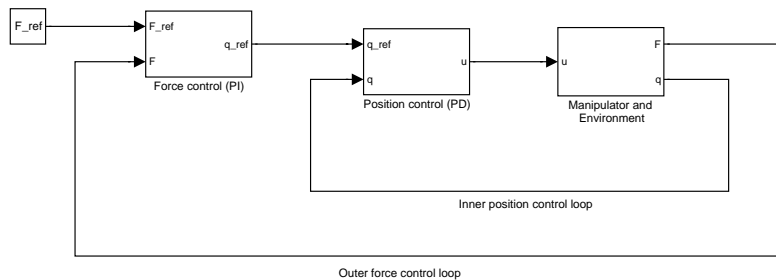


Figure 2.1 Block diagram showing direct force control by outer force-inner position control.

When not in contact, i.e. $F = 0$, the reference to the position loop will be a constant plus a ramp, because of the PI-action, and the robot will be moving. When contact is achieved, the contact force will be controlled to the specified value. The use of integral action on the force error will assure that the stationary force will be equal to the reference force.

Direct force control is used when it is necessary for the robot to exert a desired force on the environment. An example is drilling, where a certain force is commanded, and since the tool moves as material is removed no position control is needed to maintain drilling.

3. The robot system

3.1 The ABB industrial robot 2000

The robot used in the thesis is an ABB Industrial Robot 2000, *irb2000*. The robot has seven *links* which are connected by six *joints*, as shown in Figure 3.1. It is built up by two big arms and a wrist. Joint two (axis B in the figure) is used to move the lower arm back and forth, whereas joint three (A) moves the upper arm up and down. Joint four (D) is used to turn the wrist unit, and joint five (E) bends the wrist unit around its center. The sixth joint (F) is used to turn the robot end effector, which is mounted on the tip of the wrist. The end effector is not shown in the figure. Finally, joint one (C) turns the entire robot around its base.

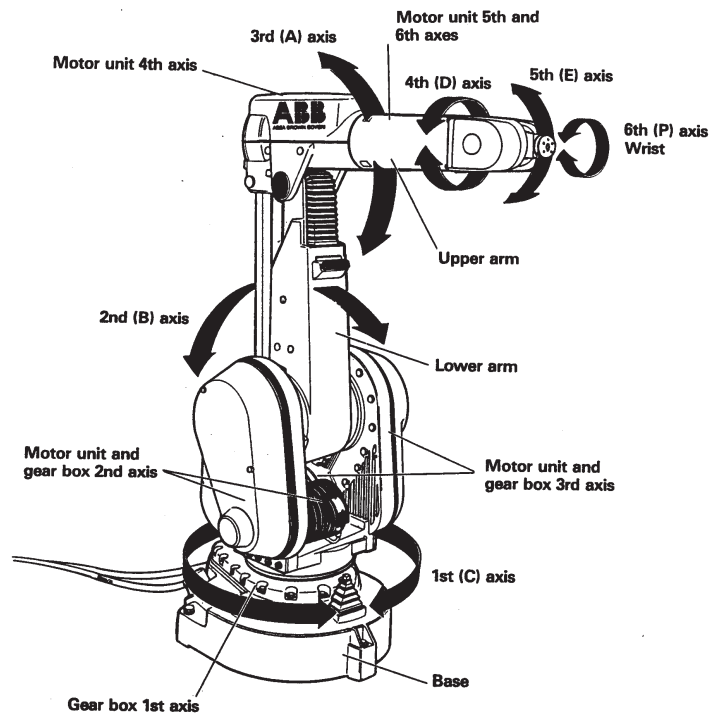


Figure 3.1 The ABB industrial robot 2000. Picture from the Product Manual [1].

The robot system has six different built-in controllers, one for the control of each joint angle. These controllers are cascaded PID-controllers and the block diagram for a single joint is shown in Figure 3.2.

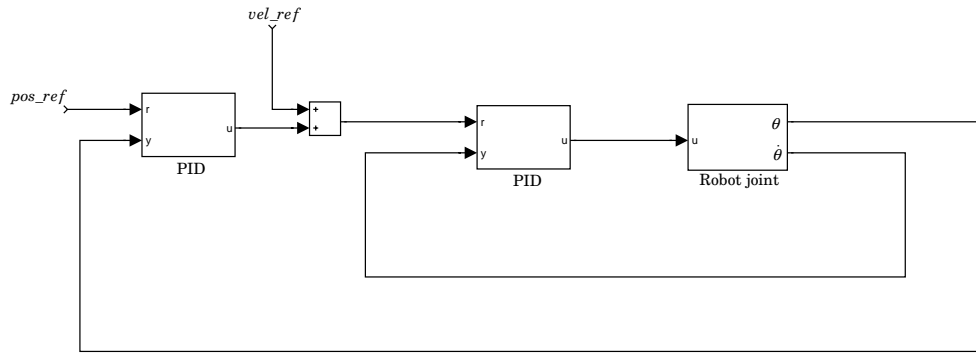


Figure 3.2 The principle of the built-in controllers used to control the different joints. The velocity signal used in the velocity loop is the position signal differentiated and low-pass filtered.

3.2 The real-time environment

All implementation of the control loops is done in Simulink using the Matlab Real-time Workshop. For this purpose the Simulink template shown in Figure 3.3 is available. This Simulink template represents one of the six joints.

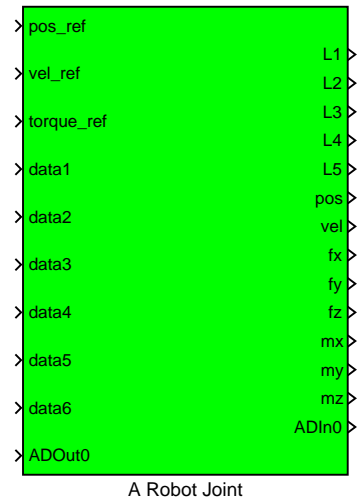


Figure 3.3 The Simulink template used for the implementation.

The Simulink template has ten inputs and fourteen outputs. The inputs *pos_ref* and *vel_ref* are used to give reference values to the built-in controller. To apply a direct torque on the joint, the input *torque_ref* is used. When designing your own controller this input is used for the control signal, and the built-in controller is turned off. The inputs *data1*–*data6* are used for logging signals. Real-time plotting of signals was done using a remote connection to the robot system via the real-time system PÅlsjö.

The outputs *L1*–*L5* are used to change parameters during run-time. This

can be done directly from Matlab. The rest of the outputs are self explanatory. The velocity signal is just the position signal differentiated.

When a Simulink model is down-loaded to the robot system, C-code is generated which is then dynamically linked and run on VME-based computers [11]. The sampling time can be changed but is five milliseconds by default.

A Matlab program called *Exc_handler* is available for simple excitation experiments on the robot. This program can be used to define velocity or position references to the built-in controllers, as well as direct torque inputs. The inputs can be steps, ramps, sinusoids, noise (PRBS), or other arbitrary signals from the Matlab workspace.

A lot of signals can then be recorded during the excitation. These include input torques, position measurements, differentiated position (velocity), and force and torque measurements from the force sensor. The recorded signals can then be exported to the Matlab workspace for plotting.

3.3 The force sensor

The force sensor used to measure the forces is shown in Figure 3.4. It measures forces in the x-, y-, and z-directions as well as the corresponding torques. These signals are then available both in the *Exc_handler* and the



Figure 3.4 The force sensor

Simulink template as described in the previous section.

The sensor is mounted on the robot wrist between the end effector and joint five. To protect the force sensor, the connection to the end effector is done by a pneumatic lock connected to the emergency stop.

4. Identification of joint one

All experiments are done one-dimensionally by moving joint one of the robot. For this reason it is necessary to have a good model of this joint to use both in simulations and for the observer dynamics. The identification of this model is presented below.

4.1 Physical modeling

The robot joint can be modeled by two rotating masses connected by a spring-damper as in Figure 4.1. The angular position and velocity on the motor side are denoted φ_1 and ω_1 , whereas φ_2 and ω_2 denote the corresponding quantities of the robot arm. The process input is the torque, τ , applied by the motor and the measured process output is the angular position on the motor side, φ_1 . The angles are given in motor radians, since this is what the position sensors give. The gear ratio is -71.44 , which means that the true robot angles are about seventy times smaller.

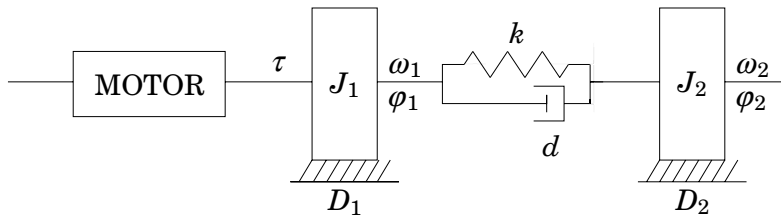


Figure 4.1 Model of a robot joint

The two masses are assumed to have moments of inertia J_1 and J_2 , respectively. They are connected by a spring constant, k , and a damping constant, d . Finally, linear friction is represented by the damping constants D_1 and D_2 . Nonlinear friction is not modeled.

The moment equations become:

$$\begin{aligned} J_1 \dot{\omega}_1 &= -k \cdot (\varphi_1 - \varphi_2) - D_1 \omega_1 - d \cdot (\omega_1 - \omega_2) + \tau \\ J_2 \dot{\omega}_2 &= k \cdot (\varphi_1 - \varphi_2) - D_2 \omega_2 + d \cdot (\omega_1 - \omega_2) \end{aligned} \quad (4.1)$$

By introducing the state variables

$$\begin{aligned} x_1 &= \varphi_1 \\ x_2 &= \omega_1 \\ x_3 &= \varphi_2 \\ x_4 &= \omega_2 \end{aligned} \quad (4.2)$$

and the input $u(t) = \tau(t)$, the system can be written on state-space form

as

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{J_1} & -\frac{D_1+d}{J_1} & \frac{k}{J_1} & \frac{d}{J_1} \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_2} & \frac{d}{J_2} & -\frac{k}{J_2} & -\frac{D_2+d}{J_2} \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \frac{1}{J_1} \\ 0 \\ 0 \end{pmatrix} u(t) \quad (4.3)$$

$$y(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} x(t)$$

The system has the transfer function

$$G(s) = \frac{K(s^2 + 2\zeta_z\omega_zs + \omega_z^2)}{s(s + \frac{1}{T})(s^2 + 2\zeta_p\omega_ps + \omega_p^2)} \quad (4.4)$$

where K , T , ζ_z , ω_z , ζ_p , and ω_p are functions of the unknown process parameters J_1 , J_2 , D_1 , D_2 , k , and d .

4.2 Experiments

First a simple step response experiment was done with the velocity as output to avoid the integrator in the process. By this experiment it was possible to determine the velocity gain, $KT\omega_z^2/\omega_p^2$, and the dominating time constant, of the process. This experiment also revealed that there was no significant time delay in the process. The step response is shown in Figure 4.2. The disturbance on the velocity signal is due to measurement imperfections, which are described in Section 7.1.

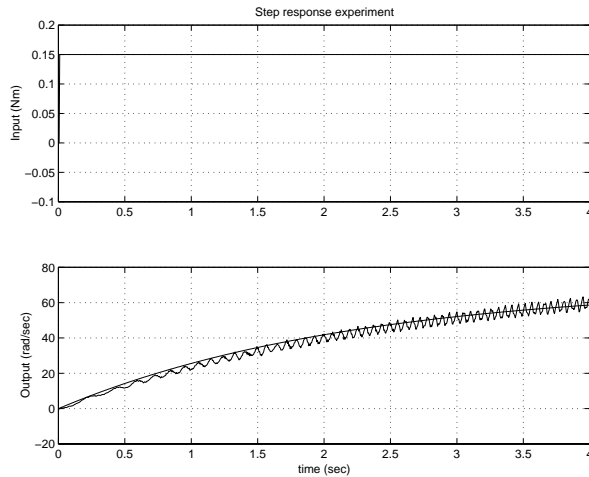


Figure 4.2 Result of step response experiment with the velocity as output. The velocity gain and the dominating time constant can be extracted.

An open-loop excitation was then performed using a PRBS-signal with a dominating frequency of 20 Hz as excitation signal. The input and output

signals are shown in Figure 4.3. These signals were used in the following estimations. The *Exc_handler* described in Section 3.2 was used for the excitation.

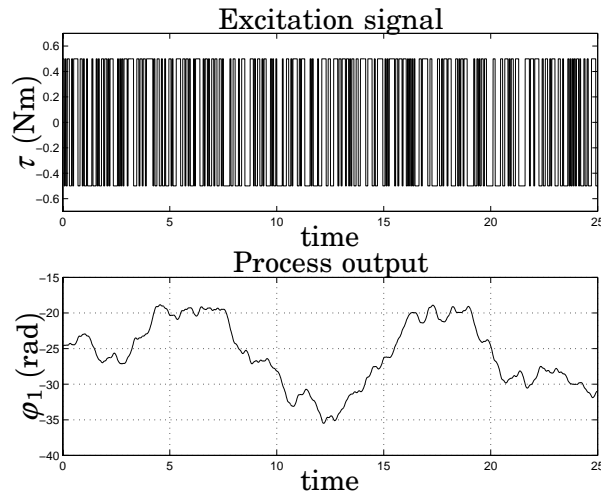


Figure 4.3 Input and output signals of identification experiment

4.3 Estimation of ARMAX model

A fourth order parametric model was fitted in Matlab using the System Identification Toolbox and the function *armax*. The Matlab code is given in Appendix B.1. A Bode plot of the estimated ARMAX model is shown in Figure 4.4. The system appears to have a resonance around 150 rad/sec. Because of the generation of additional zeros by discretization it was im-

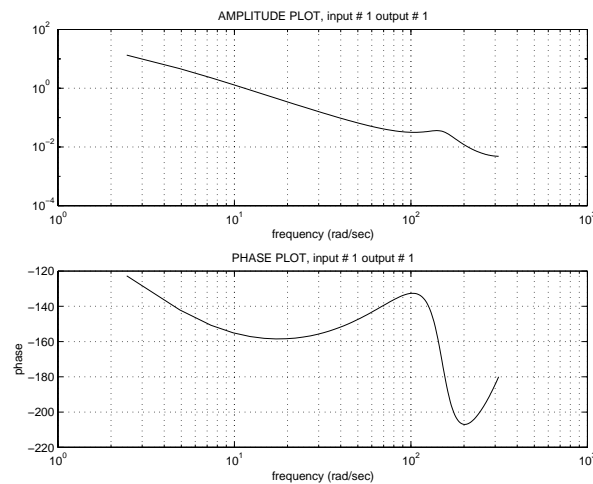


Figure 4.4 Bode plot of the estimated ARMAX model. The resonance frequency around 150 rad/s can be seen in the diagram.

possible to convert the obtained ARMAX model to a continuous state-space model according to (4.3). Another approach had to be used.

4.4 Estimation of continuous state-space model

The Matlab function *pem* was used to estimate the numerical values in the state-space representation (4.3). As described in [9] pp. 36–37, the state-space model was parameterized and converted to the Matlab *theta format*. The parameterization used was

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \theta_1 & \theta_2 + \theta_3 & -\theta_1 & -\theta_2 \\ 0 & 0 & 0 & 1 \\ \theta_7 & \theta_6 & -\theta_7 & \theta_5 - \theta_6 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \theta_4 \\ 0 \\ 0 \end{pmatrix} u(t) \quad (4.5)$$

$$y(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} x(t)$$

with $\theta_1 = -\frac{k}{J_1}$, $\theta_2 = -\frac{d}{J_1}$, $\theta_3 = -\frac{D_1}{J_1}$, $\theta_4 = \frac{1}{J_1}$, $\theta_5 = -\frac{D_2}{J_2}$, $\theta_6 = \frac{d}{J_2}$, and $\theta_7 = \frac{k}{J_2}$.

Since there are seven parameters to estimate we have a seven dimensional optimization problem. This means that it is important to choose good initial values of the parameters in order to avoid local minima. Maple was used to compute the transfer function of (4.5). By comparing this transfer function with the transfer function (4.4) and identifying coefficients we get

$$\begin{aligned} \theta_4 &= K \\ \theta_6 - \theta_5 &= 2\zeta_z \omega_z \\ \theta_7 &= \omega_z^2 \\ -\theta_2 - \theta_3 - \theta_5 + \theta_6 &= 2\zeta_p \omega_p + \frac{1}{T} \\ -\theta_1 + \theta_2 \theta_5 - \theta_3 \theta_6 + \theta_7 + \theta_3 \theta_5 &= \omega_p^2 + \frac{2\zeta_p \omega_p}{T} \\ -\theta_3 \theta_7 + \theta_1 \theta_5 &= \frac{\omega_p^2}{T} \end{aligned} \quad (4.6)$$

Suitable values of K , T , ζ_z , ω_z , ζ_p , and ω_p were chosen based on the step response experiment and the ARMAX estimation. The equations (4.6) were then solved for the parameters. This resulted in good initial values to be used in the estimation. The estimation resulted in the following state-space model

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -11476 & -8 & 11476 & 2 \\ 0 & 0 & 0 & 1 \\ 8445 & 2 & -8445 & -9 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 700 \\ 0 \\ 0 \end{pmatrix} u(t) \quad (4.7)$$

$$y(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} x(t)$$

The poles and zeros of the model are shown in Figure 4.5, and a Bode plot is shown in Figure 4.6. The Matlab code for the estimation is given in Appendix B.2.

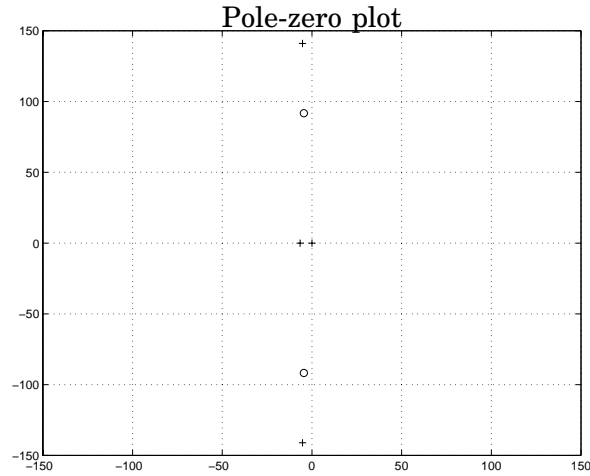


Figure 4.5 Poles and zeros of the estimated state-space model. It can be seen that the complex poles and zeros are quite poorly damped.

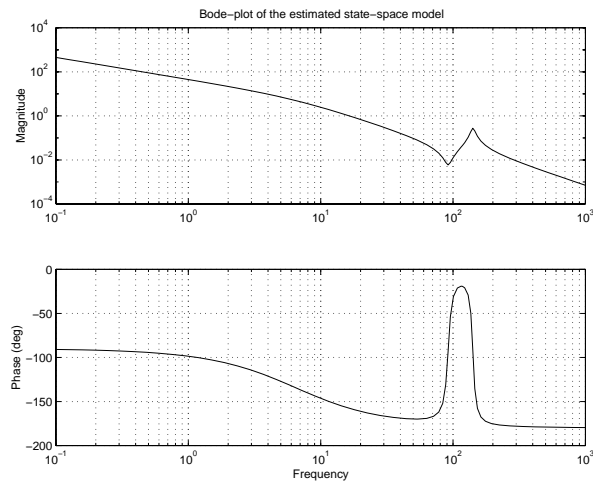


Figure 4.6 Bode plot of the estimated state-space model. The frequencies ω_z , and ω_p can be identified to 92 and 141 rad/sec, respectively.

4.5 Validation

The validation of the state-space model obtained was done by simulation. Figure 4.7 shows a comparison of true process output and a simulation of the estimated state-space model. The correspondence is good, especially for high frequencies.

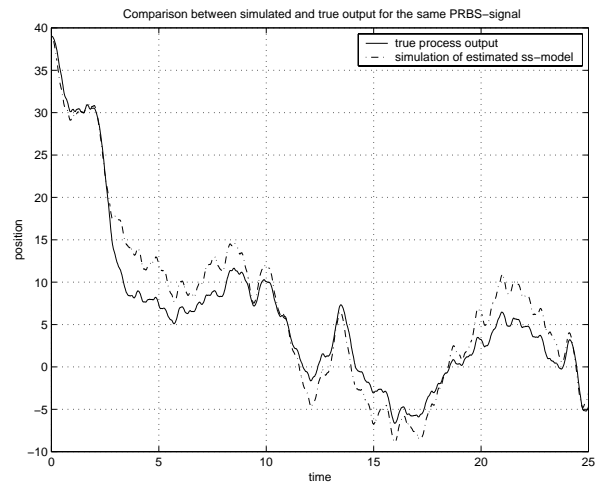


Figure 4.7 Comparison of true process output with a simulation of the estimated state-space model. The same PRBS-signal has been used as input in both cases.

5. Preliminary experiments

5.1 Problem formulation

The aim of these experiments is to study the properties of impedance control, e.g. how the impedance parameters influence the system. A comparison between impedance control and direct force control will also be made. In these experiments the velocity signal will be estimated by differentiating and low-pass filtering of the position signal.

An important issue for impedance control is stability. By modeling the contact force as a linear spring, the contact force nonlinearity will be piecewise linear, see Figure 5.3. Therefore it would be interesting to see if stability could be proved using theory for piecewise linear control systems [6].

5.2 Experimental setup

The experimental setup is shown in Figure 5.1. As mentioned in Section 4 the experiments are done in one dimension only, i.e. along a straight line, which is obtained by moving joint one. Since the angles are small, the end effector path can be approximated to be linear. The physical constraint is represented by a white-board screen, as seen in the figure. A white-board marker is mounted at the end of the end effector. This pen is used to draw a circle on the screen.

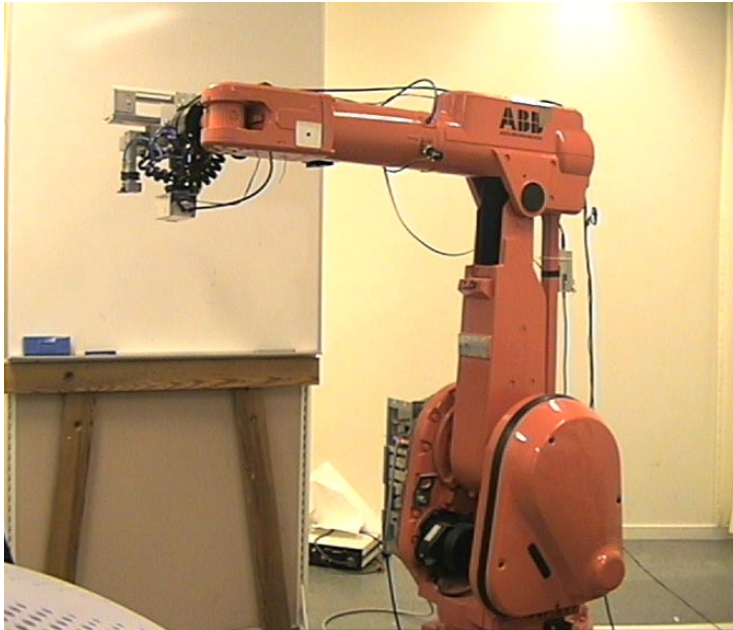


Figure 5.1 The experimental setup. The physical constraint is represented by a screen, and the impedance is controlled perpendicular to the screen.

The situation can be modeled as in Figure 5.2, where x_0 is the initial position, x_c the location of the screen surface and x_r the desired position in the case of unconstrained motion. In stationarity the robot will reach

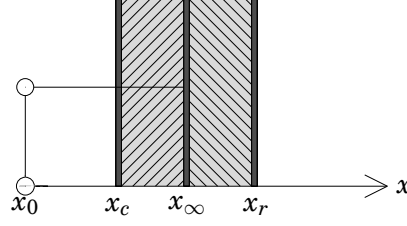


Figure 5.2 A screen representing a physical constraint. x_c is the location of the screen, x_∞ the stationary position, and x_r the desired position in the case of unconstrained motion.

the position x_∞ . If the robot is made very stiff then x_∞ will be close to x_r , whereas a stiff environment will lead to x_∞ being close to x_c .

The contact force is modeled as a regular spring, i.e.

$$F = \begin{cases} 0 & x_3 \leq x_c, \\ k \cdot (x_3 - x_c) & x_3 > x_c. \end{cases} \quad (5.1)$$

where x_3 is the position of the robot arm, as defined by (4.2). The spring constant, k , was determined experimentally to 5 N/rad . This constant reflects the flexibility of the screen as well as the pen.

The nonlinear nature of the contact force is shown in Figure 5.3.

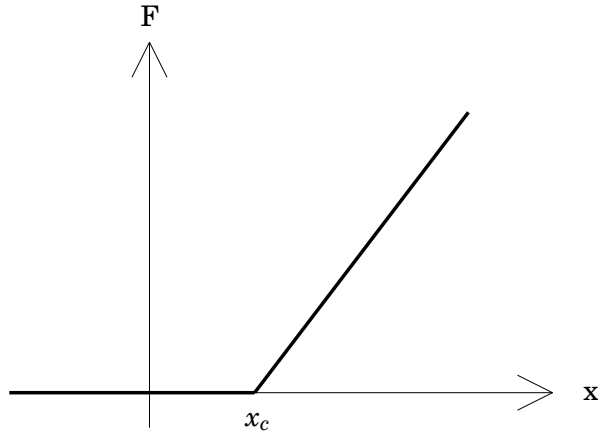


Figure 5.3 The contact force nonlinearity.

5.3 The control law

Consider the impedance output variable

$$y_i = K \cdot \Delta x_1 + D \cdot \frac{d\Delta x_1}{dt} - F \quad (5.2)$$

where x_1 is the position on the motor side as defined by (4.2), $\Delta x = x_r - x_1$ is the position error, F is the contact force normal to the screen, and $K, D > 0$.

In order to keep $y_i = 0$ either a proportional (P) or a proportional-integral (PI) controller can be used. The PI-controller gives better disturbance rejection properties. The control law for the P-controller becomes

$$u = K_c \cdot y_i \quad (5.3)$$

and for the PI-controller we have

$$u = K_c \cdot \left(y_i + \frac{1}{T_i} \cdot \int_0^t y_i(\tau) d\tau \right) \quad (5.4)$$

If the system is started in x_0 the contact force will be zero, see Figure 5.2. This will result in the robot starting to move towards the screen. When the screen is reached the contact force will be non-zero, which will prevent the robot from reaching the desired position x_r . In steady-state ($y_i = 0, \frac{d\Delta x_1}{dt} = 0$) the contact force will be determined by the value of the parameter K according to

$$F_\infty = K \cdot \Delta x_\infty \quad (5.5)$$

where $\Delta x_\infty = x_r - x_\infty$ is the steady-state position error due to the physical constraint.

Using (5.1) and (5.5) combined with the fact that $x_1 = x_3$ in stationarity, we get

$$k \cdot (x_\infty - x_c) = K \cdot (x_r - x_\infty)$$

Solving for the steady-state position, x_∞ , we get

$$x_\infty = \frac{K \cdot x_r + k \cdot x_c}{K + k} \quad (5.6)$$

Thus, (5.5) and (5.6), give the equilibrium point for the impedance control.

We can observe the following from (5.6)

$$\begin{aligned} x_\infty &\approx x_r, & \text{if } K &\gg k \\ x_\infty &\approx x_c, & \text{if } k &\gg K \end{aligned}$$

So if the robot is made stiff compared to the environment, the stationary position will be close to the reference position. On the other hand, if the environment is much stiffer than the robot the stationary position will be close to the screen surface.

5.4 Stability analysis

Stability of the closed loop system under the impedance control laws (5.3) and (5.4) will now be analyzed. The theory for piecewise linear control systems used in this section is summarized in Appendix A.

P-control

Introduce the state vector

$$\tilde{x} = \begin{pmatrix} x_1 - x_\infty \\ x_2 \\ x_3 - x_\infty \\ x_4 \end{pmatrix} \quad (5.7)$$

Proving stability of $x_1 = x_3 = x_\infty$, and $x_2 = x_4 = 0$ is then equivalent to proving stability of the origin of (5.7).

By combining (4.7), (5.1), (5.2), (5.3), and (5.7) the closed loop system can be expressed as a piecewise linear system with the contact force defining the partition of the state space. The equations become

$$\dot{\tilde{x}}(t) = \begin{cases} A_1 \tilde{x}(t) + a_1 & \tilde{x}_3 \leq x_c - x_\infty, \\ A_2 \tilde{x}(t) + a_2 & \tilde{x}_3 > x_c - x_\infty. \end{cases} \quad (5.8)$$

with

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -11476 - 700K_c K & -8 - 700K_c D & 11476 & 2 \\ 0 & 0 & 0 & 1 \\ 8445 & 2 & -8445 & -9 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -11476 - 700K_c K & -8 - 700K_c D & 11476 - 700K_c k & 2 \\ 0 & 0 & 0 & 1 \\ 8445 & 2 & -8445 & -9 \end{pmatrix}$$

$$a_1 = \begin{pmatrix} 0 \\ 700K_c K(x_r - x_\infty) \\ 0 \\ 0 \end{pmatrix}$$

$$a_2 = \begin{pmatrix} 0 \\ 700K_c K(x_r - x_\infty) + 700K_c k(x_c - x_\infty) \\ 0 \\ 0 \end{pmatrix}$$

Using (5.6) we conclude that $a_2 \equiv 0$ for all parameter values, which it should be since the origin is located in the region X_2 .

PI-control

Because of the integrator in the controller it is necessary to introduce a fifth state. Here, the integral of the impedance variable is used as the new state, i.e.

$$x_5 = \int_0^t y_i(\tau) d\tau$$

Then we have

$$\dot{x}_5 = y_i = K(x_r - x_1) + D \frac{d}{dt}(x_r - x_1) - F = K(x_r - x_1) - Dx_2 - F$$

The extended system becomes

$$\dot{\tilde{x}}_e = \begin{pmatrix} \dot{\tilde{x}} \\ \dot{x}_5 \end{pmatrix} = \begin{pmatrix} A & 0 \\ \begin{pmatrix} -K & -D & 0 & 0 \end{pmatrix} & 0 \end{pmatrix} \begin{pmatrix} x \\ x_5 \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u + \begin{pmatrix} 0 \\ Kx_r - F \end{pmatrix}$$

Using the same partition as for the P-controller, the piecewise linear system can be written as

$$\dot{\tilde{x}}_e(t) = \begin{cases} A_1 \tilde{x}_e(t) + a_1 & \tilde{x}_3 \leq x_c - x_\infty, \\ A_2 \tilde{x}_e(t) + a_2 & \tilde{x}_3 > x_c - x_\infty. \end{cases} \quad (5.9)$$

with $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$, and \tilde{x}_4 defined by (5.7) and $\tilde{x}_5 = x_5$. The system matrices become

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -11476 - 700K_c K & -8 - 700K_c D & 11476 & 2 & \frac{700K_c}{T_i} \\ 0 & 0 & 0 & 1 & 0 \\ 8445 & 2 & -8445 & -9 & 0 \\ -K & -D & 0 & 0 & 0 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -11476 - 700K_c K & -8 - 700K_c D & 11476 - 700K_c k & 2 & \frac{700K_c}{T_i} \\ 0 & 0 & 0 & 1 & 0 \\ 8445 & 2 & -8445 & -9 & 0 \\ -K & -D & -k & 0 & 0 \end{pmatrix}$$

$$a_1 = \begin{pmatrix} 0 \\ 700K_c K(x_r - x_\infty) \\ 0 \\ 0 \\ Kx_r - Kx_\infty \end{pmatrix}$$

$$a_2 = \begin{pmatrix} 0 \\ 700K_c K(x_r - x_\infty) + 700K_c k(x_c - x_\infty) \\ 0 \\ 0 \\ K(x_r - x_\infty) + k(x_c - x_\infty) \end{pmatrix}$$

again with $a_2 \equiv 0$ for all parameter values.

Analysis

Having defined the piecewise linear systems, (5.8) and (5.9) stability can be verified by computing Lyapunov functions according to (A.2). The Matlab toolbox PWLTOOL described in [3] is used, and the Matlab code is given in Appendix B.3. Here follows an example of PI-control with the following parameter values; $K_c = 0.02$, $T_i = 2$, $K = 5$, and $D = 1$. The spring constant, k , is 5 N/rad as mentioned earlier. This gives the following system matrices ($a_2 = 0$ since region two contains the origin)

$$\bar{A}_1 = \begin{pmatrix} A_1 & a_1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -11546 & -22 & 11476 & 2 & 7 & 105 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 8445 & 2 & -8445 & -9 & 0 & 0 \\ -5 & -1 & 0 & 0 & 0 & 7.5 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -11546 & -22 & 11406 & 2 & 7 \\ 0 & 0 & 0 & 1 & 0 \\ 8445 & 2 & -8445 & -9 & 0 \\ -5 & -1 & -5 & 0 & 0 \end{pmatrix}$$

The Matlab function *pwl_analysis* verifies exponential stability by computing piecewise quadratic Lyapunov functions. In this example we get

$$V(\tilde{x}_e) = \begin{cases} \begin{pmatrix} \tilde{x}_e \\ 1 \end{pmatrix}^T \bar{P}_1 \begin{pmatrix} \tilde{x}_e \\ 1 \end{pmatrix} = \begin{pmatrix} \tilde{x}_e \\ 1 \end{pmatrix}^T \begin{pmatrix} P_1 & q_1 \\ q_1^T & r_1 \end{pmatrix} \begin{pmatrix} \tilde{x}_e \\ 1 \end{pmatrix}, & \tilde{x}_e \in X_1 \\ \tilde{x}_e^T P_2 \tilde{x}_e, & \tilde{x}_e \in X_2 \end{cases}$$

with

$$\bar{P}_1 = \begin{pmatrix} 24.2613 & 0.8004 & -8.9434 & 1.0915 & -3.0902 & -1.1005 \\ 0.8004 & 0.1304 & 0.1852 & 0.1765 & -0.7585 & 0.2442 \\ -8.9434 & 0.1852 & 26.9665 & 0.2526 & -6.8410 & -5.6143 \\ 1.0915 & 0.1765 & 0.2526 & 0.2426 & -1.0310 & 0.3261 \\ -3.0902 & -0.7585 & -6.8410 & -1.0310 & 10.8290 & -0.0795 \\ -1.1005 & 0.2442 & -5.6143 & 0.3261 & -0.0795 & -8.5988 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 24.2613 & 0.8004 & -8.2098 & 1.0915 & -3.0902 \\ 0.8004 & 0.1304 & 0.0224 & 0.1765 & -0.7585 \\ -8.2098 & 0.0224 & 30.6306 & 0.0352 & -6.7880 \\ 1.0915 & 0.1765 & 0.0352 & 0.2426 & -1.0310 \\ -3.0902 & -0.7585 & -6.7880 & -1.0310 & 10.8290 \end{pmatrix}$$

Stability for other parameter values used in the following section is verified by the same procedure.

5.5 Case study

The impedance control described in the previous sections is here tested in simulations and by experiments on the real robot. First the influence of the impedance parameters is investigated, and then a comparison is made between impedance control and direct force control. A PI-controller with $K_c = 0.02$ and $T_i = 2$ has been used to regulate the impedance variable to zero.

Various impedance parameters

Simulations and experiments for various values of the impedance parameters are shown in Figure 5.4 and Figure 5.5. The simulations were based on the identified state-space model (4.7), and the Simulink model is given in Figure C.1. The Simulink model used for the experiments on the real robot is shown in Figure C.5.

The parameter K will determine the steady-state contact force as given by (5.5) and (5.6). The screen is placed at $x_c = 7$ rad., and the reference position is at $x_r = 10$ rad. The initial position is zero. For the case $K = 5$ (solid curve in the diagrams), the stationary position and force will be

$$x_\infty = \frac{K \cdot x_r + k \cdot x_c}{K + k} = \frac{5 \cdot 10 + 5 \cdot 7}{5 + 5} = 8.5 \text{ rad}$$

$$F_\infty = K \cdot \Delta x_\infty = 5 \cdot (10 - 8.5) = 7.5 \text{ N}$$

These theoretical results are confirmed by the simulation and the experiment. It is also seen that the parameter D represents damping during the transient. The correspondence between simulations and experiments is good. A small force transient is present in the experiments, which can not be seen in the simulations. This is an acceleration force as measured by the force sensor.

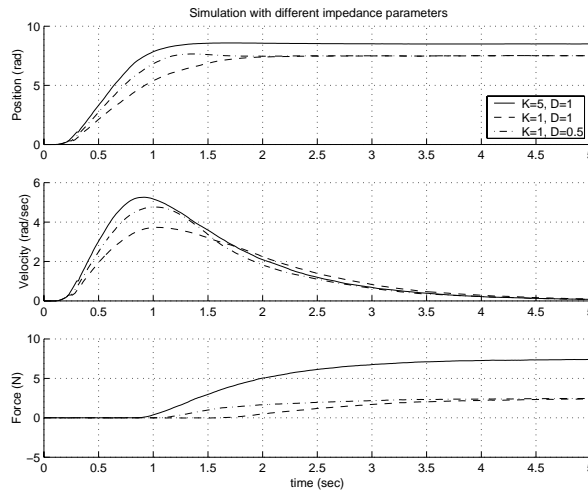


Figure 5.4 Simulation with various impedance parameters. It can be seen that K determines the stationary position and force, whereas D affects the damping during the transient.

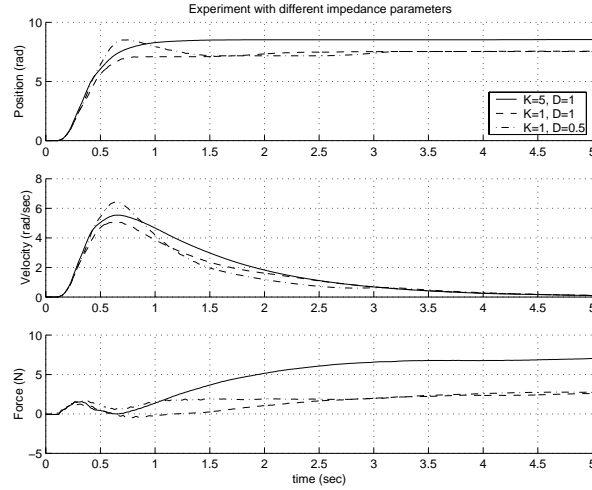


Figure 5.5 Experiments with various impedance parameters. It can be seen that K determines the stationary position and force, whereas D affects the damping during the transient. Good correspondence with simulations. The initial force transient is due to acceleration of the robot.

Comparison of impedance control and direct force control

The main drawback with impedance control is that the force can only be controlled indirectly by changing the values of the impedance parameters. Direct force control on the other hand will keep the contact force on a specified value, as described in Section 2.3.

This difference can be seen when a circle is drawn in the plane of the screen in Figure 5.2. Figure 5.8 shows how the robot moves when drawing the circle. Joints two and three are used to draw the circle, and the reference signals to these joints are shown in Figure 5.7.

If the screen is slightly tilted x_c will change during the drawing, and since the reference position is constant this will cause a varying contact force for the impedance control, as given by (5.6) and (5.5). In the case of direct force control the contact force will be constant, as seen in Figure 5.6. The Simulink models used for simulations and experiments of direct force control are shown in Figure C.2 and Figure C.6, respectively.

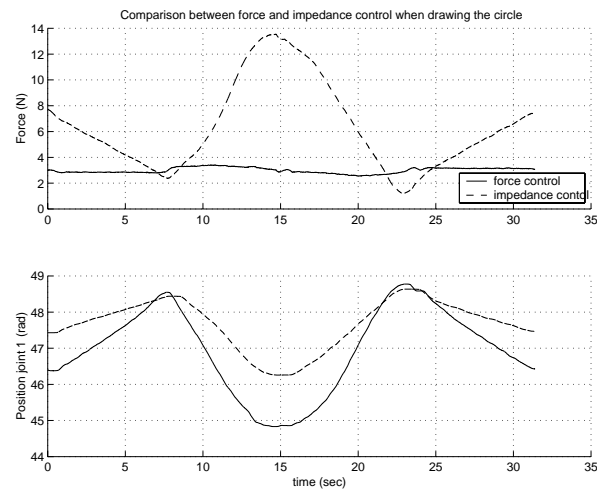


Figure 5.6 Comparison between force and impedance control when drawing a circle in the plane of the screen. The force is about constant for the direct force control, whereas it varies considerably for the impedance control. This is due to the fact that the screen is slightly tilted, i.e. x_c is time-varying.

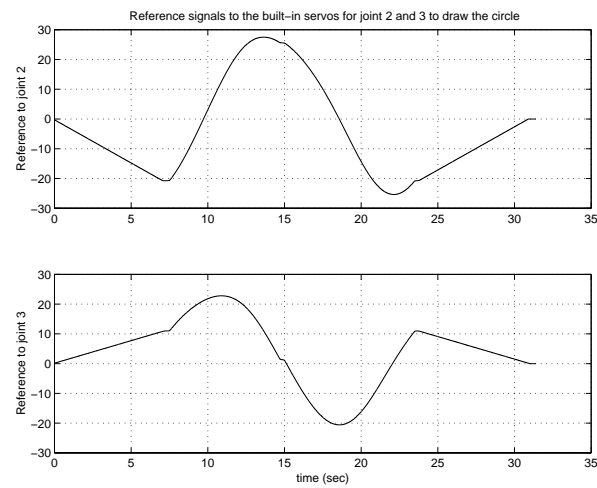


Figure 5.7 Reference signals to built-in servos for joint 2 and 3 to draw the circle.

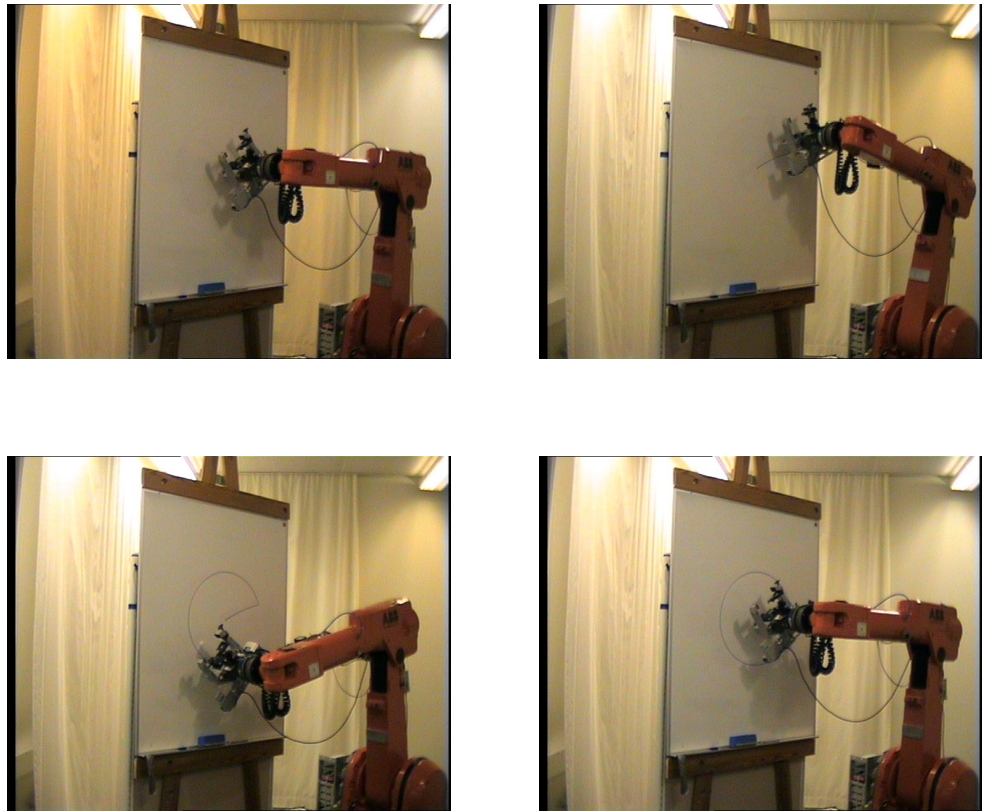


Figure 5.8 The robot drawing the circle.

5.6 Results and discussion

This chapter has been devoted to one-dimensional impedance control. The physical constraint was represented by a screen, and the impedance was controlled perpendicular to this screen. It has been shown theoretically and experimentally how the stationary force will be determined by the parameter K in the impedance law. The impedance parameter D represents damping during the transient. The contact force is also dependent on the distance between the environment and the reference position, which is quite intuitive. Since the exact location of the environment often is unknown, impedance control is not suitable if the contact force is to be controlled to a specified value. In that case direct force control is to prefer.

It has been shown how stability of the closed loop system can be proved using theory for piecewise linear systems. The analysis, however, assumes that the true velocity is used in the impedance relation. Since the velocity is not measurable, it was estimated by differentiating and low-pass filtering of the position signal.

6. Observer-based force control

6.1 Problem formulation

Industrial robots are often equipped with very accurate sensors to measure position, whereas they lack sensors for the measuring of velocity for considerations of cost, volume, and weight.

In force control applications this is a problem, especially in impedance control where one uses velocity signals explicitly in the control law. One way to solve this problem is to differentiate the position measurement and use this as a velocity estimate. This is, however, a very noise sensitive approach. A better way to deal with the problem is to use observers to estimate the velocity.

In the following an approach to observer-based force control will be made, where a full-order Kalman-filter based on the identified state-space model (4.3) is used to estimate position and velocity signals. Theoretical results presented in [7] will be applied and tested in simulations and experiments. The main attention will be given to impedance control.

6.2 Theory

The control laws will be designed by observer-based state feedback, with the state feedback gain chosen to obtain certain robustness properties [10]. The design procedure implies a choice of the state feedback gain according to

$$L = R^{-1}B^TP_c \quad (6.1)$$

where $P_c = P_c^T > 0$ is the solution to the algebraic Riccati equation

$$P_c(A - BL) + (A - BL)^TP_c = -Q_c - P_cBR^{-1}B^TP_c \quad (6.2)$$

$Q_c > 0$, and $R > 0$ are design matrices, which represent penalties on the states and the control signal, respectively.

If all states not are measurable, approximate control is made with $u = -L\hat{x}$ where \hat{x} is given by the full-order observer

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K(y(t) - C\hat{x}(t)) \quad (6.3)$$

The extended system can then be written as

$$\begin{pmatrix} \dot{x} \\ \dot{\hat{x}} \\ u \end{pmatrix} = \begin{pmatrix} A & 0 & B \\ KC & A - KC & B \\ 0 & -L & 0 \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \\ u \end{pmatrix} \quad (6.4)$$

By introducing the observer state error $\tilde{x} = \hat{x} - x$ and the control error $Lx + u$ with u given as $u = -L\hat{x} + r$, with r being a reference value, the closed loop system will get the dynamics matrix

$$\mathcal{A}_0 = \begin{pmatrix} A - BL & -BL & B \\ 0 & A - KC & 0 \\ -L & -L & 0 \end{pmatrix}$$

See [7] for details.

If we can find a Lyapunov function

$$\begin{aligned} V(X) &= X^T \mathcal{P}_0 X, \\ \frac{dV(X)}{dt} &= -X^T Q X \end{aligned} \quad (6.5)$$

with \mathcal{P}_0 and Q satisfying the YKP equation

$$-Q = \mathcal{P}_0 \mathcal{A}_0 + \mathcal{A}_0^T \mathcal{P}_0 \quad (6.6)$$

with

$$Q = \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix} \quad (6.7)$$

and $Q = Q^T > 0$,

then separation and stability will hold for the extended system and the results of [10] will hold, even for the approximate control $u = -L\hat{x}$.

A design algorithm to solve for $\mathcal{P}_0 = \mathcal{P}_0^T > 0$ is given and proved in [7].

6.3 Impedance control

The control law

We have seen that impedance control is aimed at regulating a certain output variable to zero. This variable is a linear combination of position and velocity minus the measured force. Since the states in the state-space model of the robot joint are position and velocity, observer-based impedance control will be achieved by the state feedback control law

$$u = -L\hat{x} - c \cdot F \quad (6.8)$$

with \hat{x} given by (6.3) with the system matrices (A,B,C) from the identified state-space model (4.7). The constant c is a scaling of the force signal.

The observer gain matrix, K , is chosen using pole placement. The poles of $A - KC$ are placed on a circle segment in the left half plane according to an ordinary Butterworth pattern. Since the position is measurable and the robot started at rest, it is no problem to have correct initial values for

the observer.

Different degrees of stiffness and damping can be applied by choosing different weights in the weighting matrix Q_c . Q_c is chosen as

$$Q_c = \begin{pmatrix} K & 0 & 0 & 0 \\ 0 & D & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{pmatrix}$$

The stiffness is changed by acting on the penalty K . A higher value gives a less compliant system. The penalty D will affect the damping. The weighting matrix R is a weight on the control signal, and has been chosen high to give a smooth control signal.

Stability analysis

Stability concerning the contact force non-linearity will be analyzed using the Popov criterion. The block diagram of the system under the impedance control (6.8) is shown in Figure 6.1. Stability is analyzed by plotting the Popov curve for the transfer function from A to B . Figure 6.2 shows an example for the case $K = 40$, $D = 1$, and $R = 1000$, which is used later in the experiments.

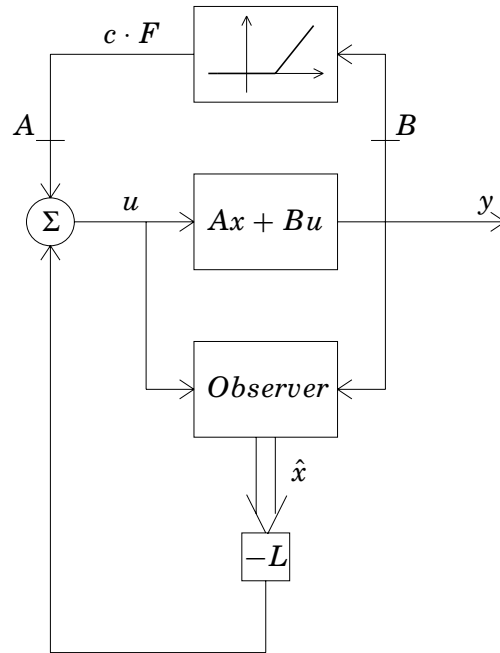


Figure 6.1 Block diagram of the impedance control. The Popov plot of the transfer function from A to B has been used to analyze stability for the contact force non-linearity.

The scaling of the force signal, c , is chosen as $c = (l_1 + l_3)/K$, to give the same steady-state interpretation of K as in (5.6). In this case we get $c = 0.005$. The slope of the non-linearity is then $c \cdot k$, with k being the stiffness of the environment. This was estimated earlier to $k = 5 \text{ N/rad}$. The maximum slope allowed is given by k_{max} in Figure 6.2, and we conclude that we have a large stability margin. This is true also for other choices of the state feedback used in the following section.

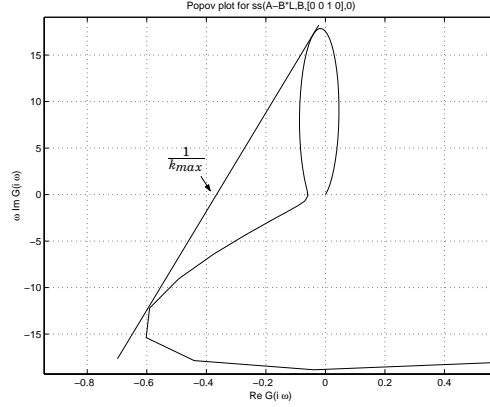


Figure 6.2 Popov plot for the transfer function from A to B in Figure 6.1. The maximum slope of the non-linearity is given by k_{max} . For the quite compliant environment used in the experiments the stability margin is large.

Experiments

Figure 6.3 shows an experiment similar to those in Section 5.5, where the robot comes into contact with a screen. The velocity signal from the observer is improved a lot as compared to the differentiated and low-pass filtered signal in Figure 5.5.

The stationary force is determined by the weight K in the design matrix Q_c , and the scaling c of the force signal. By choosing $c = (l_1 + l_3)/K$ we get the same interpretation of K as in the previous chapter, and (5.6) can be used to calculate the stationary position. The parameters in the experiment was $K = 15$, $k = 5$, $x_c = -3$, and $x_r = 0$, which gives

$$x_\infty = \frac{K \cdot x_r + k \cdot x_c}{K + k} = \frac{15 \cdot 0 + 5 \cdot (-3)}{15 + 5} = -0.75 \text{ rad}$$

$$F_\infty = K \cdot \Delta x_\infty = 15 \cdot (0 - (-0.75)) = 11.25 \text{ N}$$

The reason why the actual values are different, see Figure 6.3, is that no integral action is used when regulating the impedance.

The Matlab function `compute_fb` given in Appendix B.4 has been used to compute the state feedback as well as the Lyapunov function (6.5). The weighting matrices P_0 and Q are verified to be positive definite.

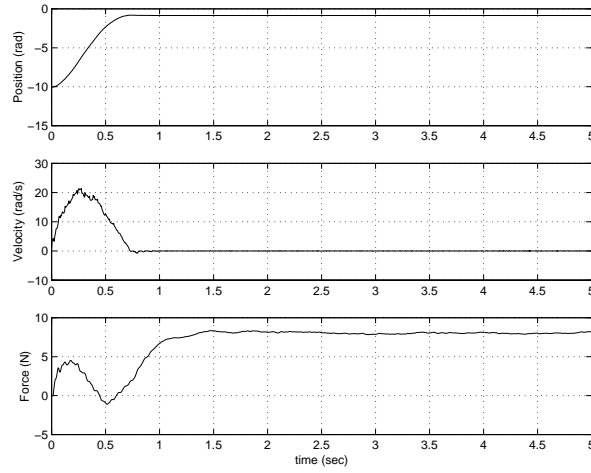


Figure 6.3 Experiment of observer-based impedance control, where the robot comes into contact with a screen. The location of the screen is $x_c = -3$ and the reference position is $x_r = 0$.

Figure 6.4 shows an experiment where the robot is affected by a constant force of 50 N. When the force is released the robot should return to the original position, which can be seen in the simulation in Figure 6.5. The reason why it does not fully return to the original position is due to friction in the robot joint, see Section 7.2. How much the robot should comply for a constant force can be altered by changing the penalty K in the design matrix Q_c . This can be seen in Figure 6.6 where K is decreased, leading to a larger position deviation.

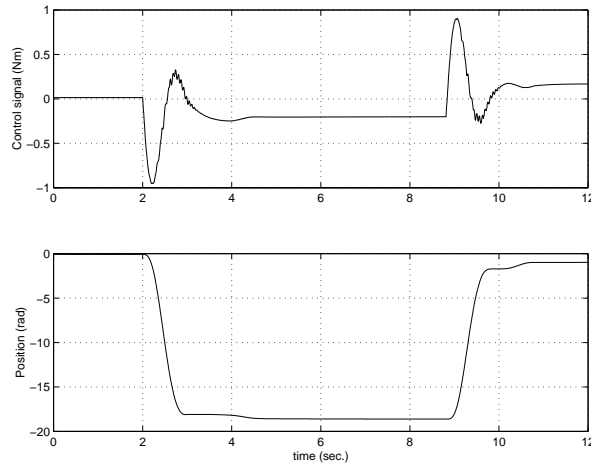


Figure 6.4 Experiment of observer-based impedance control, where the robot is affected by a constant force of 50 N. The penalty K is set to 40.

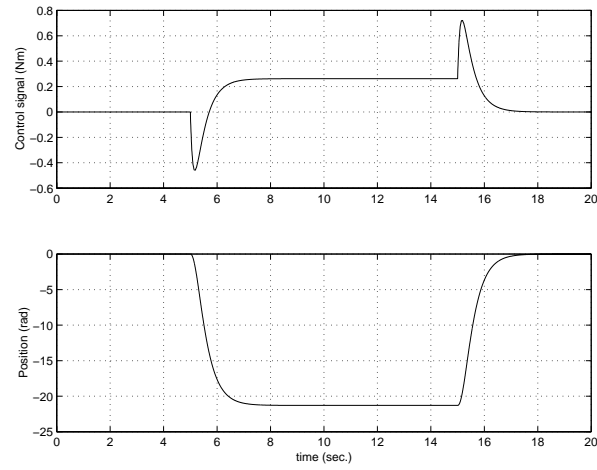


Figure 6.5 Simulation corresponding to the experiment in Figure 6.4

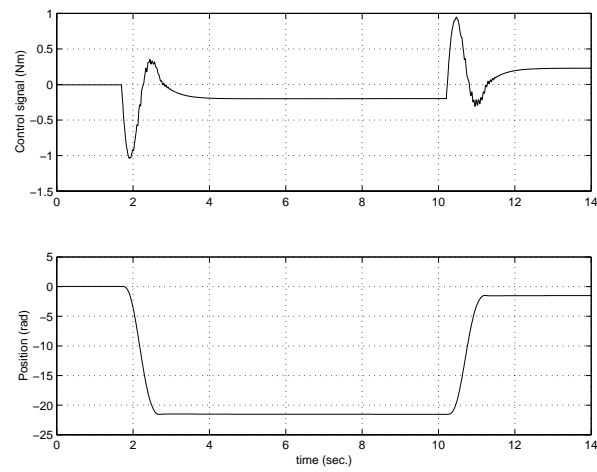


Figure 6.6 Experiment of observer-based impedance control, where the robot is affected by a constant force of 50 N. The position deviation is larger than in Figure 6.4 since the penalty K is chosen lower. Here we have chosen $K=30$.

An alternative to impedance control when the force can not be measured is compliance control. As described in Section 2.1 this is pure position control, where the contact force is treated as a load disturbance. The control law becomes

$$u = -L\hat{x} \quad (6.9)$$

Figures 6.7, and 6.8 show experiment and simulation where again the robot is affected by a constant force of 50 N. The position deviation is quite small, although K is chosen considerably smaller than for the impedance control. The friction phenomena is clearly visible here. The main drawback with compliance control is that if a very compliant behavior is desired, the gain of the controller must be low. This will in turn make the position following poor. As impedance control uses force measurements to balance out the contact force, it allows a larger gain in the position control when not in contact.

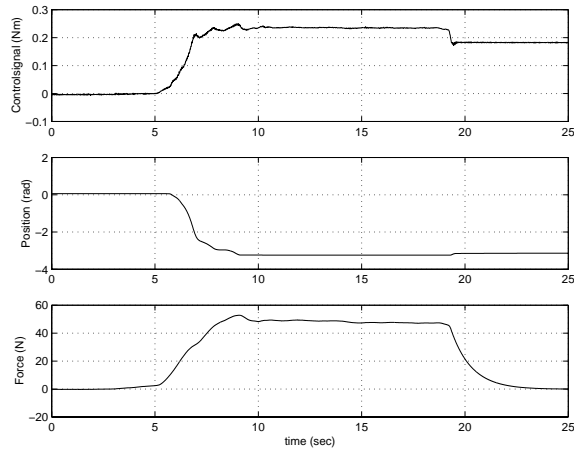


Figure 6.7 Experiment of observer-based compliance control, where the robot is affected by a constant force of 50 N. The penalty K is set to five. The friction phenomena is clearly visible, since the robot does not move when the force is released.

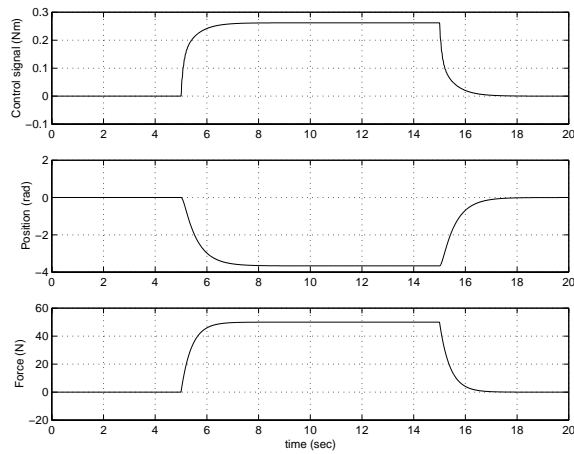


Figure 6.8 Simulation corresponding to the experiment in Figure 6.7

6.4 Direct force control

Observer-based direct force control can be achieved by using the following control law for the inner position loop

$$u = -L\hat{x} + l_r r \quad (6.10)$$

with l_r chosen to make the stationary gain equal to one.

Figure 6.9 shows results from an experiment with direct force control. The robot was first given a force reference of 3 N. Then step changes were made in the force reference while the robot was in contact. The upper plot shows the reference to the inner position loop, and the lower plot shows the force signal. The middle plot shows the actual position. The step responses are quite fast and well damped. The Simulink model is given in Figure C.8.

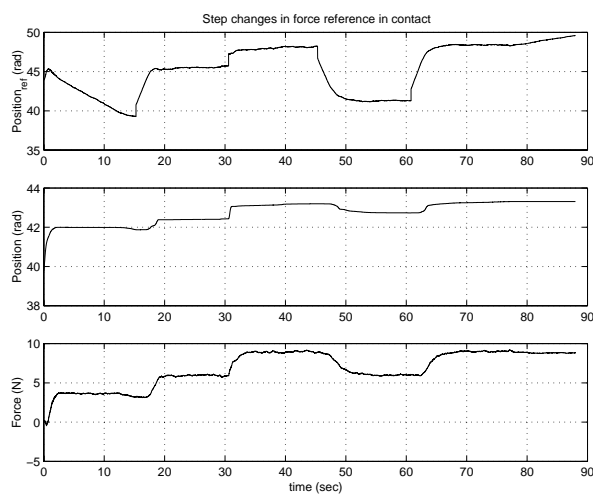


Figure 6.9 Force and position reference signals during force control with step changes in the force reference

6.5 Results and discussion

In this chapter an approach to observer-based robot force control has been made. The use of observers to get estimates of velocity signals is motivated by the fact that most industrial robots only have sensors to measure position. This is of particular concern in impedance control, where velocity signals are used explicitly in the control law.

A design procedure presented in [7] based on results in [10] was used. The suggested state feedback controller was obtained by solving an algebraic Riccati equation, where the design matrices represented penalties on the states and the control signal. It was shown how to choose the design matrices to achieve different stiffness and damping for the impedance control. Stability of the impedance control with respect to the contact force non-linearity was proved by the Popov criterion.

The experimental results correspond nicely with simulations, except that the robot does not return completely to the reference position when the force is released. This is due to friction in the robot joint and lack of integral part in the controller. The velocity estimates provided by the observer are much improved as compared to differentiation and low-pass filtering of the position signal.

Direct force control was also applied, where the inner position control loop was implemented as an observer-based state feedback. Step changes were made in the force reference when in contact. The responses were quite fast and well damped.

A limitation is that the experiments have been performed in one dimension only. The extension to several dimensions, would include the full dynamics of the robot as well as gravity compensation. The nonlinear coupling between the robot links would require the extension to nonlinear observer-based control presented in [12].

7. Implementational aspects

7.1 Resolver nonlinearity

The position is measured by electrical resolvers, whose function is described in detail in [14]. Imperfections in these measuring devices give rise to a small disturbance in the position signal. The measured signal, q_m , can be written as

$$q_m = q + \sum_k a_k \cdot \sin(kq + b_k) \quad (7.1)$$

where q is the true position.

Since $\{a_k\}$ are very small, the disturbance can be neglected in the position measurement. The problem arises when the position measurements are differentiated to obtain velocity estimates. The amplitude of the disturbance will then be scaled with the velocity. Figure 7.1 shows this phenomena.

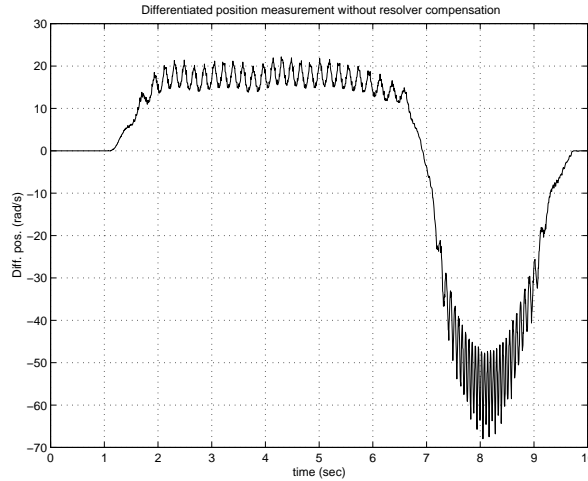


Figure 7.1 Differentiated position measurements without compensation for the resolver nonlinearity. The disturbance increases with increased velocity.

Using Fourier analysis, it is possible to detect which frequencies that are contained in the position signal, i.e. which values of k that should be used in (7.1). This has been done in [14] and if only the fundamental mode ($k=2$) is considered, Equation (7.1) can be approximated as

$$q_m = q + a_2 \cdot \sin(2q + b_2) \approx q + a_2 \cdot \sin(2q_m + b_2) \quad (7.2)$$

The constants a_2 and b_2 were estimated, and Figure 7.2 shows the differentiated position when the fundamental mode is eliminated. The noise is reduced considerably.

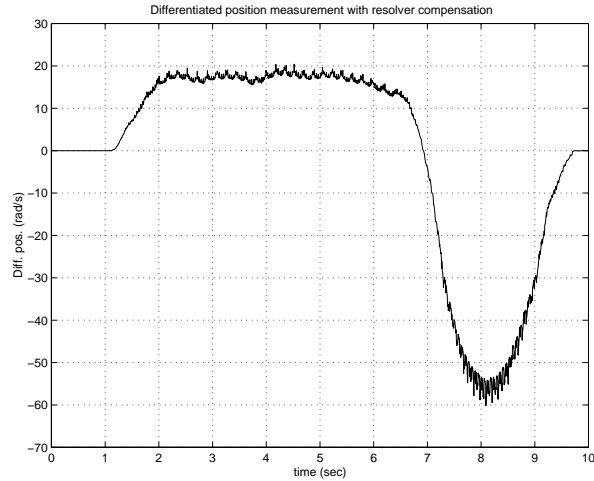


Figure 7.2 Differentiated position measurements with compensation for the fundamental mode of the resolver nonlinearity.

7.2 Friction

Friction is always a problem when dealing with mechanical systems. This was seen in the experiments in Section 6.3, where the robot got stuck for input torques below a certain value. This value is different depending on whether the robot is standing still or is moving, as seen in Figure 7.3.

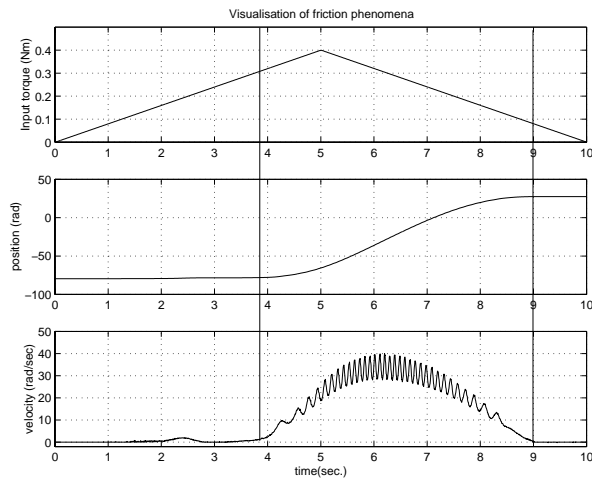


Figure 7.3 The friction problem. The robot does not start to move until the input torque exceeds 0.3 Nm. The hysteresis of the friction nonlinearity is also visible, since the robot gets stuck at a lower level (0.1 Nm) when moving in the other direction.

8. Conclusions and future work

8.1 Conclusions

The subject of this thesis has been robot force control, and particular interest has been devoted to a technique called impedance control. This technique is aimed at obtaining a dynamic relation between position and force, rather than controlling either of these variables alone. Impedance control is often used to deal with geometric uncertainty, i.e. when the exact location of the environment is unknown. Two other robot force control strategies were also treated to some extent. These were compliance control and direct force control.

The simple one-dimensional impedance control considered in this thesis is aimed at regulating the following output variable to zero

$$y_i = K \cdot \Delta x + D \cdot \frac{d\Delta x}{dt} - F$$

where Δx is the position error referred to some reference position, F is the contact force and K and D are positive constants.

The output variable above implies two difficulties. Firstly, it is nonlinear because of the nonlinear nature of the contact force. Secondly, the velocity is not available to measurement. Both these problems were dealt with in the thesis.

By modeling the contact force as a linear spring, it was shown that stability of the closed loop system could be verified using theory for piecewise linear control systems [6]. The analysis was done both when a proportional and a proportional-integral controller was used to regulate the impedance variable. Experiments were done, where the robot came into contact with a screen and the impedance was regulated perpendicular to this screen. In these experiments the velocity signal was estimated by differentiation and low-pass filtering of the position signal. It was shown theoretically and experimentally how the constant K in the impedance relation determined the stationary contact force. The constant D represented damping during the transient.

The second approach taken concerned the fact that the velocity was not available to measurement. For this reason a full-order Kalman-filter was used to estimate the velocity. A good model of the robot joint was needed for the observer dynamics, and was obtained by system identification. A design procedure proposed in [7] based on results in [10] was used. This procedure resulted in a state-feedback controller with certain robustness properties as well as a Lyapunov function used to prove the separation property. The

state-feedback controller was obtained by solving a Riccati equation, and it was shown how to choose the weighting matrices to achieve different stiffness and damping of the impedance control. Stability was proved by the Popov criterion. The theoretical results were verified by experiments and simulations. It was shown that the velocity signal obtained from the observer was considerably improved compared to differentiation and low-pass filtering of the position signal.

8.2 Future work

Since all the experiments in this thesis are done in one dimension only, an obvious extension would be to apply the results to force control in two and three dimensions. This would include use of the Jacobian to translate forces from the task space to the operational space, as well as full dynamics for the robot and gravity compensation. The simple linear model of the robot would not be enough and the coupling between the links would require nonlinear observer-based control [12].

Another possible extension would be to apply the results on other force control schemes such as parallel force/position control or hybrid force/position control.

9. Bibliography

- [1] *Product manual IRB 2000*.
- [2] S. Chiaverini, B. Siciliano, and L. Villani. A survey of robot interaction control schemes with experimental comparison. *IEEE/ASME Transactions on mechatronics*, 4:273–285, 1999.
- [3] S. Hedlund and M. Johansson. A toolbox for computational analysis of piecewise linear systems. In *Proceedings of European Control Conference, Karlsruhe*, 1999.
- [4] A. Hemami and G. Zeng. An overview of robot force control. *Robotica*, 15:473–482, 1997.
- [5] N. Hogan. Impedance control: An approach to manipulation, Parts I–III. *Journal of Dynamic Systems, Measurement, and Control ASME*, 107:1–24, 1985.
- [6] M. Johansson. *Piecewise linear control systems*. PhD thesis, Dept. Automatic Control, Lund Inst. Tech., 1999.
- [7] R. Johansson and A. Robertsson. Observer-based SPR feedback control system design. NOLCOS 2001 (submitted), 2000.
- [8] R. Johansson and M.W. Spong. Quadratic optimization of impedance control. In *Proceedings of IEEE Int. Conf. Robotics and Automation*, pages 616–621, 1994.
- [9] L. Ljung. *System identification toolbox; User’s guide*. The MathWorks Inc., 1991.
- [10] P. Molander and J.C. Willems. Synthesis of state feedback control laws with a specified gain and phase margin. *IEEE Transactions on automatic control*, AC-25:928–931, 1980.
- [11] K. Nilsson. *Industrial robot programming*. PhD thesis, Dept. Automatic Control, Lund Inst. Tech., 1996.
- [12] A. Robertsson. *On observer-based control of nonlinear systems*. PhD thesis, Dept. Automatic Control, Lund Inst. Tech., 1999.
- [13] B. Siciliano and L. Villani. *Robot force control*. Kluwer academic publishers, 1999.
- [14] E. Uddeholt. Identifiering och kompensering av motor- och resolver-rippel. Master’s thesis, KTH, 1998.

A. Stability for piecewise linear control systems

This is based on the stability results for piecewise linear systems presented in [6].

A piecewise linear system can be written on the form

$$\dot{x}(t) = A_i x(t) + a_i + B_i u(t) \quad x(t) \in X_i. \quad (\text{A.1})$$

where $\{X_i\}_{i \in I} \subseteq R^n$ is a partition of the state space into closed (possibly unbounded) polyhedral cells with pairwise disjoint interior. The index set of the cells is denoted I .

For convenience the following notation is used

$$\bar{A}_i = \begin{pmatrix} A_i & a_i \\ 0 & 0 \end{pmatrix} \quad \bar{B}_i = \begin{pmatrix} B_i \\ 0 \end{pmatrix} \quad \bar{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}$$

Let $I_0 \subseteq I$ denote the set of indices for the cells that contain the origin, and $I_1 \subseteq I$ denote the set of indices for cells that do not contain the origin. Using the notation above and assuming that $a_i = 0$ for $i \in I_0$ the system (A.1) can be written as

$$\begin{aligned} \dot{x}(t) &= A_i x(t) + B_i u(t), & i \in I_0 \\ \dot{\bar{x}}(t) &= \bar{A}_j \bar{x}(t) + \bar{B}_j u(t), & j \in I_1 \end{aligned}$$

In order to prove stability of (A.1) we want a piecewise quadratic Lyapunov function on the following form

$$V(x) = \begin{cases} x^T P_i x & x \in X_i, i \in I_0, \\ \begin{pmatrix} x \\ 1 \end{pmatrix}^T \bar{P}_i \begin{pmatrix} x \\ 1 \end{pmatrix} = x^T P_i x + 2q_i^T x + r_i & x \in X_i, i \in I_1 \end{cases} \quad (\text{A.2})$$

The following theorem (Theorem 4.1 in [6]) is central

THEOREM A.1 - Piecewise quadratic stability

Consider symmetric matrices T , U_i , and W_i such that U_i and W_i have nonnegative entries, while

$$\begin{aligned} P_i &= F_i^T T F_i, & i \in I_0 \\ \bar{P}_i &= \bar{F}_i^T T \bar{F}_i, & i \in I_1 \end{aligned}$$

satisfy

$$\begin{cases} 0 > A_i^T P_i + P_i A_i + E_i^T U_i E_i \\ 0 < P_i - E_i^T W_i E_i \end{cases} \quad i \in I_0$$

$$\begin{cases} 0 > \bar{A}_i^T \bar{P}_i + \bar{P}_i \bar{A}_i + \bar{E}_i^T U_i \bar{E}_i \\ 0 < \bar{P}_i - \bar{E}_i^T W_i \bar{E}_i \end{cases} \quad i \in I_1$$

Then every trajectory $x(t) \in \cup_{i \in I} X_i$ satisfying (A.1) with $u \equiv 0$ for all $t \geq 0$ tends to zero exponentially. □

The matrices \bar{E}_i , and \bar{F}_i are called cell boundings and continuity matrices respectively and are defined as (Definitions 4.2 and 4.1 in [6])

DEFINITION A.1 - Cell bounding

A matrix $\bar{E}_i = [E_i \ e_i]$ is called a cell bounding if it satisfies

$$\bar{E}_i \bar{x}(t) \succeq 0 \quad \text{if } x(t) \in X_i$$

where the vector inequality $z \succeq 0$ means that every element of the vector z is nonnegative. □

DEFINITION A.2 - Continuity matrix

A matrix $\bar{F}_i = [F_i \ f_i]$ is a continuity matrix for cell X_i if

$$\bar{F}_i \bar{x}(t) = \bar{F}_j \bar{x}(t) \quad \text{for } x(t) \in X_i \cap X_j$$

□

B. Matlab code

B.1 The ARMAX estimation

```
load idjoint1.mat

for i =1:2500          % Down-sampling
    unew(i)=u(i*2);
end

for i =1:2500
    ynew(i)=y(i*2);
end

h=10e-3;

z=[ynew' unew'];
z=dtrend(z);           % Elimination of trends
z=dtrend(z,1);

zi=z(300:1400,:);
zv=z(1401:2500,:);

plot(cohere(unew,ynew,[],h)) % Coherence function
pause

frg=id2fr(spa(zi,100,[],[],h)); % Frequency response
bopl(frg);                     % estimated from data.
bogrid;
pause

N=[4 2 2 1];
th1=armax(zi,N,[],[],[],[],h);
present(th1);

e=resid(zi,th1);           % Plot of residuals
pause

zpplot(th2zp(th1));        % Pole-zero plot
pause

ym=idsim(zv(:,2),th1);     % Simulation
t=h*[1:1:length(zv(:,1))];
plot(t,zv(:,1),t,ym)
pause

[H Snn]=th2ff(th1);        % Frequency response
bodeplot(H);
```

B.2 The estimation of the continuous state-space model

```
load idjoint1

z=[y' u'];

par=[-10000 0 -10 100 -10 0 10000 -25]; % Initial values

th=mf2th('mymfile','c',par,[]);

model=pem(z,th,[],[],[],[],[],5e-3);

present(model)

zpplot(th2zp(model))

grid on

function [A,B,C,D,K,x0]=mymfile(par,Ts,aux)

F=[0 1 0 0;par(1) par(2)+par(3) -par(1) -par(2);0 0 0 1;
    par(7) par(6) -par(7) par(5)-par(6)];
G=[0 par(4) 0 0]';
C=[1 0 0 0];
D=0;
K=zeros(4,1);
x0=[par(8) 0 par(8) 0]';

if Ts>0, [A,B] = c2d(F,G,Ts); else A=F; B=G ; end
```

B.3 Computation of piecewise quadratic Lyapunov function using PWLTOOL

```
function [P,t,x,te,thepl]=pwl_analysis(A,B,Kc,Ti,K,D,k,
                                       xi,xc,xr,tfinal,P,global);

xinf=(K*xr+k*xc)/(K+k);

if P==1                                % P-controller
    A1=A;
    A1(2,1)=A1(2,1)-700*Kc*K;
    A1(2,2)=A1(2,2)-700*Kc*D;
    a1=B*Kc*K*(xr-xinf);
    A2=A1;
    A2(2,3)=A2(2,3)-700*Kc*k;
    a2=zeros(4,1);
    C=[0 0 1 0];
    x0=[-xinf 0 -xinf 0]';
else                                    % PI-controller => one extra state
    A1=[A zeros(4,1) ; [-K -D 0 0] 0];
    A1(2,1)=A1(2,1)-700*Kc*K;
    A1(2,2)=A1(2,2)-700*Kc*D;
    A1(2,5)=700*Kc/Ti;
    a1=[B*Kc*K*(xr-xinf); K*(xr-xinf)];
    A2=A1;
    A2(2,3)=A2(2,3)-700*Kc*k;
    A2(5,3)=-k;
    a2=zeros(5,1);
    C=[0 0 1 0 0];
    x0=[-xinf 0 -xinf 0 0]';
end

setpart('h');
switch_plane=addhp([C -(xc-xinf)]);
no_contact=addati(A1,a1);
contact=addati(A2,a2);
X_1=addhcell(-switch_plane,no_contact);
X_2=addhcell(switch_plane,contact);
part=getpart;
thepl=part2pwl(part);

fprintf('Simulating...');
[t,x,te,reg]=pwlsim(thepl,x0,[0 tfinal]);
fprintf('done.\n');

if global==1
    [P,V,L]=qstab(thepl);
else
    [P,V,L]=pqstab(thepl);
end
```

B.4 Computation of state feedback with observer

```

function [L,Ps,Qs,P,Q]=compute_fb(A,B,C,k,d,R);

% Computes the state feedback, L, according
% to the design procedure so that
%  $L \cdot \text{inv}(sI - (A - BL)) \cdot B$  is SPR.
%
% Also computes weighting matrix for Lyapunov
% function to verify stability and separation
% property.
%
% k, d, R are used for the weighting matrices
% in the Riccati equation.
%
% Must verify that matrices P, Q are positive definite.

% Compute state-feedback, L,
Qc=[k 0 0 0;0 d 0 0;0 0 0.01 0;0 0 0 0.01];

Pc=are(A,B*inv(R)*B',Qc); % Solves Riccati equation
L=inv(R)*B'*Pc;
Qcc=Qc+Pc*B*inv(R)*B'*Pc;

% Compute K, so that  $A - K \cdot C$  is stable.
oldboxes
p=polybutt(4,110);
K=place(A',C',roots(p))';

% Computation of Lyapunov function
AL=A-B*L;
AK=A-K*C;

Pco=B*inv(B'*B)*R*L;

Qco=-(Pco*AK+AL'*Pco)+L'*R*L;

Qo=Qco'*inv(Qcc)*Qco+2*L'*R*L+Qc; % To make  $Q > 0$ 

Po=lyap(AK',Qo-2*L'*R*L);

Pstemp=[Pc Pco;Pco' Po];
Ps=[Pstemp zeros(8,1);zeros(1,8) R];

As=[AL -B*L B;zeros(4,4) AK zeros(4,1);-L -L 0];
Qs=-(Ps*As+As'*Ps);

P=[Pc Pco; Pco' Po]; % These matrices should
Q=[Qcc Qco; Qco' Qo]; % be positive definite.

```


C. Simulink models

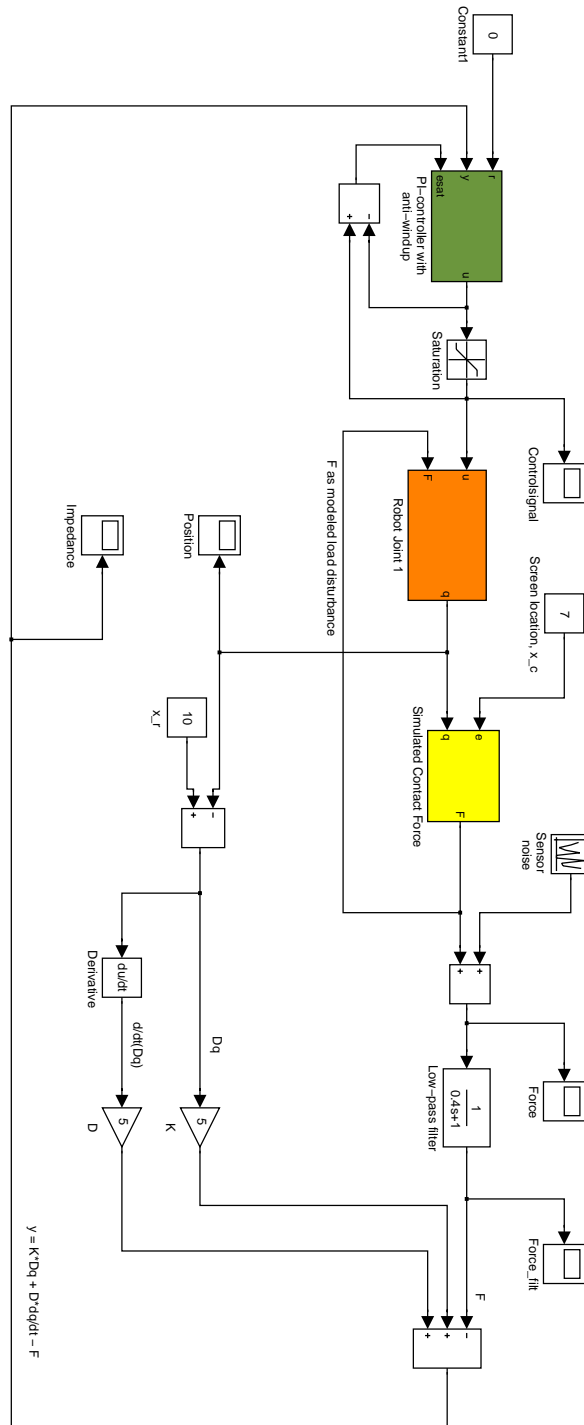


Figure C.1 Simulink model for the preliminary impedance control simulations

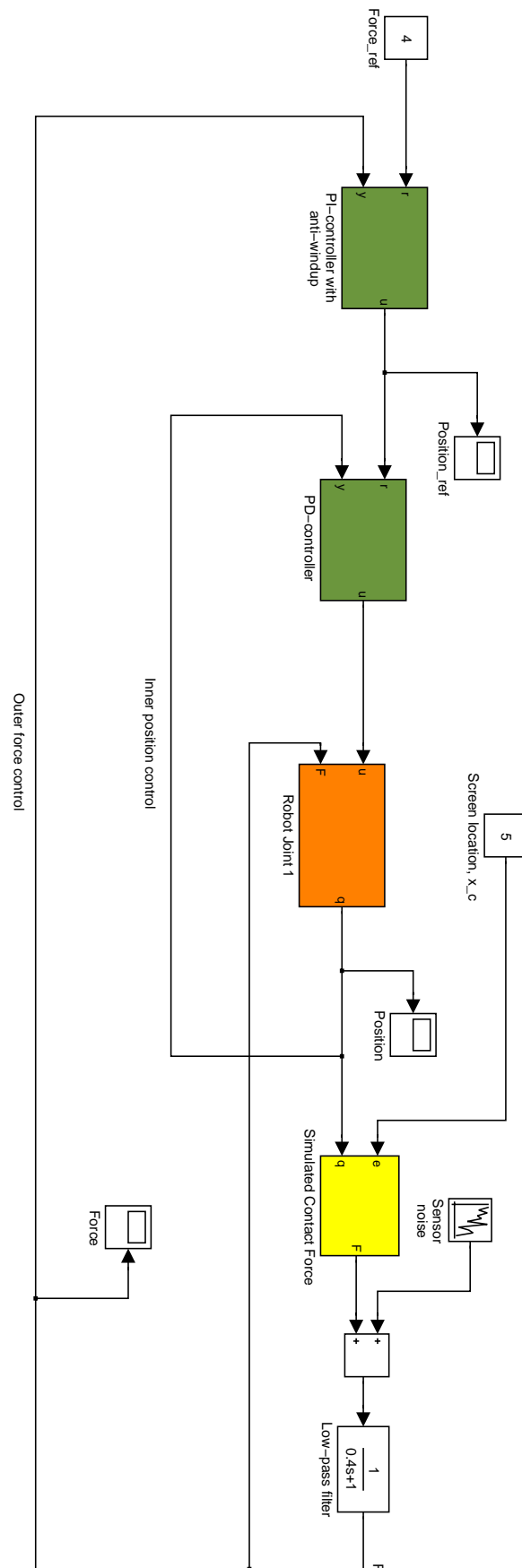


Figure C.2 Simulink model for the preliminary direct force control simulations

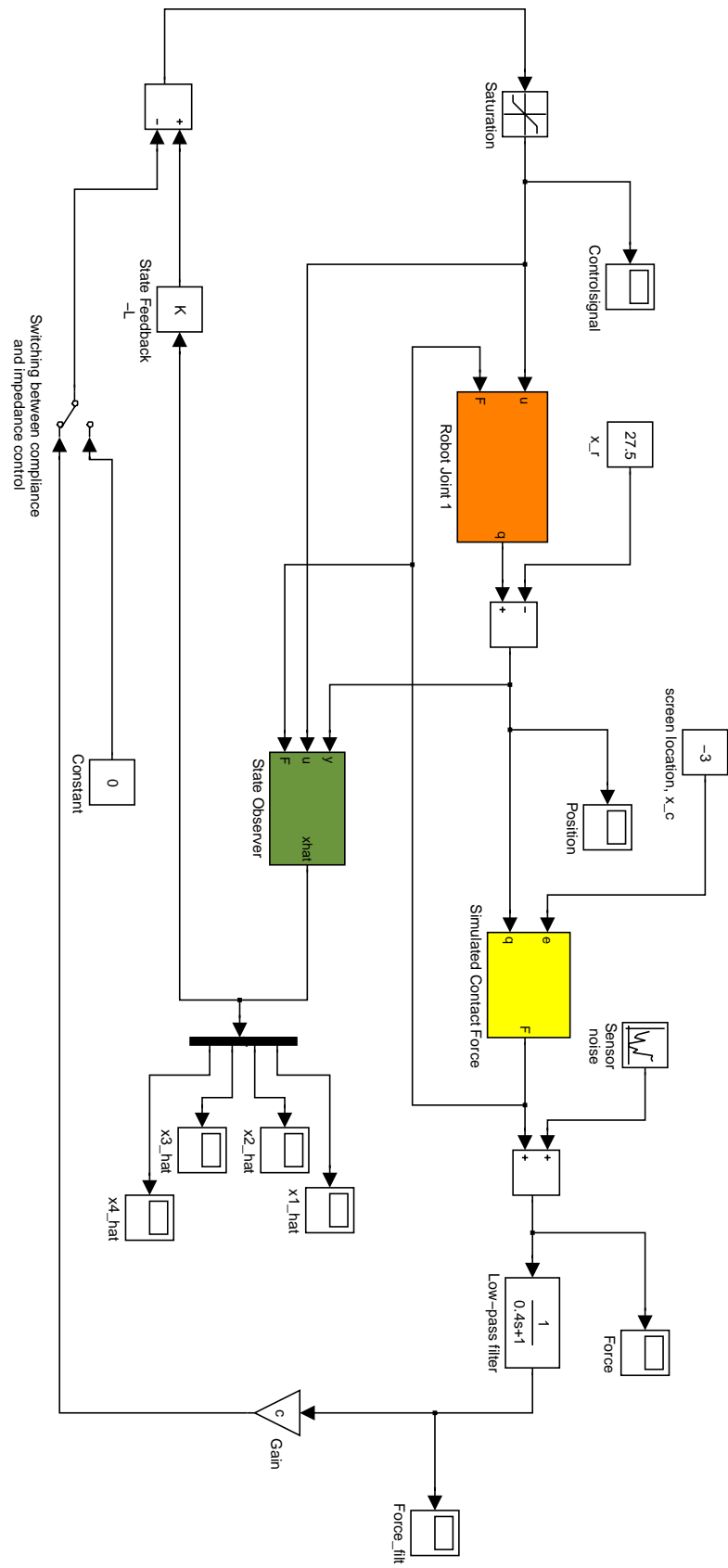


Figure C.3 Simulink model for the simulations of observer-based compliance and impedance control. The manual switch is used to change between compliance and impedance control.

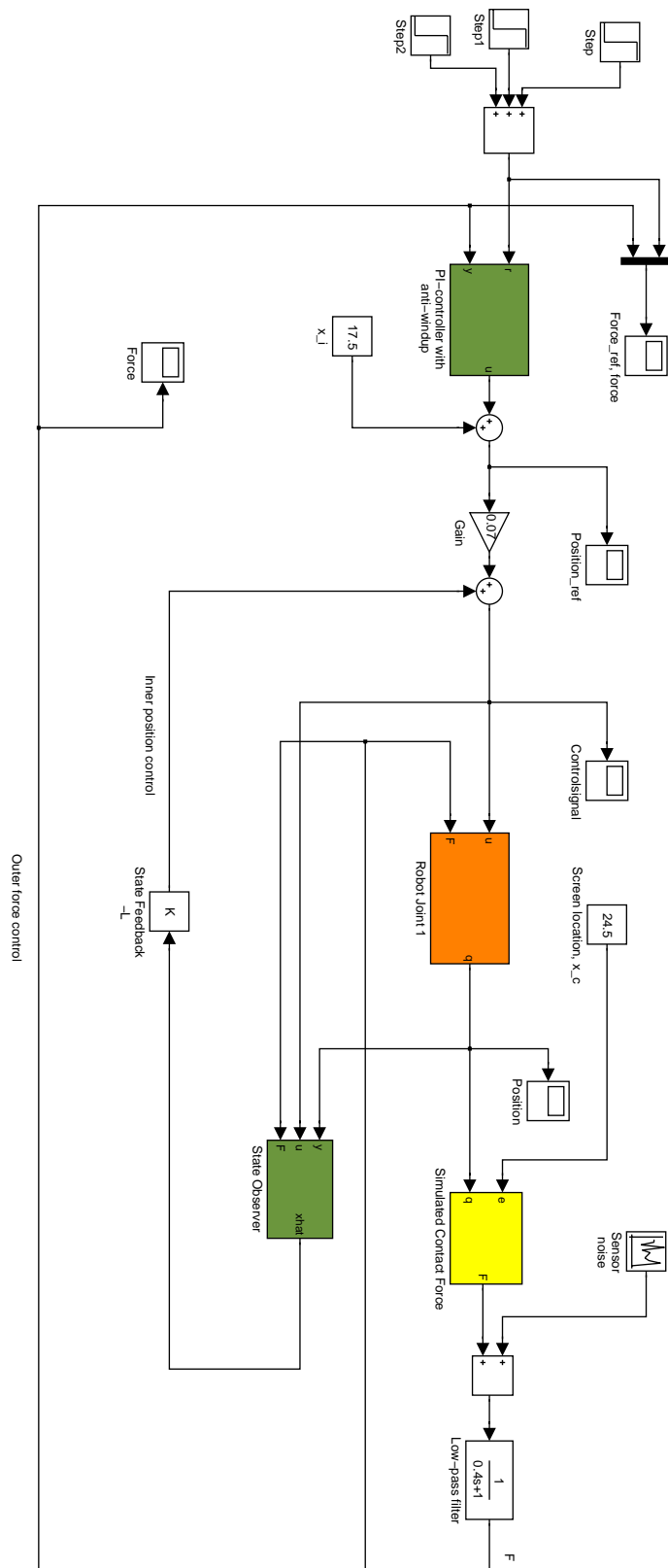


Figure C.4 Simulink model for the simulations of observer-based direct force control

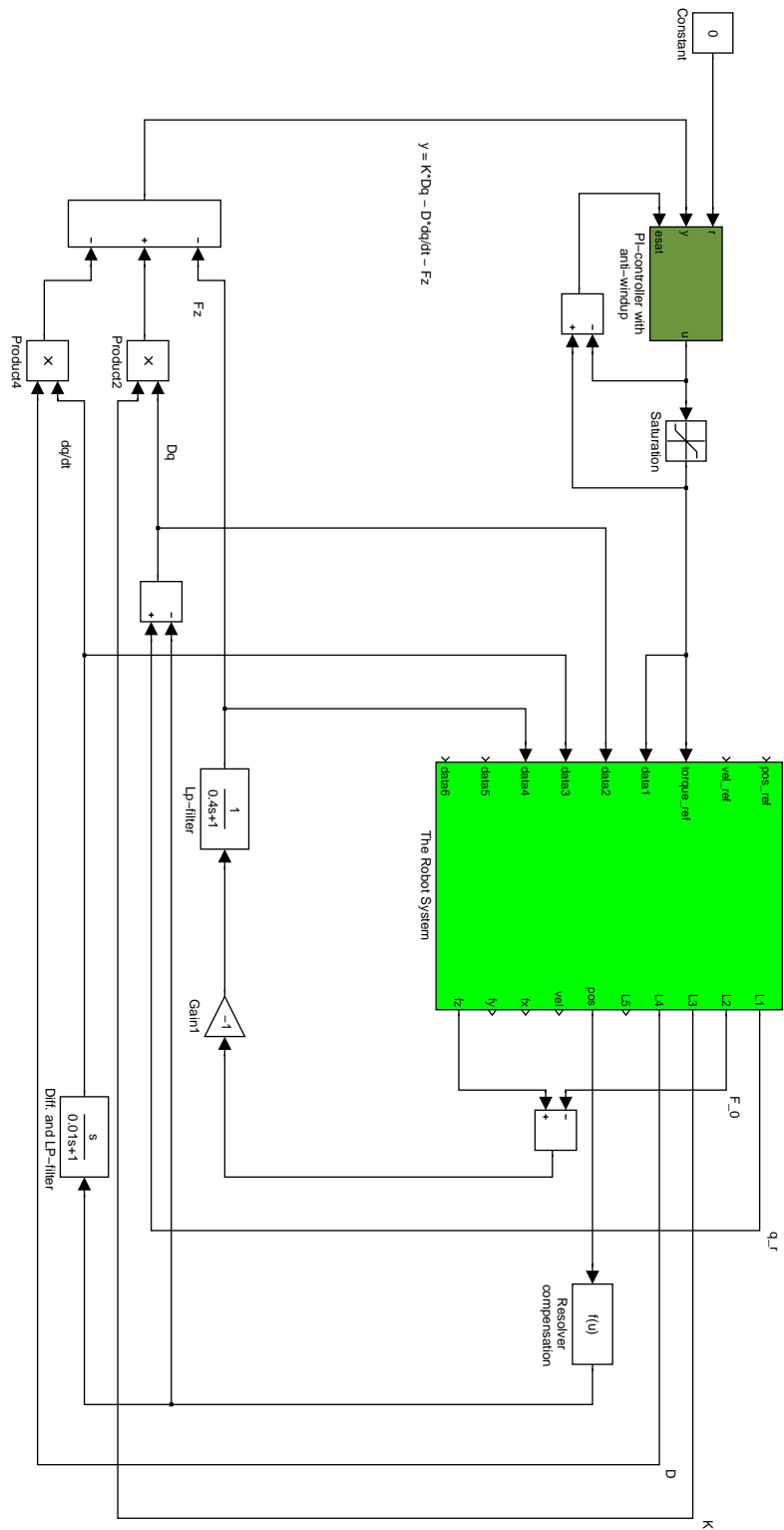


Figure C.5 Simulink model for the preliminary impedance control experiments

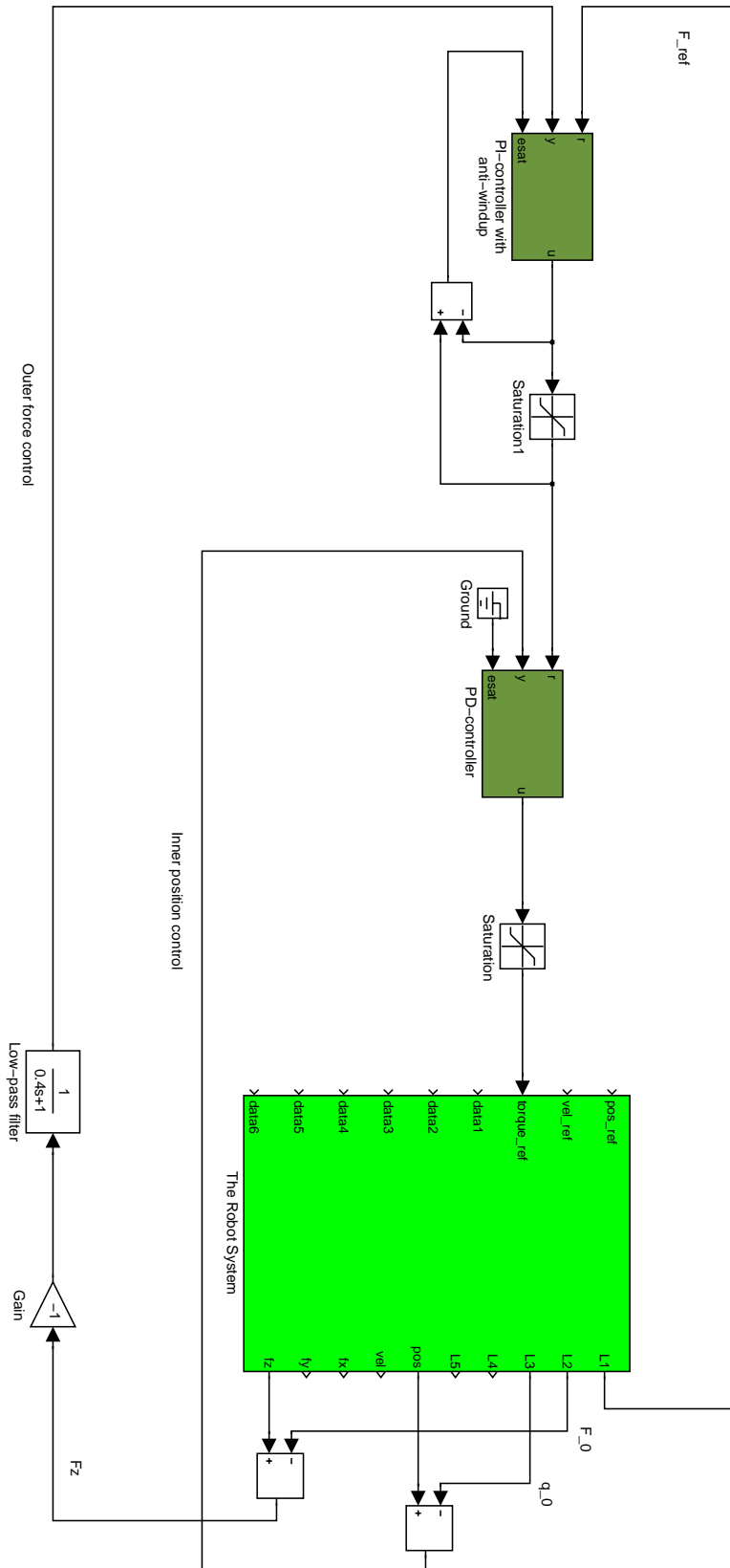


Figure C.6 Simulink model for the preliminary direct force control experiments

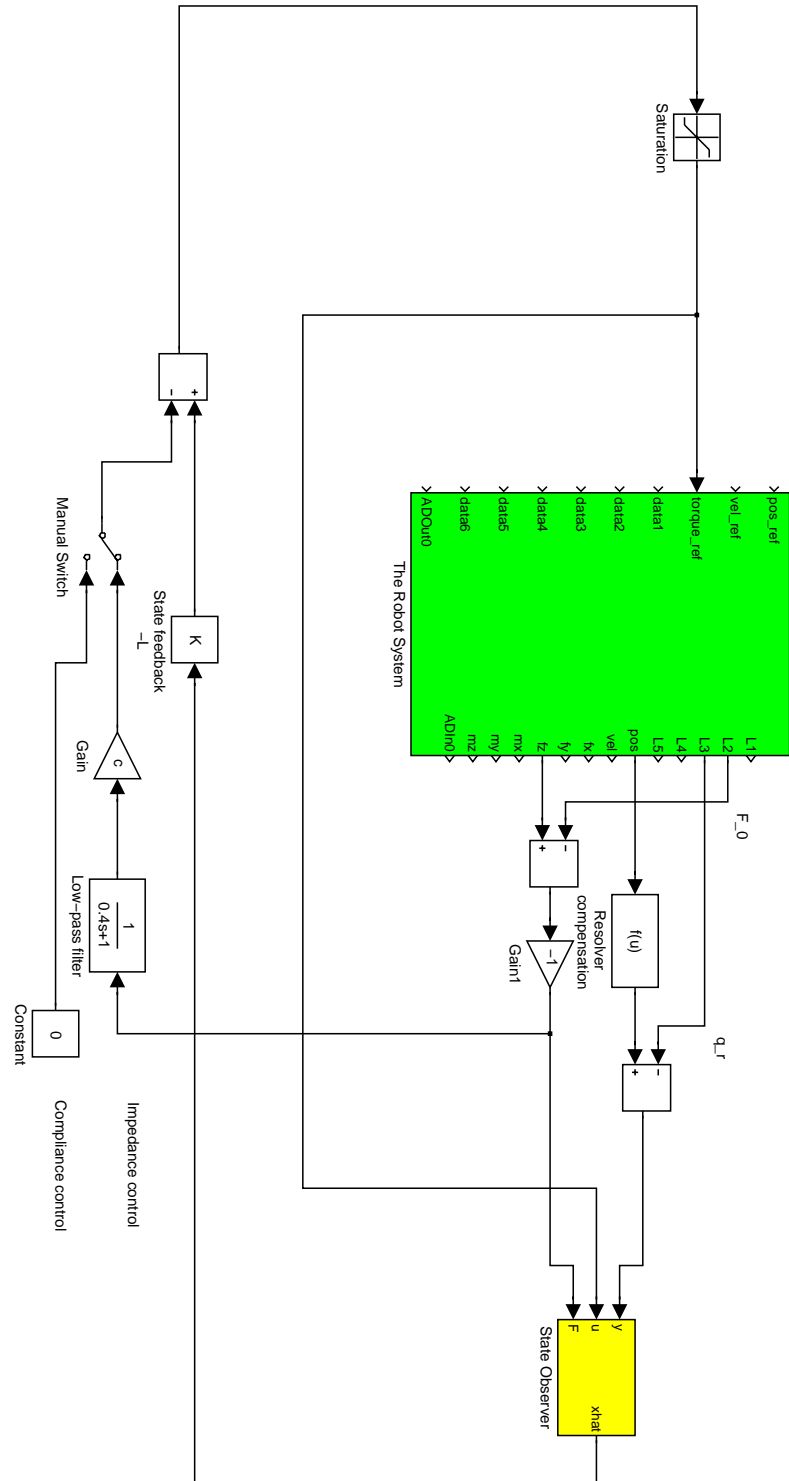


Figure C.7 Simulink model for the experiments of observer-based compliance control and impedance control. The manual switch is used to change between compliance and impedance control.

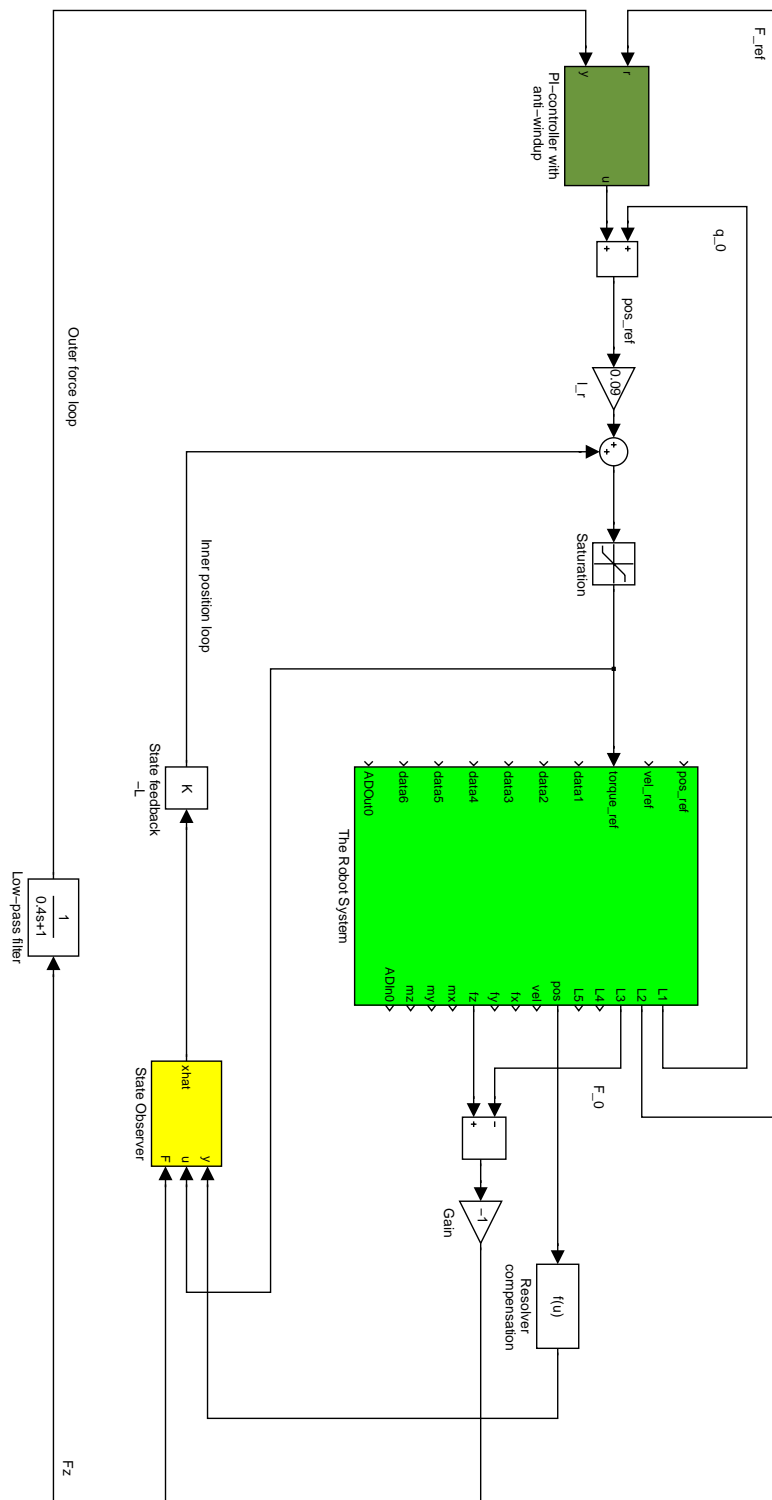


Figure C.8 Simulink model for the experiments of observer-based direct force control