NTNU
Norwegian University of
Science and Technology

Anode Milling with an Industrial Robot Manipulator

Stian Nese

Master of Science in Engineering Cybernetics
Submission date: June 2010
Supervisor: Geir Mathisen, ITK
Co-supervisor: Aksel A. Transeth, SINTEF IKT

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

This thesis assignment concerns control strategies for ABB robots which are to be used for removing deep-rooted coke on carbon anodes. A milling spindle is to be used as the robot tool.

The main tasks of this thesis are outlined as follows:

- Perform a literature survey on robot force control relevant for the thesis.
- Obtain an overview of the possibilities and limitations of the systems offered by ABB robots for doing force control.
- Choose one or more methods based on the literature survey and knowledge about the ABB robot systems as a basis for removing coke from carbon anodes.
- Adapt the chosen method(s) to an ABB robot system and suggest how the method can be implemented on the system.
- Discuss the suggested solution.
- Implement and simulate the suggested solution in ABB RobotStudio.
- Discuss the results.

Assignment given: 11. January 2010
Supervisor: Geir Mathisen, ITK

Preface

This report presents the work done on a 30 credit points master thesis, where 60 credit points makes up one year of full-time study. It is a compulsory assignment for graduate students at the Department of Engineering Cybernetics, at the Norwegian University of Science and Technology (NTNU).

The assignment covered by this report, is given by the Department of Engineering Cybernetics at NTNU in cooperation with SINTEF ICT. In addition to this, the robot vendor RobotNorge AS has also contributed to the choice of topic.

Force control in robotics is a field with a potentially great future, given its possibilities in both an industrial and non-industrial environment. It has been interesting to get an insight into such a promising field of engineering, and it will be equally interesting to follow it in the future.

All the work done on this thesis, has been conducted in an office kindly put to my disposal by RobotNorge, located in their main office facility at Klepp St.. For this, and for all the pleasant lunch hours, I would like to express my gratitude.

I would like to thank my supervisor at NTNU, Geir Mathisen, for his assistance along the way. I would also like to thank my co-supervisor at SINTEF ICT, PhD Aksel A. Transeth, for all his time and effort spent on helping and guiding me in my work. And last, but not least, I would like to thank my wife Elisabeth for her tremendous help and support over these past months.

Stian Nese
June 2010

Abstract

Force control of robot manipulators will be needed in the future to fulfill the potential of automated solutions. For this to be possible, adequate control systems for this special purpose are required. This thesis proposes an extension to an existing force control approach found in the literature, based on direct force control, for the use of an industrial ABB robot in anode milling operations.

This report presents a control system that aims to exercise force control with a robot manipulator, in order to conduct effective carbon anode milling. The control structure proposed contains both a position controller and a velocity controller, in order to enhance the final result. Because there is a gap between the theory on robot force control and the features usually available on an industrial robot, this control system is modified to be directly implementable on a standard industrial ABB robot manipulator.

Simulations, and subsequent comparison, of both this control system and a control system typically found in the industry, which is based on PID control of the milling tool power consumption, are conducted. ABB's *RobotStudio* is used to perform the simulations, where models of the tool, anode and coke are used. From these simulations we see that the industrial PID controller performs very well, and that the newly proposed control approach does not quite reach the same level of performance.

Contents

Notation

DOF	=	degrees of freedom.
q	=	vector of generalized joint coordinates.
τ	=	vector of generalized joint force.
x_d	=	vector of desired end-effector pose.
x_e	=	vector of current end-effector pose.
$\widetilde{x}$	=	vector of end-effector pose error, $x_d - x_e$.
x_{pd}	=	vector of estimated positional deviation, due to contact forces.
h_e	=	vector of contact forces exerted on the environment by the robot end-effector.
$J(q)$	=	geometric Jacobian matrix.
$J_A(q)$	=	analytical Jacobian matrix.
P_{real}	=	measured milling tool power consumption.
P_d	=	desired milling tool power consumption.

Chapter 1

Introduction

1.1 Motivation

A robot manipulator, as a mechanical structure, is never more useful than the operations it can perform. So by enabling a robot to perform a new task, it becomes more useful and increases its worth. Typical robot tasks today include welding, painting, packaging and pick-and-place. These tasks usually require a high level of control over the robot motion, where both position and velocity accuracy can be of importance.

A different set of tasks, including milling, deburring, polishing and assembly, are not as usually found within the task repertoire of an industrial robot. The reason that these tasks are not performed as often by robots as the other tasks mentioned above, is that pure motion control is often insufficient in these situations. Today most industrial robot controllers provide advanced motion control systems, while control of the interaction forces between a robot and its environment are usually limited. In interaction tasks, such as machining and assembly, it is often necessary to control both the the robot motion and the interaction forces. Hence, there is a need for robot force control systems that can accomplish these tasks.

Traditionally, production industry has been labor intensive and hard work. In times when efficiency and low cost are emphasized, this type of manual work is often not productive enough. By finding automated solutions instead, this problem can be overcome. The biggest challenge associated with such automated solutions is to find affordable ways to control robot-environment interaction forces. Solving this problem will lead to more robots in the industry, giving higher productivity and profit.

1.2 Background

The aim of this report is to derive an approach for force control of a robot manipulator, with the aim to conduct milling operations on carbon anodes. In particular, possibilities and limitations regarding force control of ABB robots have been explored, in light of a desire to use an ABB robot to perform the milling task. Carbon anodes used in aluminum industry need to have an unevenly distributed layer of coke removed from its surface before they can be utilized to produce aluminum. Today this is mostly done manually, but this can often be a hazardous operation for the worker due to various kinds of fine-grained dust, and it is not a very cost-efficient solution. Therefore an automated solution will usually be the preferred option.

Even though research on force control in robotics have been conducted for decades, there have not been a great deal of successful implementations of force control in the industry (Yoshikawa, 2000). On the other hand, motion control systems have been at the core of robot control since the beginning in the 1960s. In recent years, this motion control has become highly advanced, but for some tasks pure motion control is likely to fail anyway. Such tasks are often interaction tasks, and usually include machining and assembly.

In order to use pure motion control in interaction tasks, both the environment and the interaction forces between a robot and the environment, as well as the robot itself need to be accurately modeled, and the task needs to be accurately planned. An accurate model of the robot can to a certain degree be at hand, but an accurate model of the environment and the interaction forces between the robot and the environment are difficult to derive in most cases. If the model of the environment is not precise enough, the robot might come in unplanned contact with the environment, where the robot end-effector position will deviate from the planned position. The motion controller will then try to eliminate this error, leading to a build-up of forces between the robot and the environment. Such a situation can cause damage to both the robot and its environment, and is clearly undesirable.

Through force control, one aims to control this contact force between the robot and the environment. This control can be achieved in various ways. First of all it is possible to install a force/torque sensor on the robot, either one in each joint of the robot or at the tool. The force/torque measurements can then be feed back to the controller, which then exerts an appropriate action. A different option is to use readings from the robot actuators to calculate the applied force, from which a control action can be derived. In the industry today a third option is probably the most common, especially in tasks such a milling, grinding, deburring and polishing. The idea here is to measure the tool power consumption as an estimate of material removal, and enforce a control action based on this signal.

In the industry today there is a large potential for automated solutions in machining operations, but the control system solutions are either not good enough or come a too high a cost. Therefore robust, versatile and cost-efficient force control systems for robot manipulators are of great interest.

1.3 Contribution

A review of some of the literature available on the topic of force control in robotics, is provided in Chapter 2. This review covers the major approaches resulting from over 30 years of research, including two methods from both indirect and direct force control. In addition to this, a study of the possibilities and limitations connected to the use of an industrial ABB robot with force control, is done in Chapter 3.

From the information found in Chapter 2 and 3, a force control system is presented in Chapter 4, for the task of carbon anode milling using an industrial robot manipulator. Even though this control design is especially linked to the anode milling task, it is a general purpose control system and can potentially be used in other interaction tasks as well.

In a combination between force and motion control, the approach proposed in this report aims to control the robot-environment interaction forces through usage of the existing robot motion controller. The approach proposed in this thesis is an extension of an approach described in Chapter 2. Through an outer force control loop an estimate of the current interaction forces are obtained, without the use of a force/torque sensor. The estimated force error is then transformed into a position error, which can be used as input to the robot motion controller.

In addition to this position control loop, a velocity control loop was added, with the aim of enhancing the final result through less positional adjustments. The idea is to decrease the number of position adjustments, by reducing the robot travel velocity when the robot tool is encountering large amounts of coke. If a reduced travel velocity is enough to keep the tool within its operational window, this is likely to produce a smoother end-result than a number of positional adjustments.

An important initial condition for the work presented in this thesis, was that the control approach should, at a final stage, make use of the possibilities and handle the limitations of an ABB robot motion controller. But the control system proposed in Chapter 4 can not be directly implemented on a standard industrial robot of today, because there is a gap between the theory developed on force control, and the features available on an industrial robot.

The proposed control system has been adapted from its theoretical starting point to be implementable on a standard ABB industrial robot, in order to be useful on a wide range of robot systems. This adaptation includes coping without kinematic and dynamic robot model and access to joint actuator force/torque.

The modified control system has been implemented on a virtual ABB robot in *RobotStudio*, and its performance compared to the performance of a force control approach more common in the industry. This industry approach use the measured tool motor current to estimate the material removal, and then applies a control law on this signal.

For these simulations the velocity loop of the adapted system was omitted, in order to enable a meaningful comparison of the process time. A connection between the tool power consumption and the interaction forces was assumed, and this assumption made it possible to simulate the same environment for both control systems. Since this is an uncertain assumption, several different such connections was tested. Based on these simulations, a non-linear connection was found to be the most correct one.

1.4 Outline

The remainder of this report is organized as follows:

Chapter 2 presents a literature study on robot force control. It starts with introducing the theory usually involved in robot force control, and presents a total of four different force control approaches. Towards the end the process of deburring is briefly presented.

In Chapter 3 some of the possibilities and limitations associated with ABB robots and force control are explored. This includes special ABB software and also the ABB industrial robot controller IRC5.

In Chapter 4 a control system for the force control of a milling process is presented. This control system design is inspired by some of the control systems described in Chapter 2.

Chapter 5 presents an approach which transforms the control system proposed in Chapter 4, into a control system that can be implemented on a standard ABB industrial robot.

In Chapter 6 the adapted control system from Chapter 5 is simulated, and compared to a different approach more common in the industry.

Chapter 7 contains a discussion and a conclusion on the work done in chapters 4, 5 and 6, and also suggests ideas for further work.

Chapter 2

Review of Literature Study on Robot Force Control

This chapter is a summary of the preliminary study of literature on the topic of robot force control. Several different force control strategies have been proposed over the past 30 years, with different advantages and limitations depending on the researchers' main focus of attention. Since every strategy has its own positive and negative features, the best choice of force control strategy is very much dependent on the problem at hand. To give a broader understanding of the issues involved when dealing with force control of robots, this chapter describes several different approaches, not all of which are suited for solving the problem in question for this report. The two most common approaches, *impedance control* and *hybrid force/motion control*, are discussed along with two other approaches. The review of the literature study will be in the following order.

In Section 2.1 a short introduction on force control in robotics is given. This introduction offers a brief overview as to why force control is needed, and some challenges associated with it.

In Section 2.2 two force control strategies under the category of *Indirect force control* is presented. These strategies are *Impedance control* and *Active compliance control*.

Section 2.3 presents the force control category known as *Direct force control*. Two of the more common force control strategies from this category are presented in some detail, namely *Force control with inner motion control loop* and *Hybrid force/motion control*.

In Section 2.4 a brief overview of the process of deburring is presented. The use of robots in such operations are emphasized, and some ways of doing this are mentioned.

2.1 Introduction

When a robot is performing a task that requires it to be in contact with its environment, it is essential that this interaction is somehow controlled. Usual interaction tasks are milling, deburring, machining and assembly. By using purely motion control, which is the common strategy for control in industrial robots, interaction tasks may become very difficult. Motion control is reliant on accurate modeling of both the robot and the environment, in order to operate with the desired accuracy. While robot models can be derived with ample precision, modeling the environment is generally not an easy task. Hence the demand for a different strategy on robot control arise.

A force control strategy is one that modifies position trajectories based on the sensed contact force, or an estimation of the contact force. In robotics, force control usually presents a trade-off between freedom of motion and some desired force characteristics. Conflicting demands may occur between a desired path and a desired contact force, and how this is dealt with is strategy dependent.

Several different force control strategies have been proposed over the past 30 years. These strategies can be divided into passive and active interaction control (Siciliano and Khatib, 2008). In passive interaction control the physical nature of the robot itself is compliant towards the environment, and is therefore not reliant on a control system to modify the robot pose. This compliance can originate from flexible links or joints, or purpose built end-effector. Since the robot is made compliant through the mechanical structure of the robot, there is no need for a force/torque sensor to measure the contact force/torque. Overall this approach have a quicker response time, but it has low versatility to task changes and the mechanical compliance can only deal with relatively small positional deviation (Siciliano et al., 2009).

The opposite to passive interaction control is active interaction control, which uses a control system to provide the compliant behavior. This is often achieved by using a force/torque sensor with a feedback connection to the motion controller, in order to edit the robot path as required. Active interaction control can be split into two categories; indirect force control and direct force control.

2.2 Indirect Force Control

Indirect force control aims to achieve force control through motion control, without explicit regulation of the contact force. Instead, the idea is to change the apparent dynamic behavior of the robot, thus trying to indirectly control the interaction forces between the robot and its environment. Impedance control is the indirect force control strategy that has received the most attention in recent years (Yoshikawa, 2000), while active compliance control/stiffness control is another noteworthy approach.

2.2.1 Impedance Control

According to Hogan (1984), a system described by a function taking force as input and produces motion as output can be classified as an admittance. And conversely, a system described by a function accepting motion as input and producing force as output is an impedance. So in other words, an impedance control schema will react to a positional error by generating forces. This schema aims to regulate the coupling between the interaction force and the positional deviation originating from this contact force. The interaction between the end-effector and the environment can be presented using the joint space dynamic model of the robot (Siciliano et al., 2009)

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau - J^T(q)h_e \tag{2.1}$$

where $D \in \mathbb{R}^{n \times n}$ is the inertia matrix of the robot, $C \in \mathbb{R}^{n \times n}$ contains the Centrifugal and Coriolis terms and $g \in \mathbb{R}^n$ contains the gravity terms. $\tau \in \mathbb{R}^n$ is the generalized joint torque and $h_e \in \mathbb{R}^n$ is the contact force from the end-effector on the environment, with $J(q) \in \mathbb{R}^{n \times n}$ being the geometric Jacobian of the contact force. Note that in this representation, for simplicity, both viscous and static friction have been neglected. According to Siciliano et al. (2009) we can choose the control input τ as

$$\tau = D(q)y + C(q, \dot{q})\dot{q} + g(q) \tag{2.2}$$

where we have a new control input, $y \in \mathbb{R}^n$. Inserting (2.2) into (2.1) we end up with the expression

$$\ddot{q} = y - D^{-1}J^T h_e \tag{2.3}$$

As we can see from (2.3), in the absence of an environmental contact force we have that $\ddot{q} = y$. In other words, the new system is linear and decoupled with respect to y, and the desired end-effector position is tracking the desired position. In the presence of a contact force, we need to find a suitable expression for the new input y in order to achieve the desired system behavior. In Siciliano et al. (2009) it is suggested to take

$$y = J_A^{-1}(q)M_d^{-1}(M_d\ddot{x}_d + K_D\dot{\tilde{x}} + K_P\tilde{x} - M_d\dot{J}_A(q, \dot{q})\dot{q}) \tag{2.4}$$

where $J_A \in \mathbb{R}^{n \times n}$ is the analytical Jacobian, M_d, K_D and K_P are positive definite diagonal matrices $\in \mathbb{R}^{n \times n}$ and $\tilde{x} = x_d - x_e$. x_d is here the desired robot end-effector pose, while $x_e = \begin{bmatrix} p_e \\ \phi_e \end{bmatrix}$ is the current end-effector pose, thus making $\tilde{x}$ the end-effector pose error. From Siciliano et al. (2009) we have that the relationship between the joint space acceleration and the operational space acceleration is given by

$$\ddot{x}_e = J_A(q)\ddot{q} + \dot{J}_A(q,\dot{q}\dot{q}) \tag{2.5}$$

We can then rewrite (2.3) by inserting (2.4) and (2.5), which yields

$$M_d\ddot{\tilde{x}} + K_D\dot{\tilde{x}} + K_P\tilde{x} = M_dJ_AD^{-1}J^T(q)h_e \tag{2.6}$$

This control system can be represented, in a similar fashion as in Siciliano et al. (2009), as the block diagram shown in Figure 2.1. In (2.6) we can

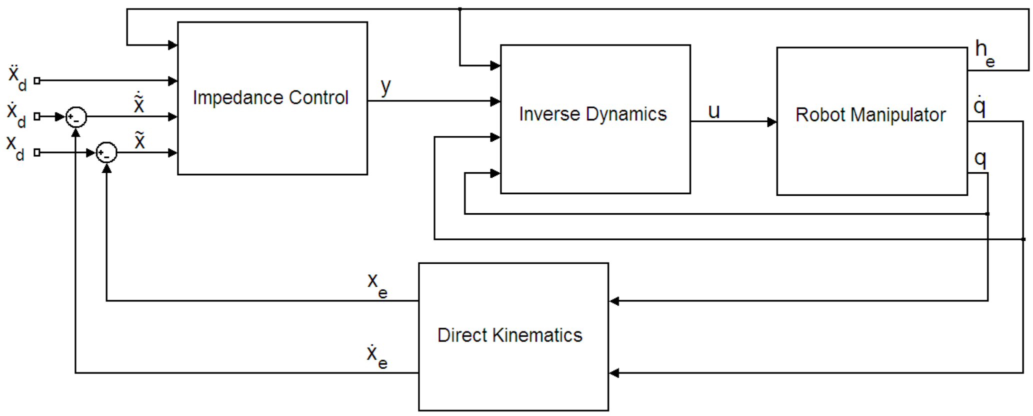

Figure 2.1: Block diagram of impedance control.

identify the force vectors on the right hand side, and a mass-spring-damper system on the left hand side with mass M_d, stiffness K_P and damping K_D. These system matrices are user defined, and can be used to enforce the desired dynamic characteristics. As a consequence, the system with the given control inputs (2.2) and (2.4) will interact with the environment just like a mechanical mass-spring-damper system, with user defined mass, stiffness and damping. When the robot end-effector comes in contact with the environment, the end-effector will have a compliant behavior and the contact forces will be bounded at the expense of positional deviations.

During interaction motion and free motion the dynamics of the controlled system changes, which can cause problems. The control system parameter estimation depends on the robot end-effector orientation, through the transformation matrix T_A (Siciliano et al., 2009). This transformation matrix can be defined as $J = T_A(\phi_e)J_A$, where it is clear that the orientation of the end-effector is influencing the transformation. Therefore the parameter estimation is a difficult task. The dependency can be avoided by redesigning the control input y in (2.4) as a function of operational space error (Siciliano et al., 2009).

2.2.2 Active Compliance Control

Compliance control aims to achieve a desired compliant behavior from the robot, either achieving this in a passive or active form. Passive compliance control is usually accomplished using some specially designed mechanical device, giving this approach low adaptability. Active compliance control on the other hand, achieves the desired compliance through specially designed control systems. This approach is thus much more versatile, and therefore more suited to the ever changing requirements found in the field of robotics. In addition, this approach does not require the robot manipulator to be equipped with a force sensor, which is sometimes preferred from an economical point of view.

Consider again the dynamic model of a robot manipulator in (2.1). From Siciliano et al. (2009) we have that a control input of

$$\tau = g(q) + J_A^T(q) K_P \widetilde{x} - J_A^T(q) K_D J_A(q) \dot{q} \tag{2.7}$$

will bring the end-effector to the desired pose in the case of no interaction forces, $h_e = 0$. However, when $h_e \neq 0$ the desired pose will not be tracked. At the equilibrium, instead of $J_A^T(q) K_P \widetilde{x} = 0$, we then get

$$J_A^T(q) K_P \widetilde{x} = J^T(q) h_e \tag{2.8}$$

If we assume that the Jacobian matrices have full rank, we can rearrange (2.8) to

$$\widetilde{x} = K_P^{-1} J_A^{-T}(q) J^T(q) h_e = K_P^{-1} T_A^T(x_e) h_e \tag{2.9}$$

where $T_A^T(x_e)$ is the transformation matrix between the two Jacobians. From Siciliano et al. (2009) we know that by defining

$$T_A^T(x_e) h_e = h_A \tag{2.10}$$

where h_A is the vector of equivalent forces, we can insert (2.10) into (2.9), resulting in

$$\widetilde{x} = K_P^{-1} h_A \tag{2.11}$$

From (2.11) we have that the system will act as a generalized spring, with K_P^{-1} being the compliance with respect to h_A. By assuming the compliance matrix K_P to be diagonal, we have from Siciliano et al. (2009) that the linear compliance is independent of pose, while the torsional compliance is dependent on the current end-effector pose. Note that the linear compliance is due to force components, while the torsional components are due to moment components.

When the robot is operating in an environment, that environment will have certain stiffness/compliance properties which can be modeled into a stiffness matrix K_E. The interaction between the robot manipulator and its environment is then affected by both the manipulator compliance and the environment stiffness. This relationship can be altered through the manipulator compliance matrix K_P^{-1}, in order to make either the robot or the environment dominant. In interaction tasks this can be especially useful, since it is then possible to specify directions in which the robot or the environment has to comply, by changing the values of K_P^{-1}.

Parameter estimation for active compliance control depends on the robot end-effector orientation, just like for impedance control, and is therefore difficult. The solution proposed for impedance control, to define the control input as a function of operational space error, can also be used in this case.

2.3 Direct Force Control

Direct force control aims at controlling the contact force to a predefined value, in most cases utilizing a force sensor to measure the current contact force. The essence of direct force control is that the measured force errors are transformed to a force/torque to be applied to the robot motors. The major approach in direct force control is Hybrid force/motion control (Yoshikawa, 2000), while another interesting strategy involves a outer force control loop with a inner motion control loop.

2.3.1 Force Control With Inner Motion Control Loop

One approach that can be useful in many implementations, is to use a setup where an outer force loop is converting the measured force error into a desired motion. This motion is then executed by a inner motion control loop. Converting the force error to motion can be done in several different ways, and depends on whether or not a dynamic model of the robot is available. If the dynamic model of the robot is known, the procedure presented in 2.2.1 can be followed. This procedure assumes the same dynamic model as in (2.1), and the control input in (2.2). In the case of inner position control loop, the input y becomes (Siciliano et al., 2009)

$$y = J^{-1}(q)M_d^{-1}(-K_D\dot{x}_e + K_P(x_F - x_e) - M_d\dot{J}(q,\dot{q})\dot{q}) \tag{2.12}$$

where the new term $x_F \in \mathbb{R}^n$ defines the relationship between the desired contact force $f_d \in \mathbb{R}^n$ and the estimated current contact force $f_e \in \mathbb{R}^n$. This relationship is depending on the compliance matrix $C_F \in \mathbb{R}^{n\times n}$,

$$x_F = C_F(f_d - f_e) \tag{2.13}$$

where the estimation of the contact force, exerted by the robot end-effector on the environment, is based on an elastic model of the environment. If one assumes that during interaction the contact will only produce forces and no torques, the environment can be modeled as (Siciliano et al., 2009)

$$f_e = K_s(x_e - x_r) \tag{2.14}$$

where $K_s \in \mathbb{R}^{n\times n}$ is the stiffness matrix of the environment and $x_r \in \mathbb{R}^n$ is the robot end-effector reference position. Inserting (2.12) into the dynamic model of the robot (2.1), again using (2.5), we get

$$M_d\ddot{x}_e + K_D\dot{x}_e + K_P x_e = K_P x_F \tag{2.15}$$

and by inserting (2.13) and (2.14) into (2.15) we end up with

$$M_d\ddot{x}_e + K_D\dot{x}_e + K_P(I + C_F K_s x_e)x_e = K_P C_F(K_s x_r + f_d) \tag{2.16}$$

In Figure 2.2 we can see the block diagram version of (2.16), where the

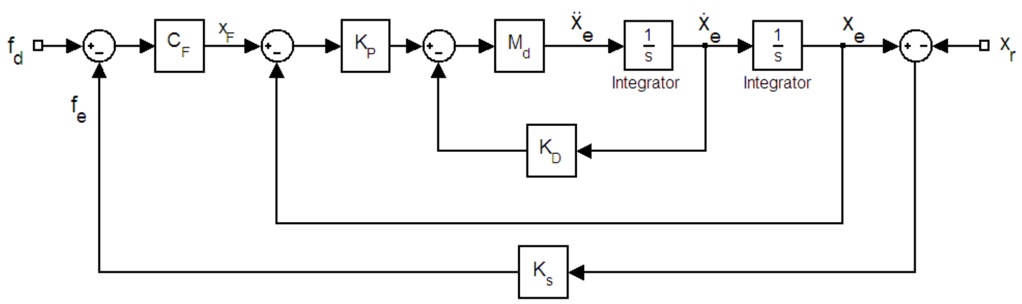

Figure 2.2: Block diagram of force control with inner position control loop.

two user defined inputs are desired contact force f_d and reference position x_r. For this control scheme to operate properly, the desired contact force is required to be consistent with the geometric characteristics of the environment. Whether or not this is possible to achieve is problem dependent, and could pose a major problem during implementation.

If instead a inner velocity loop is desirable, Siciliano et al. (2009) gives us that the input y can be chosen as

$$y = J^{-1}(q)M_d^{-1}(-K_D\dot{x}_e + K_P x_F - M_d\dot{J}(q,\dot{q})\dot{q}) \tag{2.17}$$

which, using the same assumptions as above, leads to the dynamic system model

$$M_d\ddot{x}_e + K_D\dot{x}_e + K_P C_F K_s x_e = K_P C_F(K_s x_r + f_d) \tag{2.18}$$

Note that the difference from (2.16) to (2.18) is that the position feedback loop has been opened, and x_F is now a force regulated velocity input instead of a position input. This new approach is shown in Figure 2.3.

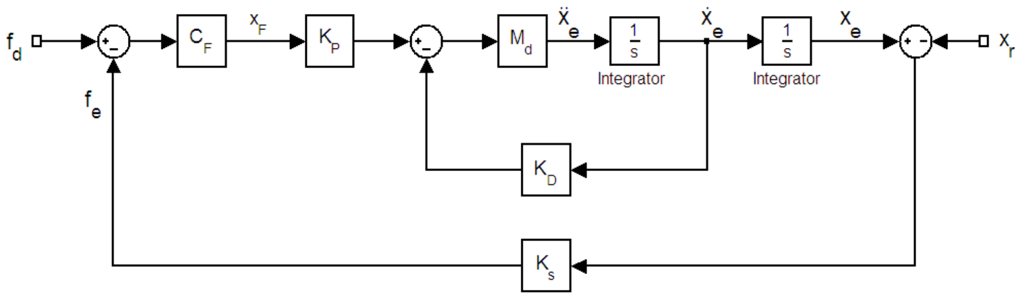

Figure 2.3: Block diagram of force control with inner velocity control loop.

The conversion from force error to position change depends on the configuration of the compliance matrix C_F. Several different configurations are possible, but the more natural choices includes variations of proportional-, integral- and damping control. In the case of a inner velocity control loop a proportional control law can be used, resulting in the desired contact force achieved in steady state. For the case of a inner position control loop, it is a bit more complicated. Using a simple P-controller, we are likely to encounter the problem of deviations in steady state. In other words f_e would never reach f_d, so the desired contact force would never be achieved. Apart from the simple P-controller, the most usual variations are PI, PD or full PID control. With P control, the C_F term can be stated as

$$C_F = K_F \tag{2.19}$$

while when adding integral action, (2.19) can be rewritten as Siciliano et al. (2009)

$$C_F = K_F + K_{int} \int^t (\cdot) d\varsigma \tag{2.20}$$

In (2.19) and (2.20) we have the proportional gain K_F, and in (2.20) we have the integral gain K_{int}. To eliminate the steady state error, including integral action in the controller would be useful. Whether or not to use a simple P controller, a full PID controller, or the slightly simpler PI controller, depends on the specifications of problem at hand, and therefore cannot be answered in general.

To determine the values of the unknown matrices of this approach can be a challenging task. The stiffness matrix K_s is in many cases likely to be very hard to find analytically, and experimental values would have to be used. Also, the positive definite diagonal mass matrix M_d, spring matrix K_P and the damping matrix K_D would have to be defined, just as in Section 2.2.1.

2.3.2 Hybrid Force/Motion Control

In many robotic tasks the environment imposes certain constraints on the robot motion, while the successful execution of the task is likely to require additional constraints. This natural partitioning of the motion constraints can be used if one designs a force control strategy. In Siciliano et al. (2009) this partitioning is defined as:

- *Natural constraints* are those constraints given by the task geometry, and they can be either motion constraints or force constraints. A velocity constraint implies that the environment does not allow a translation along a direction or a rotation about an axis, while a force constraint implies that the exertion of a force in a direction or a torque about an axis is not allowed.

- *Artificial constraints* include all the constraints not defined as natural constraints. The artificial constraints can be used by the control system to obtain the desired robot behavior.

For a generalized surface, the natural motion constraints are along the normals to the surface, while the force constraints are along the tangents to the surface (Raibert and Craig, 1981). With the definitions of natural and artificial constraints in hand, it is possible to specify a given task in terms of these constraints. By doing this, one is essentially dividing the task into a motion controlled subspace and a force controlled subspace. In other words, with a hybrid force/motion controller both the contact force and the end-effector motion are controlled in independent subspaces. This decomposition of robot axes into force and motion controlled direction is the essence of hybrid force/motion control (Patarinski and Botev, 1993).

In order to split the degrees of freedom (DOF) of the robot into two subspaces, selection matrices can be used. Assuming n DOF, if $\lambda_d \in \mathbb{R}^m$ and $\nu_d \in \mathbb{R}^{n-m}$ represent desired force and velocity, respectively, then the desired end-effector force $h_d \in \mathbb{R}^n$ and velocity $v_d \in \mathbb{R}^n$ can be stated as (Siciliano et al., 2009)

$$h_d = S_f \lambda_d \tag{2.21}$$

$$v_d = S_v \nu_d \tag{2.22}$$

where the selection matrices $S_f \in \mathbb{R}^{n \times m}$ and $S_v \in \mathbb{R}^{n \times n-m}$, have to be defined based on the task geometry.

For the purpose of designing the hybrid control system it can be useful to rewrite the dynamic model in (2.1), with respect to the end-effector acceleration. The end-effector acceleration is given by Siciliano et al. (2009) to be

$$\dot{v}_e = J(q)\ddot{q} + \dot{J}(q)\dot{q} \tag{2.23}$$

which is analogous to (2.5).

This hybrid force/motion approach is often labeled the *acceleration-resolved approach*. Rearranging (2.1) and inserting for $\ddot{q}$ in (2.23) yields

$$DJ^{-1}\dot{v}_e = -C\dot{q} - g + \tau - J^T h_e + DJ^{-1}\dot{J}\dot{q} \tag{2.24}$$

Through the insertion of $\tau = J^T \gamma_e$ into (2.24), where γ_e is the end-effector force resulting from individual joint forces, and multiplying by J^{-T}, we end up with

$$J^{-T}DJ^{-1}\dot{v}_e + J^{-T}C\dot{q} + J^{-T}g - J^{-T}DJ^{-1}\dot{J}\dot{q} = \gamma_e - h_e \tag{2.25}$$

For simplification, we can rewrite (2.25) as

$$D_e(q)\dot{v}_e + \eta_e(q, \dot{q}) = \gamma_e - h_e \tag{2.26}$$

where

$$D_e = J^{-T}DJ^{-1}$$

$$\eta_e = J^{-T}(C\dot{q} + g) - D_e\dot{J}\dot{q}$$

Further hybrid force/motion control design depends on the dynamic properties of the robot environment. Because this report is concerned with anode milling, which can be defined as a rigid environment, it will only consider the case of a rigid environment. For control systems with compliant environment, the readers are referred to Siciliano et al. (2009).

Given a rigid environment, the contact force from the end-effector on the environment can be stated as

$$h_e = S_f \lambda \tag{2.27}$$

In order to single out the desired force λ, we have to introduce some additional equations. First we have that the time derivative of the holonomic constraints ($\varphi(q) = 0$) is

$$J_\varphi(q)\dot{q} = 0 \tag{2.28}$$

Secondly we have from Siciliano et al. (2009) that the force selection matrix S_f can be stated as

$$S_f = J^{-T}(q)J_\varphi^T(q) \tag{2.29}$$

We can then combine (2.28) and (2.29) to give

$$S_f^T v_e = J_\varphi(q)J^{-1}(q)J(q)\dot{q} = 0 \tag{2.30}$$

where the relationship $v_e = J(q)\dot{q}$ between operational space and joint space velocity have been used. Taking the time derivative of (2.30) yields

$$S_f^T \dot{v}_e + \dot{S}_f^T v_e = 0 \tag{2.31}$$

If we rearrange and insert for (2.27) in (2.26), and also insert for $\dot{v}_e$ in (2.31), this will result in

$$\begin{aligned} &S_f^T(D_e^{-1}(\gamma_e - S_f\lambda) - D_e^{-1}\eta_e) + \dot{S}_f^T v_e = 0 \\ &S_f^T D_e^{-1}\gamma_e - S_f^T D_e^{-1}\eta_e + \dot{S}_f^T v_e = S_f^T D_e^{-1} S_f \lambda \\ &\lambda = D_f(S_f^T D_e^{-1}(\gamma_e - \eta_e) + \dot{S}_f^T v_e) \end{aligned} \tag{2.32}$$

where $D_f = (S_f^T D_e^{-1} S_f)^{-1}$. As we can see, the force multiplier λ is dependent on the joint actuator inputs γ_e.

Moving a step further we can insert (2.27) into (2.26), and apply the expression just found for λ in (2.32). This results in a new dynamic model,

$$D_e(q)\dot{v}_e + S_f D_f(q)\dot{S}_f^T v_e = P(q)(\gamma_e - \eta_e(q, \dot{q})) \tag{2.33}$$

which incorporates the environmental constraints imposed on the robot. The matrix P is defined as $P = I_6 - S_f D_f S_f^T D_e^{-1}$, with I_6 being a (6×6) identity matrix. Since $PS_f = 0$, the matrix P is a projection matrix filtering out all end-effector interaction forces lying in the range of S_f (Siciliano and Khatib, 2008). By choosing a fitting control force γ_e and using (2.27), one can achieve direct control of the m components of the end-effector interaction force that contravenes the constraints.

In addition to the force control achieved above, this approach aim at controlling the motion as well. Since we have m directions that are force controlled, we have $6-m$ that are velocity controlled. According to Siciliano and Khatib (2008), we can premultiply (2.33) by S_v^T and expressing the end-effector acceleration as

$$\dot{v}_e = S_v \dot{\nu} + \dot{S}_v \nu \tag{2.34}$$

in order to get the velocity based model

$$D_v(q)\dot{\nu} = S_v P(\gamma_e - \eta_e(q, \dot{q}) - D_e(q)\dot{S}_v \nu) \tag{2.35}$$

where $S_v^T P = S_v^T$ and $D_v = S_v^T B_e S_v$ have been used. To get a similar form as (2.35), we can use the identity $\dot{S}_f^T S_v = -S_f^T \dot{S}_v$ (Siciliano et al., 2009) to rewrite (2.32) as

$$\lambda = D_f(q) S_f^T D_e^{-1}(\gamma_e - \eta_e(q, \dot{q}) - D_e(q)\dot{S}_v \nu) \tag{2.36}$$

Based on (2.35) and (2.36) we can according to Siciliano and Khatib (2008) choose the control input γ_e as

$$\gamma_e = D_e(q) S_v \alpha_\nu + S_f f_\lambda + \eta_e(q, \dot{q}) + D_e(q)\dot{S}_v \nu \tag{2.37}$$

where the terms α_ν and f_λ are new control terms for velocity and force, respectively. Inserting (2.37) into (2.35) and (2.36) we get the following expressions

$$\dot{\nu} = \alpha_\nu$$
$$\lambda = f_\lambda$$

which shows that the choice of control input γ_e results in a orthogonal decoupling between the subspaces of force and velocity controlled directions. The block diagram for this control approach is shown in Figure 2.4. The desired force and motion can be specified using the variables λ and ν and imposing a control on this, for example a type of PID controller.

An issue with this approach is that under certain conditions the inverse selection matrices S_f^{-1} and S_v^{-1} can be undefined. Therefore it can be useful to employ pseudo-inverse matrices instead (Siciliano et al., 2009). Such a pseudo-inverse matrix can be defined as

$$A^+ = (A^T W A)^{-1} A^T W$$

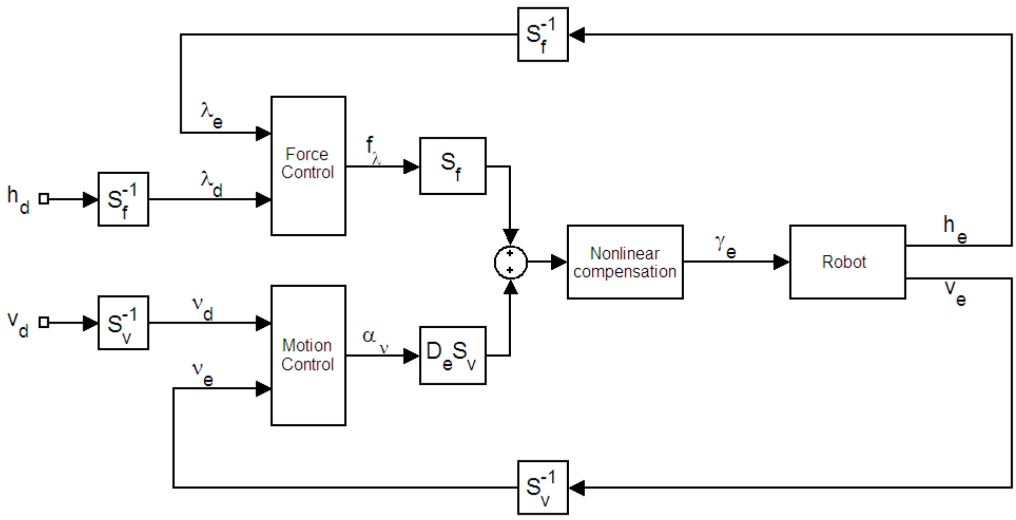

Figure 2.4: Block diagram of hybrid force/motion control.

where W is a positive definite, symmetric weighting matrix.

Another problem with this hybrid approach is that the decomposition is based on the assumption of an ideal environment, with rigid and frictionless interaction between the robot and the workplace (Siciliano and Valavanis, 1998). In a real-world application, this is obviously not the case, and this assumption can therefore pose a problem. Non-ideal environment, along with environment dynamics and errors in the modeled geometrics, can result in contact forces in motion controlled directions and motion in force controlled directions.

2.4 Deburring

Milling and deburring of parts in a production process is a common task in the industry today. Over the years the demand for increased quality of machined parts have been steadily growing, along with the desire to reduce the cost per unit (Aurich and Dornfeld, 2010). This type of challenging tasks will in most cases be performed manually, even though it is labor intensive work and not a cost efficient solution (Abou-El-Ela and Isermann, 1996). Deburring is a non-productive operation, and can account for up to 35% of the total part production price in some industries (Kazerooni et al., 1986). In addition to this it can also be a hazardous operation for the worker, given that he/she can be exposed to various kinds of fine-grained dust. To avoid all these problems, the task operation can be automated through the use of robot manipulators.

According to Aurich and Dornfeld (2010), the current approaches in deburring today can be divided into two categories; rigid tools and compliant tools. The main difference between these two approaches, is that rigid tools use the robot actuators to generate the tool force, while the compliant tool gets its force from an independent device. This means that with a rigid tool, there exists a coupling between the tool force and the robot motion, while with a compliant tool such a coupling is not present. This coupling can lead to a reduced performance, and eventually to situations where the tool or part can be damaged.

A compliant tool can be either passive or active. Passive compliant tools are not actively regulated, but relies on its compliance to achieve the desired contact force. This is a cheap approach, but not a very versatile one. Active compliant tools are actively regulated using special actuators to achieve the desired contact force. According to Aurich and Dornfeld (2010) there are several compliant tools on the market. Radial-compliant and axial-compliant tools are two types of such compliant tools available.

An approach proposed by Abou-El-Ela and Isermann (1996) aims to control the deburring process using only the signals available from the cutting tool. Analyzing these signals, the aim is to use them to control the robot motion and thereby the tool cutting depth. The concept of using information available from the cutting tool to control the robot motion, is probably one of the more common strategies used in the industry today. One way to do this is to measure the tool motor current, and use this as an estimate for how much material the tool is currently removing. From Kazerooni et al. (1986) we have that there exists a proportional relationship between the volume of material to be removed and the cutting force required, at constant robot velocity. Therefore it is possible to estimate a desired tool motor current, in order to regulate the actual motor current towards this value. This regulation can then be implemented using for instance a PID controller.

Chapter 3

Force Control with ABB robots

Different robot vendors have different robot design and structure, and the available software also differs. Since this report is concerned with a task that is to be carried out by a ABB robot, this chapter will present some of the aspects of interest when using force control with ABB robots. There are both possibilities and limitations associated with using an ABB robot for this purpose, and this chapter will highlight some of them. The chapter is organized as follows.

In Section 3.1 some software solutions for force control, developed by ABB, is presented. In particular, three different approaches to force control are covered, with differing strengths and weaknesses. The software is aimed at easing the use of robots in machining and interaction tasks, and can also be used as inspiration in other force control strategies.

In Section 3.2 the control system of the ABB robots are inspected, in terms of force control possibilities and limitations. The robot controller in question is *IRC5*, ABB's 5^{th} generation robot controller.

3.1 ABB Force Control Software

There exists some software solutions that can be useful for the purpose of force control of robot manipulators. This chapter will present three such solutions, all developed by ABB. First *SoftMove* is presented, which is an application to lower the stiffness of the robot in a given direction. Then two different applications from the *Force Control for Machining* package is presented, each with different properties and areas of applicability.

3.1.1 SoftMove

SoftMove is an interesting concept developed by ABB, and is an optional add-in to the robot controller. The idea is to make the robot compliant in one Cartesian direction, while the other directions are controlled as usual. Any one of the Cartesian directions can be made compliant. To achieve this the robot is made to act as a spring in the desired compliant direction, with the stiffness of this spring being determined by the programmer. A high stiffness makes the robot less compliant, while a stiffness of zero will make the robot float in the given direction (ABB Robotics, 2008b).

Areas of application for this functionality can be in picking up a from a machine, or when the robot is to follow a defined path while allowing interaction with the environment in one direction. This can typically be the case when the path contains uncertainties in one direction, for example with variations in the size of parts in a production series.

Because the path might change and the speed might be reduced, including SoftMove in a program can lead to an increase in the process cycle time. On the other hand, SoftMove can in many cases reduce the time spent on programming the robot process, so it can become a trade-off between development time and execution time. When programming a path using SoftMove, this path may not be followed with the usual accuracy. Because of this special nature of the SoftMove motion, some features such as *Collision Detection* and *Force Control* can not be active when SoftMove is enabled (ABB Robotics, 2008b).

3.1.2 Force Control for Machining Package

The software package *Force Control for Machining* contains two different applications for exerting force control on a robot manipulator. The first is called *Force Control Pressure*, later referred to as FC Pressure, and the second *Force Control SpeedChange*, later referred to as FC SpeedChange. Both of these applications can be applied to different machining tasks, such as grinding, milling and deburring.

FC Pressure, like the name suggests, aims to keep a given pressure from the robot tool to the surface of the work piece. This pressure, or contact force, is maintained through adjustments of the pre-programmed robot path (ABB Robotics, 2006a). When, for example, using a robot to polish a surface, this constant contact force is likely to result in a smooth end-product.

For the robot to be able to maintain the desired pressure against the surface, it requires some sort of feedback on the current contact force with the environment. With the Force Control for Machining package, it is assumed that this feedback is provided by a force/torque sensor. This sensor can be mounted either on the wrist of the robot, or in a fixed location in the environment.

With FC SpeedChange, the robot travel *speed* is changed rather than the robot path. So when large contact forces are encountered, the robot slows down from its original speed until contact force is below the limit. The advantage of this approach is that the planned path can be followed accurately, which in certain applications can be very useful. One example can be when removing unevenly distributed material, where deviations from the robot path can have a negative effect on the end-product. In a milling operation, the robot might need to slow down if there is much material to be removed.

Just as with the FC Pressure, the FC SpeedChange requires a feedback as to the current contact force. In a milling process this can be in form of a signal from the tool, representing the power consumption by the tool. Or it can be a wrist-mounted force/torque sensor, transmitting its measurements back to the robot controller.

The control system for this force control approach is based on a set of rules about the robot behavior, in a rule-based logic controller (ABB Robotics, 2006a). By increasing the number of rules, one can decrease the step from one controller output to the next. At the same time, this can make the response time slower. The controller simply compares the desired contact force with the measured one, and applies the controller output associated with the rule that is activated. In this case the controller output is a percentage of the original robot speed, from 0 to 100, so the robot can not set the speed above the original path speed.

3.2 ABB Robot Control System

The controller for new ABB robots today is the IRC5 robot controller. This is the 5^{th} generation of ABB industrial robot controllers, and it replaced the old S4CPlus, inheriting most of its predecessors' design and properties (ABB Robotics, 2004). In addition to this some new ideas and capabilities have been added, such as *SafeMove*, *TrueMove* and *QuickMove* (ABB Robotics, 2004). SafeMove is a combined software and electronics safety system that enables the robot and its operator to collaborate closely together, aiming to minimize the risk of injuries. TrueMove is an embedded function to make the robot track its planned path accurately, independent of the robot speed. QuickMove is a feature aiming to reduce the process cycle time, by keeping maximum acceleration all along the robot path. The operating system for

IRC5 is *RobotWare* OS, which has a range of options on e.g. advanced motion control, I/O control and communication.

Just like the S4CPlus controller, the IRC5 is designed in modules. Separate physical unites contain the control module and the drive module for the robot. One drive module drives one robot, while with IRC5 one control module can control up to four drive modules. To enable the control of up to four robots simultaneously, IRC5 contains *MultiMove* embedded in its motion control functionality. MultiMove enables coordinated control of up to four robots, which can be set to operate individually or in groups. This use of a single control unit for multiple robots means that the external network only has one entry point to the whole robot cell, simplifying network design and connectivity.

The controller module contains the CPU and the service port, along with operator panel and space for additional costumer equipment. Because the control module is based on an open system architecture, with commercial Intel processor and a PCI bus, the upgrade to new technology has been simplified. Inside the drive module lies the drive unit of the robot, and also the axis computer.

While the IRC5 controller has an advanced motion controller, it is not as accomplished when it comes to force control. The force control packages offered by ABB require some additional hardware to be added to the IRC5, including a PCM (PCI Mezzanine Card) to communicate with a wrist-mounted force/torque sensor. At the core of IRC5 and its RobotWare OS is ABB's *RAPID* robot programming language. This is a well-documented and flexible high-level programming language, and like IRC5 it is most suited for motion control.

When it comes to force control, sampling frequency and response time can play an important role. If the robot is operating in a rigid environment with high stiffness, a high response time might allow excessive contact forces to build up. This situation can lead to either the robot or its environment being damaged, which in turn will decrease the productivity of the robot cell. A higher environment stiffness requires a lower robot response time for successful operation.

Another feature that could be useful in connection to force control of robots, is the possibility to update the current robot target a any given time. As seen in Chapter 2, some force control approaches uses information about interaction forces to change the path and motion of the robot. In IRC5 there is no such feature to update the robot path in real-time, without having to make special arrangements of any kind. What is possible, is to use the function *SpeedRefresh*, which enables the programmer to change the robot velocity in real-time to a percentage of the current robot velocity. This option can only lower the robot velocity, since the new velocity must be within 0-100% of the original robot velocity.

In ABB Robotics (2008a) the response time for the IRC5 controller is said to be 10-100 milliseconds (ms), while ABB Robotics (2004) states that the normal response time on movement instructions are 5-120 ms, depending on the calculation load of the instruction. From Robertsson et al. (2006) we have that a response time of 4 ms represent a good trade-off between engineering effort and control safety and performance. Since 4 ms response time results in good performance, a response time of 5-10 ms can be assumed to produce acceptable results as well. However, a higher response time might require the robot speed to be reduced in order to restrict the interaction forces to an acceptable level. Based on this it can be perceived possible to implement a force control structure using the high-level RAPID language, but it might not achieve the process cycle time achievable through direct joint access.

In robot interaction control, situations may arise where information about the current joint torques are of interest. With this information at hand, for instance estimates of the interaction forces between the robot and its environment can be calculated. Unfortunately, these data are not accessible from high-level RAPID instructions.

For control purposes, direct control of the individual joint motor torques would in many cases be very useful. And in most of the force control strategies proposed, including those presented in Chapter 2, this direct access is assumed to be available. Unfortunately, in most industrial robots today this kind of access is not possible (Blomdell et al., 2005). Therefore, in order to implement most force control strategies, a modification is required. This modification can either be done in the robot controller, which has been done in Blomdell et al. (2005) and Robertsson et al. (2006) with promising results, or in the control strategy. To modify the control strategy would involve converting the calculated joint torque into a reference motion, which is possible to execute through RAPID.

When developing programs and applications for ABB robots, there are a few different alternatives to choose from. One option is to use the standard operator interface connected to the IRC5 control module, the FlexPendant. The FlexPendant is designed to be a stand alone unit, with Windows CE operating system and an open system PC architecture. User interfaces can be thus be created using standard development tools, such as Microsoft's C#. Included in the FlexPendant is a RAPID editor, making it possible to develop entire programs for the robot using just this teaching unit. Alternatively the programming can be done in RobotStudio, which contain both a virtual model of the robot and a virtual controller. This way simulations of the robot behavior can be tested, enabling a shorter development process.

Another option is to utilize the Robot Application Builder, with which the user can develop applications in Visual Studio using any .NET language. These applications can be implemented and tested on either a PC, running a virtual IRC5, or the FlexPendant. When implementing on a FlexPendant, the only accepted .NET languages are C# and Visual Basic (ABB Robotics, 2008a).

Chapter 4

Proposed Control System for Anode Milling

Based on the findings in Chapter 2 and 3, this chapter presents a control system design for the task of carbon anode milling. Several different force control strategies were presented in Chapter 2, all of which can be used as sources of inspiration when designing a new control system. Although this control design is especially linked to the anode milling task, it is a general purpose control system and can potentially be used in other interaction tasks as well. The content of this chapter is as follows:

Section 4.1 sums up most of the information available on the task in question. Some requirements and properties are also mentioned here.

In Section 4.2 some assumptions and considerations regarding the task are laid out. This includes what equipment can be assumed available, and what effect this has on the control system design.

In Section 4.3 the actual control system design for the anode milling is presented. This control system consists of a position controller with an outer force control loop, and a velocity controller based on tool power consumption measurements.

4.1 Task Description

The task that this report is concerned with, originates from the aluminum industry. Ever since the beginning, carbon anodes have been used in the production of aluminum. In fact, the process of reducing alumina to aluminum has remained almost the same for over 100 years (Aref and Phillips, 2002). These anodes have an uneven surface of coke, and can even have an uneven surface itself. For the aluminum production it is desirable to get rid of this coke before the anodes enter the production process.

The task of anode milling aims at removing the coke depositions from the carbon anode. Using a milling machine, the coke is to be removed from the anode with a given precision in as short time as possible. The removal process should remove the desired amount of coke, without cutting too deep into the carbon anode itself. Figure 4.1 gives an illustration of the carbon anode with the robot and the milling tool.

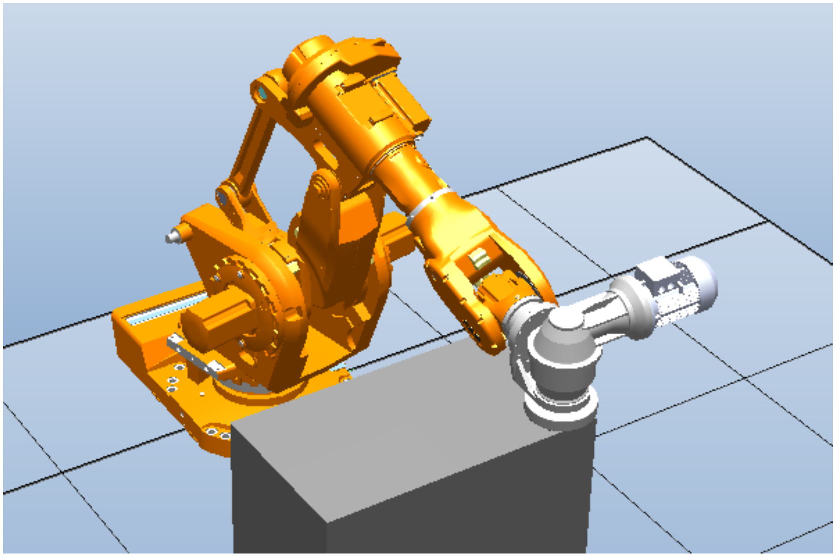

Figure 4.1: Illustration of the anode milling process (Tool model courtesy of RobotNorge AS).

It can often be useful to define a workobject coordinate system, which can then be used in the task specification and robot programming. The workobject in this case is of course the carbon anode, and its coordinate system is defined according to Figure 4.2.

Involved in this milling/grinding operation is usually quite heavy equipment, and therefore the robot manipulator used must be capable of handling heavy loads. For this report it is assumed that a suitable milling tool exists, and that it is possible to mount this on a standard industrial robot. As noted, such milling tools are often quite heavy so the robot must be chosen among the larger models available, for example ABB's IRB6660-205/1.9. This particular robot can handle a tool weighting up to 205 kg, including payload. In addition to this it has a pose accuracy of 0.15 mm and a pose repeatability of 0.07 mm (ABB Robotics, 2006b), where these numbers represent the deviation from the programmed path. All of this makes the IRB6660-205/1.9 well suited for the task in question.

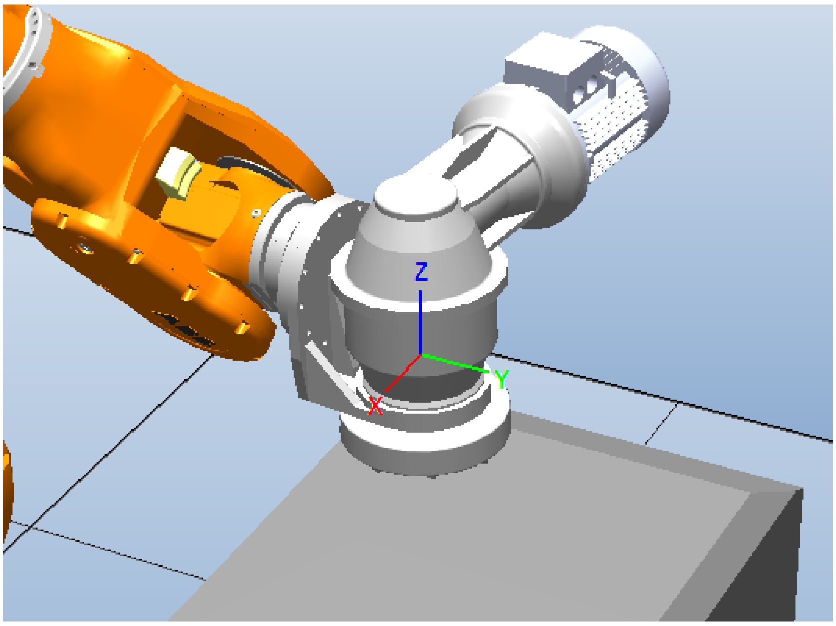

Figure 4.2: Close-up view of the anode milling process, with attached coordinate system (Tool model courtesy of RobotNorge AS).

To obtain an acceptable process time, it may be necessary to involve more than one robot to do the task. However, this report will only consider the case of a single manipulator in the robot cell.

When imposing force control on a manipulator, the choice of control strategy is influenced by the resources available. For instance, if the robot is fitted with a force/torque sensor, a hybrid force/motion control strategy can be employed. Otherwise a different approach has to be taken, for example impedance control. Or if the robot is equipped with a compliant tool, a specially designed control system may not be required at all.

If the robot is equipped with purpose built software solutions, it is possible to implement quite different strategies than is possible without it. Such software solutions include ABB's SoftMove, FC Pressure or FC SpeedChange, all described in Section 3.1. None of this is necessarily available in an industrial robot today, and therefore not considered an option in this report.

4.2 Assumptions and considerations

When designing a control system to enforce force control of a robot, there are a few factors to consider. First we need to know what tools and equipment can be expected to be available. As mentioned above, this knowledge will influence the choice of control design. The robot cell assumed to perform this particular task of anode milling can be expected to have a wrist-mounted milling tool, attached to a suitable robot manipulator. This is of course essential for the task execution. We can expect to be able to measure the tool's power consumption during operation, which can then be used for control purposes.

Since we aim to achieve control of the interaction forces between the robot and the environment, a force sensor could be useful. With a force sensor mounted between the tool and the robot, a value of the contact force would be measured. It is not difficult to understand that this kind of information could be very useful for force control purposes. Unfortunately, this type of equipment is usually quite expensive, and not very common in industrial robots today. As a consequence, this report assumes no force sensor available in the robot cell, and therefore concentrate on the force control strategies that incorporates this. From Chapter 2 we can note that only one of the four strategies presented, namely the hybrid force/motion control, is explicitly reliant on force sensor feedback. The other approaches are also likely to be enhanced through force sensor feedback, but are able to operate without it as well.

A robot operating closely with its environment, can easily damage both itself and its environment without proper control. If the environment has a high degree of stiffness the manipulator must be compliant to avoid injuries, and vice versa. In the case that this compliance is not provided through mechanical parts on the robot, as in passive compliance, a purpose built control system has to. If the environment is stiff, the control system needs to react quickly to avoid excessive contact forces being built up. Otherwise parts might break or other kinds of damage can occur. A quick reaction requires a short response time in the controller. Keeping this is mind, it is clear that a short response time is a vital and challenging part of a force control system. In this case the controller would be the combination of the IRC5 robot controller along with an outer force control system. From Section 3.2 we know that the IRC5 has a response time in the range of 10-100 milliseconds, and that the higher response times results from heavy calculations. Therefore it is necessary to keep the outer control loop as simple as possible, to attain a low response time.

The type of interaction task in question might influence the choice of force control strategy. Force control in robotics usually takes place in either machining or assembly tasks. Machining tasks include material removal and surface polishing processes, while assembly covers tasks like peg-in-hole and different insertion and installation tasks. Because these two categories of tasks have different requirements, they may require different control strategy.

Keeping this in mind can help when deciding on what force control strategy to use.

4.3 Control System Design

The control system required for the task described in Section 4.1 needs to encompass the assumptions and requirements laid out in Section 4.1 and 4.2. First of all, that means to get by without a force/torque sensor. The main reason for this is that force sensors usually come at a high cost, and it is therefore to see if it is possible to design an adequate control system without it. This will inevitably have a large bearing on the force control system, and will be decisive for the eventual choice of control system.

A second important factor is what we want to achieve. We have the robot, the tool and the anode, and in order to design the control system we must know what kind of end result we are looking for. Two things are then important to note: The amount of coke on a given anode is unevenly distributed and the shape of the carbon anode itself cannot be perfectly modeled. Therefore it might be of interest to control both the velocity at which the coke is removed, and the normal contact force between the robot tool and the anode.

In this thesis we propose an extension of the force control approach described in 2.3.1. The control system proposed here is a combination of two different strategies, each with a separate control task. The first strategy is the force control with inner motion control loop, described in Section 2.3.1, while the other strategy is one based on the measurements of the motor current in the the milling tool.

A wrist-mounted milling tool is expected, and access to the tool's power consumption is assumed. By comparing the power consumption during free motion with that of material removal operations, we can estimate a suitable window for operational power consumption. If the measured consumption is outside this window, motion changing routines can be invoked, in order to bring the consumption back to the desired window of operation.

There are several different possibilities on how to change the motion, in order to achieve the desired milling effect. One is to create a rule-based logic controller, with predefined actions for every measured tool power consumption. The benefits of this approach is that it is quite easy to understand, and that the rules needed are usually fairly intuitively to derive. This is a method inspired by the Force Control SpeedChange method described in Section 3.1. The draw-back here is that the controller output is divided into a limited number of discrete steps, which can harm the controller accuracy.

An advantage is that it can be relatively easy to modify, even for a robot operator with little knowledge about the control system.

Another option is to use a type of the much-used PID controller. In this approach the deviations in the measured power consumption from the desired one is multiplied by a proportional gain P, possibly with the addition of integral (I) and\or damping (D) action. This is a well-established concept in control theory, and its behavior and performance are well documented in the literature. Some considerations have to be taken with this approach, including whether to use a P, PI, PD or full PID controller. A block diagram of a full PID controller is shown in Figure 4.3, where the s is the the Laplace transformation, giving s the meaning of derivative and $\frac{1}{s}$ the meaning of integration.

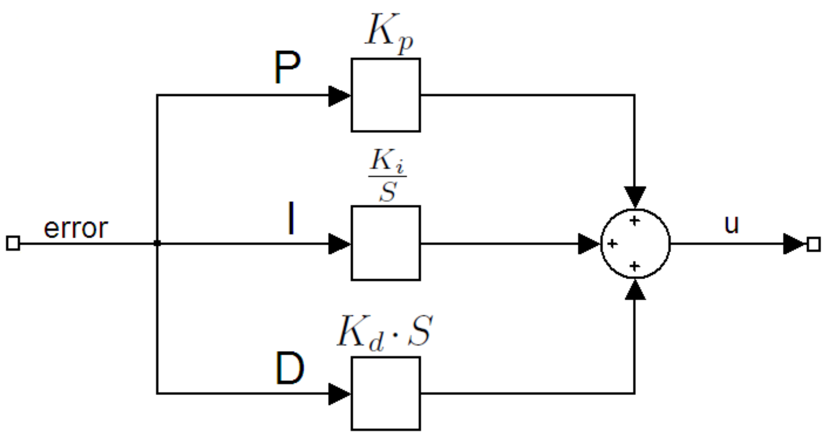

Figure 4.3: Block diagram of PID controller.

As mentioned above, in addition to this tool power consumption control a force control with inner motion loop is to be used. Which control loop to control which variable, position in Z-direction and velocity in X-direction, is not plain to decide. Based on the tool power consumption measurements, one has two choices on how to stay within the desired limits. Either a positional change in the Z-direction can be made, or the robot travel velocity can be altered. Similarly, it is also possible for the other approach to either have an inner position loop or an inner velocity loop, thereby having the possibility to control either the Z-position or the X-direction velocity.

The approach chosen contains a force control with inner position loop controlling the Z-position, and a tool power consumption control loop controlling the robot travel speed (X-direction) velocity. This set-up was decided based on the assumption that the milling process would generate large amounts of noise in the X-direction velocity estimations, thereby making the inner velocity loop difficult to tune correctly. Less noise can be expected in the Z-direction, therefore making a inner position loop controlling the Z-position the natural choice.

For the velocity control in this task, it can be sound to choose a PID controller. A steady state error in the velocity may not have a significant impact on the final result, but it can influence the process cycle time in a negative manner. Therefore integral action in the controller can be useful. It can also be of interest to avoid too abrupt changes in the robot velocity, in order to keep the strain on the mechanical parts low. This can be achieved by adding the dampening action in the controller. Keeping a simpler structure on the controller might ease the implementation, since less parameters would have to be set. But the overall target is to obtain the best possible performance, and therefore a full PID controller is chosen. By including the dampening effect in the PID controller, the system becomes more sensitive to noise. Based on real-world experiments, this inclusion might need to be reconsidered. Because the option of *SpeedRefresh* can be used when setting the speed, we are assured that the target robot speed is within the possible speed range of the robot. Therefore it is not necessary to limit the controller velocity reference output. By giving the tool power consumption loop higher

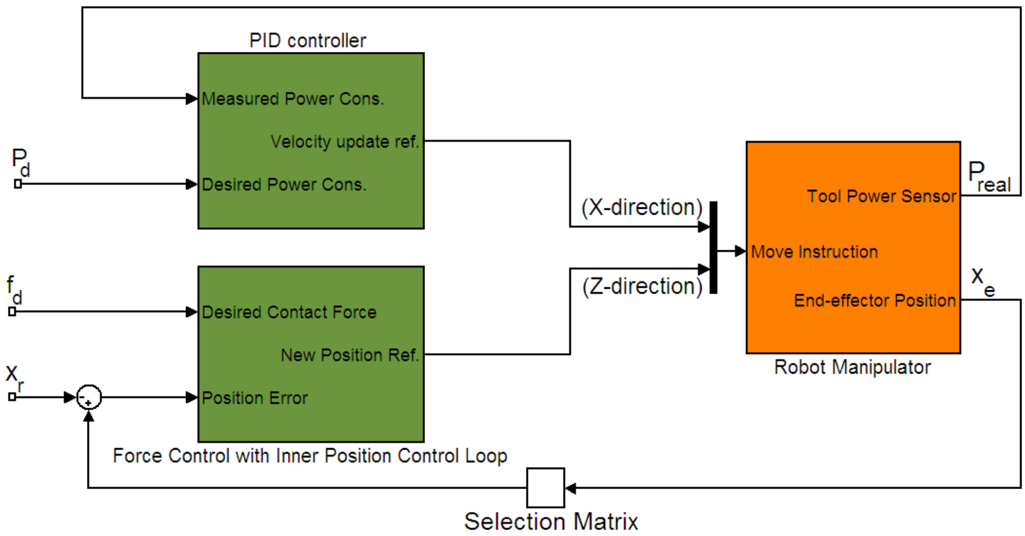

Figure 4.4: Block diagram of proposed Outer Force Control system design.

priority, one can expect a smoother surface with this set-up, given that the velocity can then be changed more often than the position. Changing the robot velocity will not leave noticeable marks on the anode, while changing the Z-position of the end-effector is likely to do so. A block diagram for this combined control system approach, later referred to as *Outer Force Control*, is shown in Figure 4.4.

The two subsystems found in Figure 4.4, are shown in Figure 4.5 and Figure 4.6. As we can see, the *Force Control with Inner Position Control Loop* in Figure 4.5 is the same as the control approach discussed in Section 2.3.1. The PID controller in Figure 4.6 is a standard PID controller, with both integral and derivative action on the error signal, in addition to the proportional action.

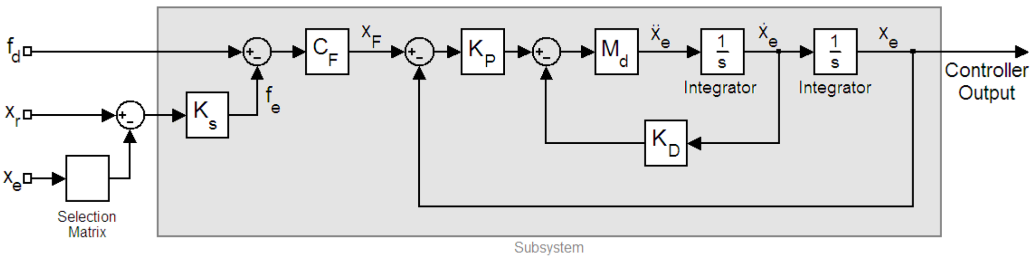

Figure 4.5: Block diagram of subsystem in Force Control with Inner Position Control Loop.

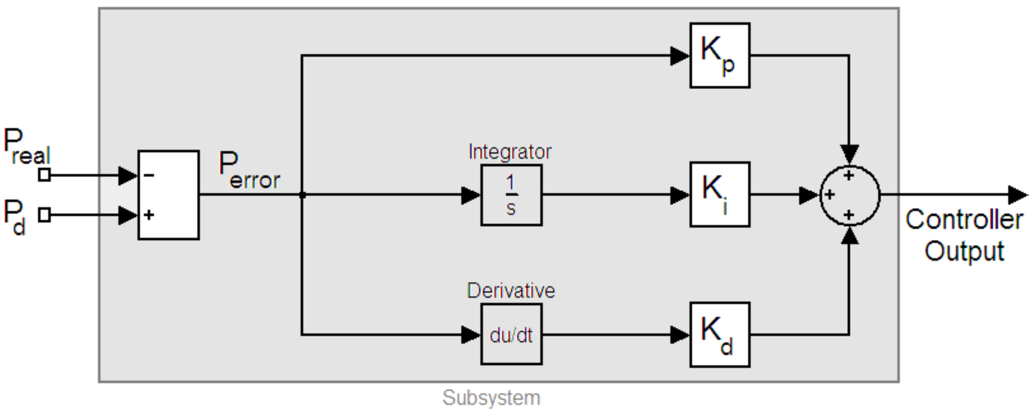

Figure 4.6: Block diagram of subsystem in PID controller.

From comparing Figure 4.4 to 2.4, we can note that the overall control structure suggested here is somewhat inspired by the ideas behind the hybrid force/motion approach. Both control approaches aims to control both the interaction force and the robot motion, by separating the task into independent subspaces.

Chapter 5

Adaptation of Control System Design to Industrial Robot

The control system proposed in Chapter 4 can not be directly implemented on a standard industrial robot of today. This is because there is a gap between the theory developed on force control, and the features available on an industrial robot. In this chapter these differences are examined, and possible solutions are presented. The chapter is divided into four subsections:

Section 5.1 looks at the issue of kinematic and dynamic models, which are assumed known in control theory but usually not available in practice. A solution in this particular case is presented.

Section 5.2 presents the numerical approximations needed in the integration and derivation parts of the control system. Such features are not available in the RAPID robot programming language, and therefore a numerical solution is given.

In Section 5.3 the new, and adapted, control system is presented. Here the considerations given in Section 4.1 and 5.2 are taken into account.

Section 5.4 briefly presents an alternative approach to the force control task in question for this report. This approach is similar to an approach often used in the industry.

5.1 Kinematic and Dynamics Models

A gap exists between the theory on force control and todays industrial robots, because the theory often assumes full knowledge of both kinematic and dynamic robot model. Such models are not supplied by the robot vendors today, and one can therefore not expect to have this information at hand. The kinematic model might be possible to derive from scratch, to some extent, with acceptable accuracy. Dynamic modeling, on the other hand, is a much more complicated affair that would also require experimental identification of the dynamic parameters. For some control schemes this problem can be difficult to circumnavigate, such as impedance control, and they might become difficult to implement in industrial robots. There exists a large amount of literature on how to derive these robot models, for instance Khalil and Dombre (2002), but for this report they will be regarded as unknown.

As a part of the assumption that the dynamic model of the robot is known, is the assumption of full access to the force/torque of the individual robot joints. This full access include setting the force/torque of the joint actuators, and the possibility of real-time readings of the current joint force/torque. Such access is not available in industrial robots today, and therefore alternative approaches must be considered. This issue is also briefly discussed in Section 3.2.

In Section 4.3 a control system was presented for the task at hand. Due to the issues mentioned above, this particular control system is quite difficult to implement in an industrial robot. In the position control, the subsystem 'Force Control with Inner Position Control Loop' in Figure 4.4, knowledge about the robots dynamic model is assumed. This is also apparent from the description of this method in Section 2.3.1. One way to remedy this is of course to derive the dynamic model from scratch, with all the time-consuming challenges this would involve.

Another option is to adapt the control system design to avoid the mentioned obstacles, and at the same time utilizing more of the features available in an industrial robot. One such feature is the presence of advanced motion control, which implies that the inner position control loop in Figure 4.4 is somewhat redundant. By instead converting the estimated contact force error into a positional reference for the robot controller, the IRC5 will take care of the low-level motion action. The new position controller is shown in Figure 5.1.

5.2 Numerical approximations

Advanced calculations like derivatives and integrals are not possible in RAPID, therefore some numerical approximations need to be used instead. This can be done in a number of ways, and some of these are mentioned below.

In its simplest form, the differentiation can be achieved by subtracting the previous error from the current one, and dividing it by the time between them. This simple setup is desirable, and can be assumed to fulfill the

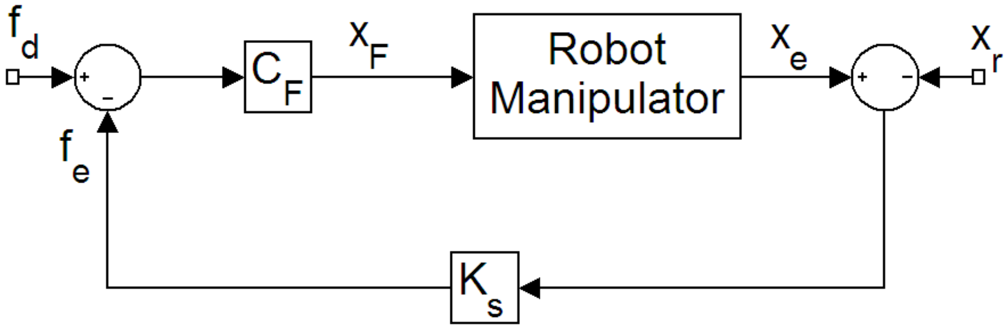

Figure 5.1: Block diagram of adapted force control with inner position control loop.

requirements of the system.

Numerical integration is a bit more difficult, and can be done in a number of ways. The rectangle method and the trapezoidal method are two well-known approximation methods. The method of choice is not likely to have much of an impact on the overall performance here, and therefore the rectangle rule will be used. One problem with integrating action in a controller, is the possibility of integrator wind-up. This is a phenomenon that occurs when the output of the controller is saturated, and the process error is non-zero (Visioli, 2006). This can lead to process overshoots and unwanted transient behavior. Different strategies have been developed to deal with this problem, involving avoiding saturation and putting limits on the integral action amongst others. One approach is to shorten the time frame the error is integrated over, thus reducing the potential overshoot and also the time required to get the controller back to normal operation again. In certain cases this approach might be too simple to be an acceptable solution, but due to its simplicity it is chosen for this report. The time frame T can be chosen by the operator, and later in this report the value $T = 3$ will be used.

5.3 Modified Proposed Control System

An adaptation of the control system proposed in Section 4.3 is necessary, regardless of the minor adjustments discussed in Section 5.2. Combining these adjustments with the adaptations made for Figure 5.1, we end up with the control system depicted in Figure 5.2. It can still be labeled *Outer Force Control*, since we have an outer force control loop giving the input to an inner motion control loop.

As we can see from Figure 5.2, the control system is now designed so

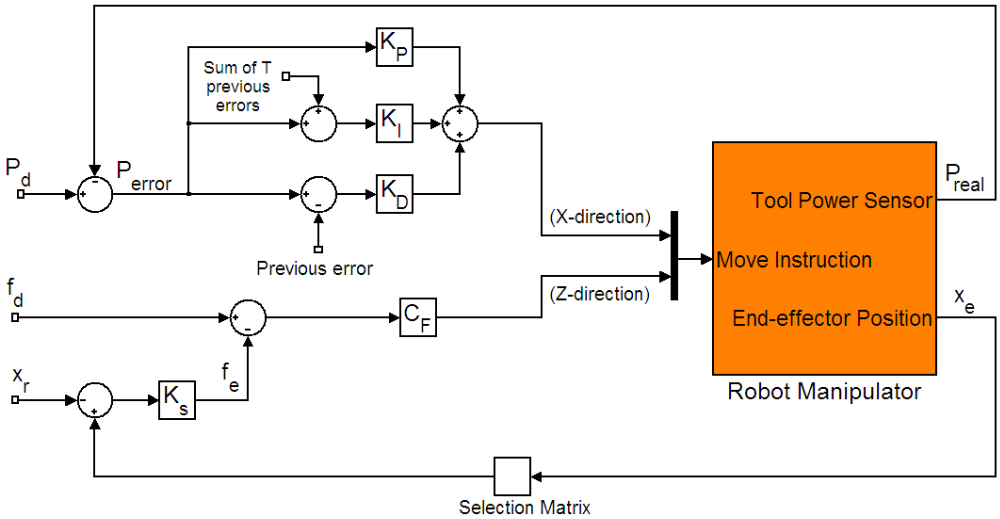

Figure 5.2: Block diagram of adapted Outer Force Control system.

that it does not require accurate information about either the kinematic or the dynamic model. This will be the actual situation in many cases, and is therefore of great interest. Both the tool power consumption and the estimated contact force, arising from environment interaction, combine to create the next move instruction for the robot controller. One problem that is likely to be encountered with this approach, is how to change the robot target position during motion, based on the control system output. For the velocity control part, the speedRefresh option enables the program to change the travel speed of the robot during the execution of a move instruction. Such a feature is not usually available for positional changes.

5.4 Alternative approaches

As a consequence of the adaptation measures discussed in this chapter, it might be the case that the position controller part of the proposed control system is not working as intended. One possibility then is to do a more drastic alteration of the system design, and completely remove the parts affected by the lack of a dynamic model. By doing this, we are left with only the tool power consumption controller, and this might be sufficient for the anode milling process. In its current form, though, it might be unsuitable.

When large amounts of coke needs to be removed, pure velocity control of the X-direction might not be enough. This is due to the fact that even at the slowest robot motion, the milling tool removal rate could be too low. The uncertainty regarding the shape of the anode, can also render pure velocity control unsuitable. Therefore position control of the Z-position of the end-effector can be more important. Switching the PID controller for the tool power consumption from a velocity to a position controller can then be a sound choice. Such a control scheme can be seen in Figure 5.3. The approach

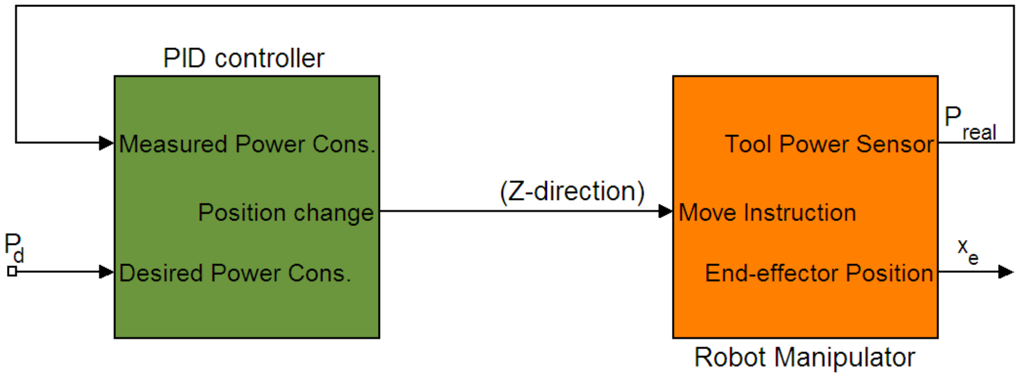

Figure 5.3: Block diagram of PID position control from tool power consumption.

depicted in Figure 5.3 is probably one of the more common control strategies for milling operations in the industry today. Due to its lack of expensive requirements, it is cheap and often reliable choice of control design. The PID control structure is well-known, and this approach is also possible to use in combination with other solutions, such as *SoftMove* described in Section 3.1. It is also likely that with the current setup, the response time of the controller based on cutting tool signals, are shorter. This is because the tool signals can be accessed without going through the IRC5 robot controller, thus avoiding delaying the signal flow.

Chapter 6

Implementation and Simulation of Control System

When designing a new control structure, it is of interest to check the system performance and behavior. There are a few different ways to do this, including mathematical proof, simulations and implementation/testing on a real system. The method of choice for testing and verification of a control system design, depends on the actual case at hand. Sometimes a mathematical proof is a more obvious, or maybe desirable, choice than in other cases. Sometimes testing on a real system is just not possible, given that this requires access to all parts of the system that is to be tested. Simulations of a system and a control system performance can be a useful and cost-efficient solution. During simulations one can observe the system behavior, without the challenges and dangers associated with similar testing on a real system. The control system design suggested in Section 5.3 was decided to be simulated in ABB's *RobotStudio* software.

Section 6.1 contains considerations and choices associated with the simulation of the control system. It also presents assumptions and measures needed to make the simulations possible. What we want to simulate, what we want to compare and how to do this, is also covered in this section.

In Section 6.2 the simulations itself are presented, along with all the system parameters. Graphs depicting the simulation results are also presented and discussed.

6.1 Preliminary Work

In order to perform simulations of a robot, one needs to know, with some accuracy, the behavior of the robot. This information is usually contained in the kinematic and dynamic models of the robot. Such models are not necessarily known for an industrial robot, as discussed in Section 5. As a consequence, accurate simulations can be difficult to carry out. When working with ABB robots, without having the necessary models at hand, simulating in ABB's RobotStudio can be a natural choice. With RobotStudio it is also possible to visualize the robot process, which can provide additional insight into the operation.

For this report, it is of interest to see if the proposed control structure yields a better result than some of the more commonly used control systems in the industry today. In order to do this, one will need to figure out whether to simulate the whole control system, or just parts of it. It is also necessary to know what factors of the two simulated systems' behavior we want to compare.

6.1.1 Comparison Criteria

From the proposed control system in Section 5.3 we have two control loops, one for the velocity and one for the vertical positioning. We want to compare this control system to one preferred in the industry, which is briefly discussed in Section 5.4. To find useful benchmark factors to compare the two approaches on, is not an easy task. For this report the comparison was decided to be based on the following factors:

- Process time
- Robot path
- Response time

where the process time is taken to be the time required by the robot to cover one length of the anode, excluding the approach and retract phase. When comparing the paths chosen by the two control systems, it can be useful to look at potential overshoot, steady state error and oscillations. The response time is the time from a change in controller input to the controller output is at the desired value.

Since the control system proposed in this report aims to control the travel velocity of the robot, comparing the process time can be challenging. As a consequence, the velocity control loop is omitted from the system during simulation. Matching up a system that is position controlled with a system that is position and velocity controlled, will make it difficult to analyze the performance based on the process time. This assumption is believed to be a viable simplification, since the two control loops are independent and completely separated. Controlling the velocity is regarded as a secondary object, aiming to enhance the final result, and will in most cases not be sufficient in itself. What we achieve then is a comparison of two controllers for the vertical positioning of the robot tool, a setup which makes it easier to contrast the the two approaches from one another.

6.1.2 Assumptions

For the simulations carried out in this report, certain assumptions had to be made. First of all a reasonable travel speed of the robot during the milling operation was needed. For a similar project, RobotNorge AS assumed in its task description that the travel velocity of the robot during the milling operation was about 12 meter per minute (RobotNorge AS, 2009). When programming an ABB robot, the travel speed is stated as millimeters per second. So 12 meters per minute equals 200 millimeters per second, labeled *v200*. For the simulation is was decided to set the travel speed a bit lower, to *v150*. In addition to this, for the milling operation the robot target *zone* was set to $z1$ while for the rest of the operation it was set to $z100$. This zone decides how close to the programmed target the robot must be, before it can move on towards the next target.

The second assumption made was regarding the update method for the vertical position target of the robot. As mentioned in Section 5.3, a method for changing the robot target in real-time is not present in ABB robots. Therefore a way around this problem has to be devised. As a solution, the simulated process was divided into two phases. The first phase is the approach/retract phase, where the robot is moving from its home position to the process approach point and from the end of the process back home. During this phase the robot is in normal motion control mode. The second phase is the actual milling phase, where the robot is under force control. To achieve the desired possibility of updating the vertical position target for the robot, the path was divided into small pieces. Starting with robTarget A, the next target, robTarget B, was defined as a small offset from target A. To keep the robot constantly moving across the anode, this offset in the x-direction was set to be 2 millimeters. It is possible to program the whole sequence this way, for instance at the end of the anode set the x-direction offset to zero while increasing that of the y-direction. The vertical position target, the z-direction offset, was set based on the output of the force control system.

In order to compare and contrast the performance of the two chosen control systems, it is necessary that they are applied to the same situation. For the PID controller we need to simulate a tool power consumption signal, while for the *Outer Force Control* approach we are required to simulate a contact force acting on the robot tool. To make the simulation situations as similar as possible, it was assumed that the contact force can be estimated based on the tool power consumption signal. This is an uncertain assumption, and therefore several different connections between the two were tested, both linear and non-linear. Any such connection is likely to be subject to saturation under certain conditions, but in this situation we assume that the values stay within the unsaturated area.

Because the tool power consumption signal is depending only on the time, it will not be altered due to robot motion. In other words, if the robot is moving into the material the signal will not change, and it will always keep the same shape. Through this assumption, it is possible to test the two controllers in the same simulated environment.

6.1.3 Simulation Setup

For the case of *Outer Force Control*, the simulation setup is presented in Figure 6.1. The terms x_{pd} and x_{est} represents estimated positional deviation due to contact forces and estimated end-effector position, respectively. G_L and G_N are gain factors for the linear and non-linear terms P_{real} and P_{real}^2. C_F can be arranged in a number of different ways, as mentioned in Section 2.3.1. For these simulations it was decided to try both a C_F with P and PI controller characteristics.

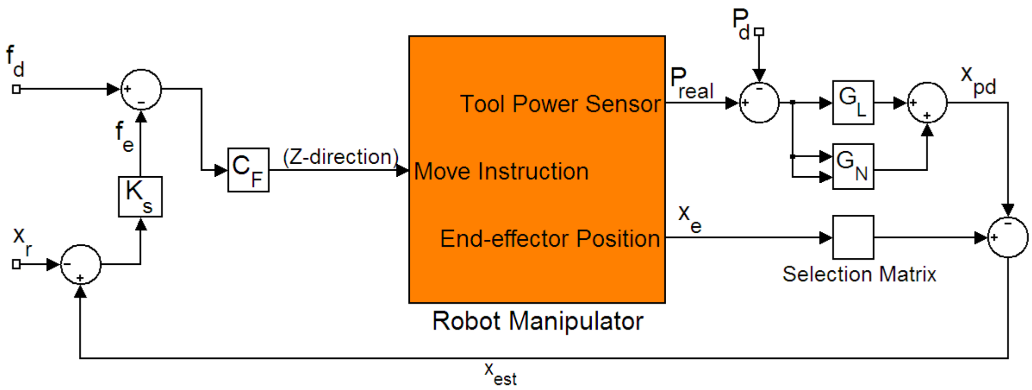

Figure 6.1: Block diagram of simulation setup for Outer Force Control approach.

The assumption that the contact force can be estimated from the power consumption of the tool requires some additional attention. Due to the uncertainty associated with this assumption, some trial and error testing had to be done in order to find a useful connection. This testing was done with a varying values for the *Outer Force Control* parameters, and the resulting robot path was compared to the location of coke on the anode. Three different estimation connections will be presented in Section 6.2, with and without the integral action added to C_F.

The simulation setup used when controlling the robot with a PID controller is presented in Figure 6.2. As mentioned in Section 5.2 the integration and differentiation part of the PID controller had to be numerically approximated, as seen in Figure 6.2. The number of previous errors included in the numerical integration is $T = 3$, and this queue was arranged as a first-in first-out (FIFO) queue.

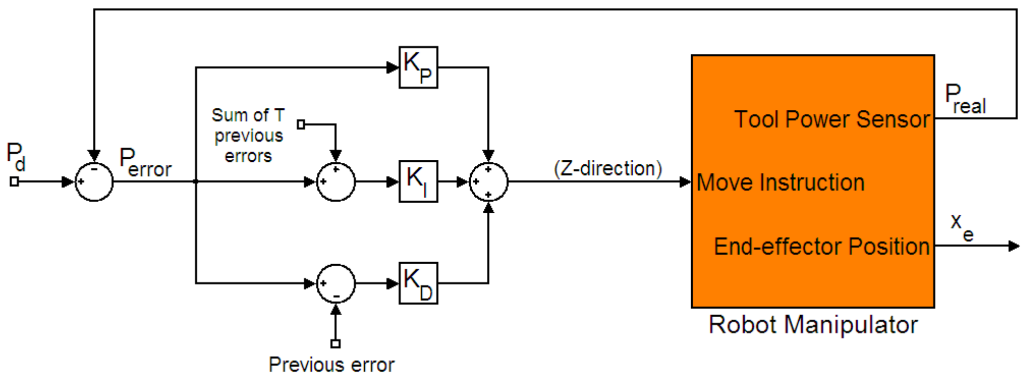

Figure 6.2: Block diagram of simulation setup for PID control approach.

To get the simulations as real as possible, it is necessary to have accurate knowledge about the task. One area of this particular task that one has little available information about, is the amount and distribution of the coke on the anodes. Whether the coke lies as a thin layer all around the anode, or as a number of bumps unevenly distributed, can have an impact on how suited the different approaches are to this task. For the simulations it was assumed that the coke was bunched up, so that when the robot was crossing the anode on the longest side it encountered two large concentrations of coke. The first was made somewhat smaller and the coke increased more gradually, while the other was larger and steeper in its shape. The reason for these differences is to get a better understanding of the robot behavior under different circumstances.

6.2 Simulations in RobotStudio

When performing simulations in RobotStudio, it is necessary to first create a robot station. The station used for this report consisted of an ABB IRB6660 205/1.9 industrial robot, a milling tool and an anode. The tool used is a model of a real milling tool, while the anode has approximately the same dimension as the average carbon anode used in the aluminum industry today.

The output of the two control systems have different format, and this requires some special attention during implementation. The PID controller output is a position update, while the Outer FC output is a z-position target. Therefore the PID controller output could be used as a simple target offset, while the Outer FC output XF had to be incorporated into the previous target, using the same orientation and configuration. Since the force control was only to be enforced during the milling operation, a suitable end criteria was needed. As mentioned in 6.1.2 the robot path is divided into a number of small steps, with a constant step size. During operation the program will iterate through all of these steps, until the end of the anode. Therefore the milling phase was set to last until $iterations \times stepsize$ equaled one anode length, which means that the tool has reached the end of the anode.

The controller parameters used in the simulations, for both the Outer FC controller and the PID controller, are shown in Table 6.1. In the Outer FC we observe that the reference z-position is X_r = -5.0 mm, the desired force is f_d = -2.5 N (where the minus sign indicates into the surface), the environment stiffness K_s was given the experimental value of 0.75. The proportional gain K_F = 0.53 and the integral gain, only used when we want integral action in C_F, K_{int} = 0.02. The PID controller is the benchmark that the Outer FC controller will be tested against, so in all simulations the PID controller parameters were kept the same.

Outer FC		PID	
X_r	-5.0	P_d	4.5
f_d	-2.5	K_P	0.8
K_s	0.75	K_I	0.2
K_F	0.53	K_D	0.2
K_{int}	0.02		

Table 6.1: Controller parameters used in the simulations.

At first a linear connection between P_{real} and the contact force was assumed. By using the values stated in Table 6.1, except K_{int} = 0, both the PID controller and the Outer FC with P control in C_F was simulated. The linear connection used was

$$x_{pd} = G_L(P_{real} - P_d) + G_N(P_{real} - P_d)^2 \tag{6.1}$$

where the linear term $G_L = 60$ and the non-linear term $G_N = 0$.

In Figure 6.3 the resulting vertical robot positions are plotted against each other, along with an approximate profile of the material. In all subsequent graphs the outcome of the PID controlled case will remain the same, since its control parameters are fixed and the connection differences will not affect the PID controller. As we can see in Figure 6.3 the difference in performance is quite notable. It is assumed to be desirable to cut 3-5 mm into the material, and the two bumps of coke were made to be about 24 and 29 mm above the rest of the surface. We see that the PID controller cuts about 4.5 mm into the material when there are small changes in the amount of coke.

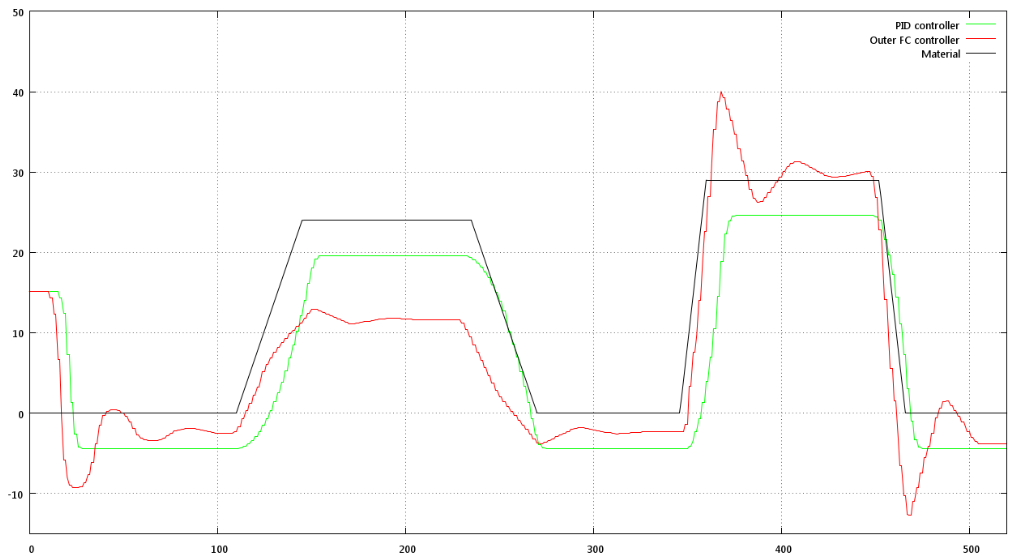

Figure 6.3: Plot of robot z-position with linear connection between P_{real} and the contact force and P controller in C_F, PID controller vs Outer FC controller.

Observe also that after an area of coke, the PID controller is slow to react, and is cutting very little into the material. With the Outer FC the robot is oscillating a bit in the beginning, cutting 9.0 mm into the material at the most, which is much more than wanted. It seems to stabilize itself at around 2.5 mm into the material, before reaching the first area containing coke. During the first interaction with the accumulated coke the Outer FC reacts quicker, but ends up far from the desired cutting depth. The PID controller acts a bit slower both up and down, but in return it achieves a much better cutting depth of around 4.5 mm in steady state. When encountering the largest bump of coke we again see that the Outer FC reacts quicker, but this time it overshoots the coke, and is almost not in contact with the coke at all. On the way down it keeps an acceptable cutting depth, better than the PID controller, but overshoots badly at the end, cutting almost 15 mm into the material. Finally it stabilizes itself at a depth of around 4.0 mm, the same as for the PID controller. The 15 mm cut depth that occurs towards the end of the task is so deep that it would be a cause for concern in a real operation.

By switching from a linear to a non-linear connection, while keeping the P control in C_F, we obtain the results shown in Figure 6.4. This time the connection parameters used were $G_L = 120$ and $G_N = -35$.

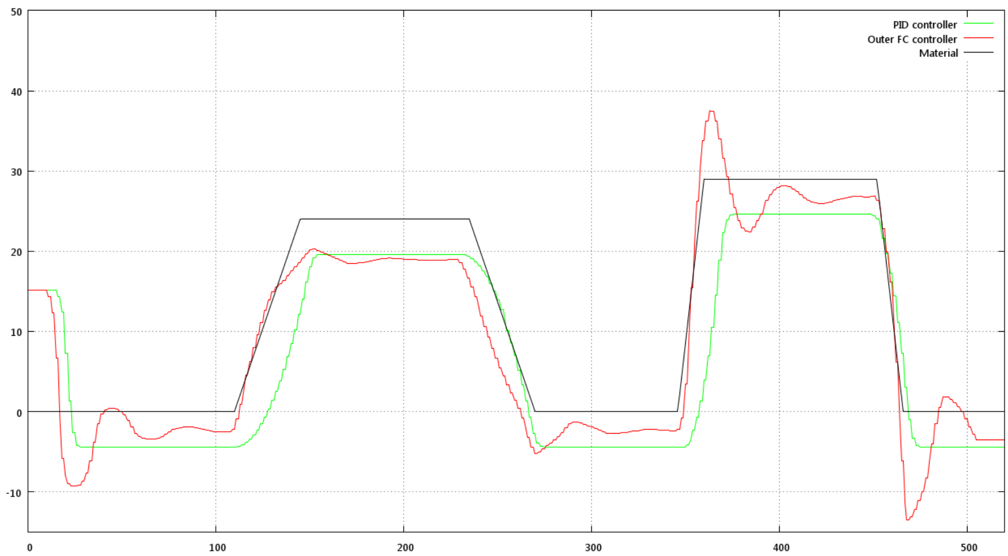

Figure 6.4: Plot of robot z-position with non-linear connection between P_{real} and the contact force and P controller in C_F, PID controller vs Outer FC controller.

Now we can observe that the beginning in Figure 6.4 is quite similar to that of Figure 6.3, with some oscillations for the Outer FC case and a maximum cutting depth of about 9 mm. Again, when encountering the first bump of coke the Outer FC is reacting faster that the PID controlled case, but this time it is stabilized at an acceptable cutting depth. The second coke bump makes the Outer FC react strongly, with an overshoot of about 8 mm which is somewhat less than in the first case. However, this time the cutting depth varies from 7 to 1 mm, which is much better than in Figure

6.3. Towards the end there is a slightly less overshoot than in the previous case, this time about 13 mm deep.

The last connection to be tested was another non-linear one, this time with a slightly greater non-linear term, with its results shown in Figure 6.5. Now the connection parameters were set to be $G_L = 120$ and $G_N = -40$.

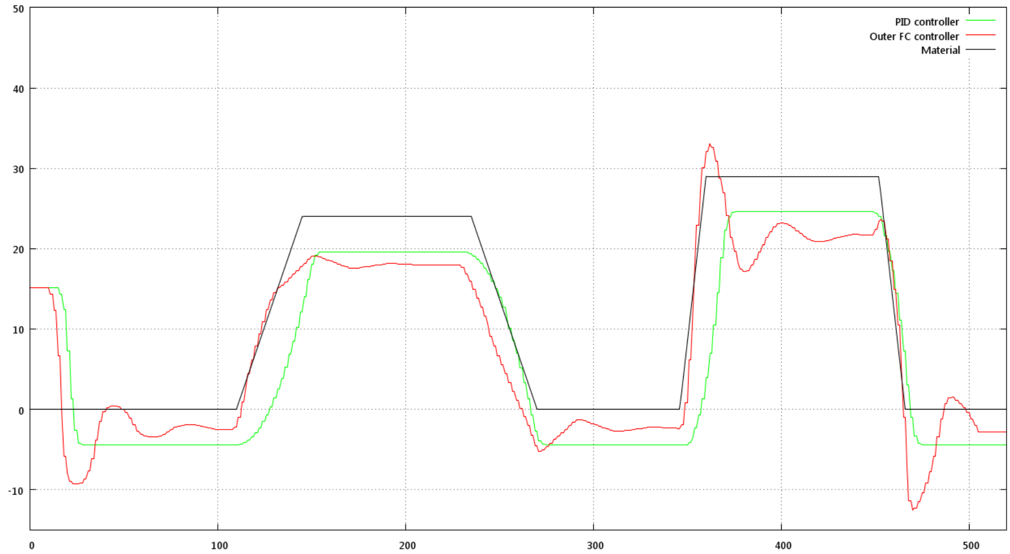

Figure 6.5: Plot of robot z-position with non-linear connection between P_{real} and the contact force and P controller in C_F, PID controller vs Outer FC controller.

The start phase is similar to the previous two cases. The first difference to be noted from Figure 6.5 to Figure 6.4 is that now the robot is cutting a bit too deep into the first coke bump, at around 6 mm. Since the non-linear term G_N is now more negative, it is as expected that the cutting depth will be deeper, relative to that in Figure 6.4. This is also the case for the largest coke bump now, where the overshoot has been reduced to 4 mm but the cut depth is increased to 6-12 mm. It is worth to note that this change in G_N results in a 2 mm jump in cutting depth right towards the end of the bump. This is probably a result of the increased non-linearity of the connection. The depth of the cut after the last bump is now reduced to 12 mm, which is still much more than what is desirable.

Instead of using pure proportional control in C_F in the Outer FC, it can be interesting to see what happens when an integral term is added. First we can consider the case of a linear connection between P_{real} and the contact force, the same as for Figure 6.3, only this time the Outer FC controller has PI control in C_F.

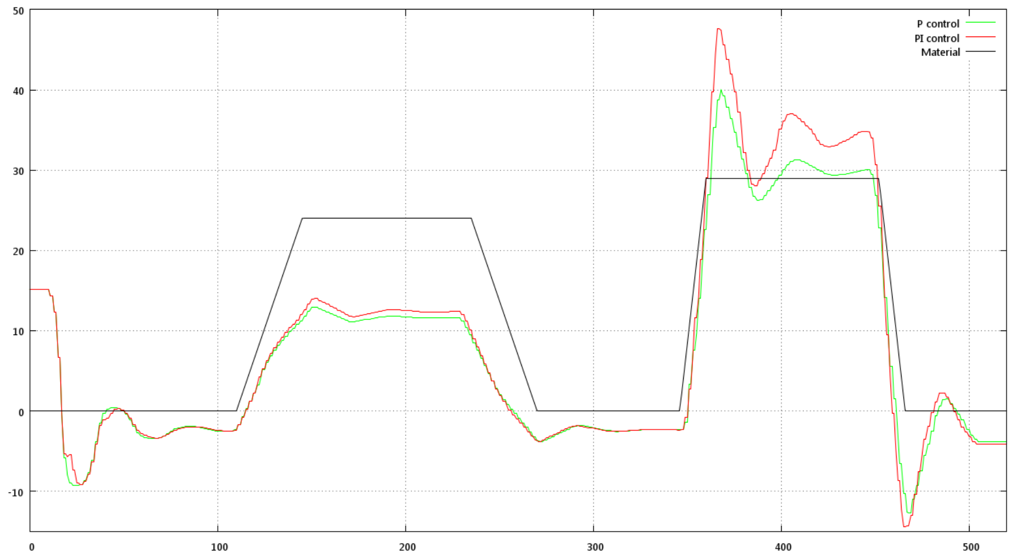

Figure 6.6: Plot of robot z-position with linear connection between P_{real} and the contact force, P vs PI controller in C_F term of Outer FC.

This time the Outer FC controller with PI control action in C_F is plotted against the case of Outer FC with P control in C_F, using the same linear contact force connection, and not the PID controlled case. One small but noticeable difference now is that in the start phase of Figure 6.6 the robot stays a shorter time around the peak values. The oscillations are also smaller, which is a positive sign. When it comes to the first coke area, the cut depth is about 1 mm deeper with proportional control, seen in Figure 6.3, contra PI control. In the area between the two coke bumps there are hardly any differences. With the PI control the Outer FC overshoots the second coke bump by about 18 mm, the highest registered so far. This is so high that during the following oscillations, the robot is hardly in contact with the material at all. The PI control also makes the overshoot after the coke area about 2 mm deeper, to 14 mm cut depth.

Figure 6.7 shows the difference in the vertical position of the robot end-effector using a P and PI control structure in the C_F term, using the same connection parameters of $G_L = 120$ and $G_N = -35$ as for Figure 6.4.

What we observes in the beginning of Figure 6.7 is that same as for Figure 6.6, with some smaller oscillations but generally the same behavior. The reaction time is almost identical when encountering the first part with coke, but the PI controller achieves a more desirable cutting depth of 4 ± 0.5 mm. Then there are no differences at all until reaching the top at the other coke part. Here the PI control overshoots the top by 13 mm, while the P

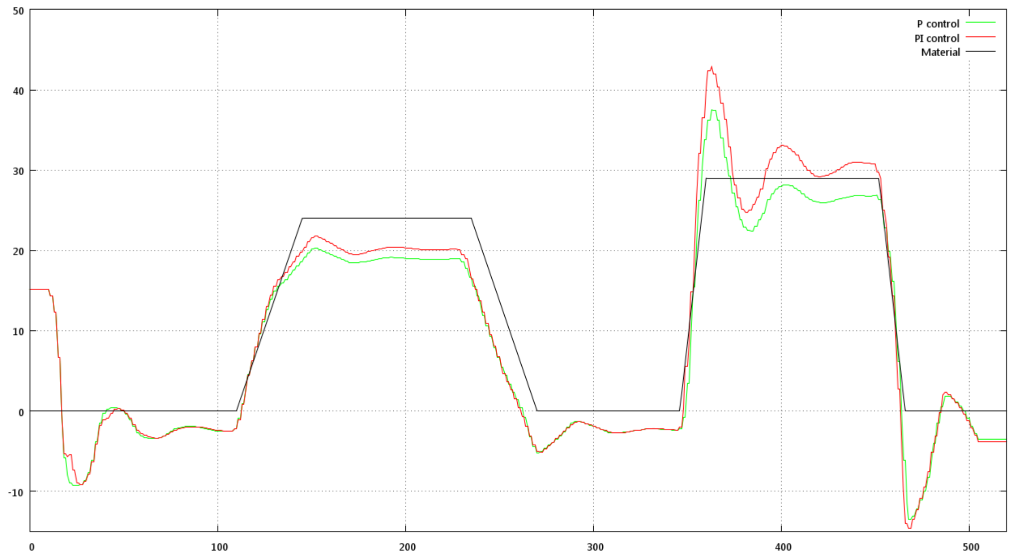

Figure 6.7: Plot of robot z-position with non-linear connection between P_{real} and the contact force, P vs PI controller in C_F term of Outer FC.

control overshoots by 8 mm. The PI controlled robot is only in contact with the material for a short period, thereby not removing the desired amount of coke. On the way down from the second coke bump and the finish, there are no differences in the positioning.

Finally we look at the same situation as in Figure 6.5, with a different non-linear connection of $G_L = 120$ and $G_N = -40$. This is the same connection assumed for Figure 6.5.

As before, we can look at the performance differences when implementing integral effect in the C_F control term for the Outer FC approach. In the start phase of Figure 6.8 we have an identical scenario as in both Figure 6.6 and Figure 6.7. The tendency of a PI control approach that give a lesser cutting depth than the P control continues, while the transient periods are almost identical. For the smallest coke bump the PI approach is cutting about 2 mm shorter into the anode, which is closed to the desired cutting depth. In the case of the larger coke bump, the overshoot of the PI controller is 8 mm, which is smaller than with the previous non-linear connection. For the remainder of the second bump of coke the PI control robot keeps a cutting depth in the range of 2-11 mm, which is somewhat more than wanted. Towards the end we observe that the PI version has a larger overshoot on the way down, but not by much. Both approaches perform well in the finish.

To sum up the simulations above, Table 6.2 presents the key differences between the various experiments, and also the process time of each. As we

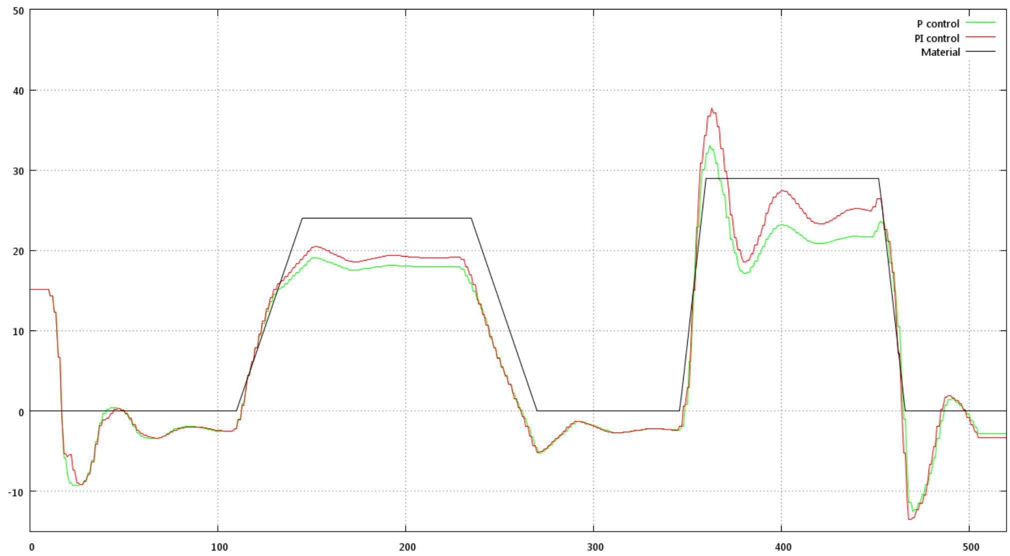

Figure 6.8: Plot of robot z-position with non-linear connection between P_{real} and the contact force, P vs PI controller in C_F term of Outer FC.

Controller	Figure	Connection	C_F	Process Time	% over shortest
PID	6.3-6.5	-	-	14.55	0.00
Outer FC	6.3	linear	P	14.94	2.68
Outer FC	6.4	non-linear	P	15.04	3.37
Outer FC	6.5	non-linear	P	14.99	3.02
Outer FC	6.6	linear	PI	15.06	3.51
Outer FC	6.7	non-linear	PI	15.13	3.99
Outer FC	6.8	non-linear	PI	15.08	3.64

Table 6.2: Overview of the simulations performed.

can see from analyzing the numbers in Table 6.2 the PID controller gives the shortest process time, and a proportional control structure in C_F looks to yield a shorter process time than when integral action is added. These differences are very small though, within 1.5% over the shortest process time. With a maximum difference of a about 4%, it is clear that the total process time will dictate how much emphasis one should put on these results. For a very short overall process time, like here, the difference in time will be small. But on a larger scale, and when time is of the essence, these results can be of more importance.

Chapter 7

Discussion, Conclusions and Further Work

7.1 Discussion

7.1.1 Preliminary work

From Chapter 2 it is clear that there are many different ways to approach the challenge of force control in robotics. Every approach can be classified as either a *passive* or *active* interaction control approach, and the active interaction control approaches will often be preferred due to their versatility. Versatility is important in a field like robotics, and given that the robot itself is designed to be an all-round machine, it will usually be beneficial to have a flexible control system. This is achieved with active interaction control.

In this report, the relationship between ABB robots and force control has been of particular interest. In Section 3.1 we saw that there are several specially designed solutions for force control, all with different areas of applicability. For the task of anode milling discussed in this report, for instance a combination of *SoftMove* and a velocity controller could be very promising. However, all of these solutions come at a cost, which can influence the decision to maybe choose a different option.

An issue discussed in Section 3.2 that makes ABB robots and force control not a perfect match, is the response time of the controller. This is in fact likely to be true for almost any industrial robot today, and not just ABB robots. Most industrial robot controllers are developed around motion control, which has different needs and requirements than force control, and therefore the response time is not favorable for force control. Because of this, research has been done to add some extra hardware to the robot controller, to facilitate force control. This option was not pursued in this report, due to an important initial condition for the work presented in this thesis. The before-mentioned condition was that the control approach proposed should be based on the existing ABB industrial robot controller.

The control approach presented in this report is developed based on a study of existing literature on robot force control, and of ABB force control software and robot controller. The information obtained here gives a good basis to develop a control approach on, since it contains both theoretical and practical aspects.

7.1.2 Control System

In Chapter 4, a control system for the anode milling operation was proposed. This control system is not tied too closely to this specific task, and can be used in different operations as well. One factor that influenced the choice of method, was the equipment assumed available at the robot cell. The most important assumption here was that no force/torque sensor was available, and this obviously puts a number of bearings on the control system development. Another factor was that the controller to be used for the control approach proposed in this thesis, ABB's *IRC5*, does not support direct control of the individual joint actuator torques, which also limits the possibilities.

The proposed control system consists of two separate control loops, one for position and one for velocity control. This structure was chosen because position control is obviously essential for the milling operation, given that the robot must be able to deal with varying amounts of coke, and velocity control was added to increase the total performance of the robot system. This increase will be achieved by reducing the number of positional changes, thus making the end-result smoother. For the position control, the approach presented in Section 2.3.1 was chosen, while the velocity control is dependent on the possibility to measure the motor current in the milling tool. This motor current, or the tool power consumption, is used as a way to determine how much coke the tool is currently removing, and it was decided to use a PID controller to adjust the robot velocity to attain the desired material removal rate. This choice was made based on the PID controllers well-known characteristics and performance, and that there exists useful methods for correct tuning.

Theory developed on force control of robots, usually requires extensive knowledge of the robot and its structural design, such as kinematic and dynamic models. Because this knowledge is usually not provided by the robot vendors, this information will often be unavailable. This situation leaves two options, either to derive the missing models from scratch or adapt the theoretical control system to the real robot. In this report the latter approach was taken and an adaptation of the proposed, strictly theoretical, control system was undertaken. This particular approach was taken because, for this thesis, we were interested in finding an approach to force control that could make use of the possibilities and handle the limitations of an industrial ABB robot manipulator. The lack of information about the robot models is clearly a limitation, hence an adaptation was decided to be the appropriate option.

In addition to the robot models, an adaptation was needed with regards to the derivative and integral action in the PID controller for the robot velocity. Numerical approximations of these mathematical functions are needed, as discussed in Section 5.2, since the ABB robot programming language RAPID does not have built-in functions for differentiation or integration.

The combination of unknown robot models and no force/torque sensor can make robot force control a difficult prospect. In the control system proposed in this report, an estimation of the interaction forces between the robot and the environment constitutes the basis of the control approach. This estimation is not directly affected by the lack of robot models, but would be surplus to requirements if the robot was equipped with a force/torque sensor. In other words, such a sensor could be integrated into the current control system to replace the estimation process. In situations like this, direct feedback will probably always yield better results than feedback of an estimated value.

7.1.3 Simulations

For the simulations of the control system proposed in this thesis it was decided to use ABB's *RobotStudio*, because it is a very useful application when simulating ABB robots. This application contains accurate models of most ABB robots, and it is also possible to add robot tools and environments.

The proposed control system, labeled *Outer FC* in this report, was decided to be compared to a control system based on PID control of the milling tool power consumption, described in Section 5.4. This choice was made because the latter control system is an approach more commonly used in the industry, and it can be useful to see how the newly developed control system performs compared to an industry approach.

For both control systems, it is necessary to be able to update the robot position target during the milling operation. As discussed in Section 3.2, such a feature is not available in the ABB robot programming language RAPID. Therefore an approach consisting of cutting the path, only during the milling operation, into smaller pieces and let the program iterate over each piece was chosen. Then for each piece the program had the opportunity to set a different target for the robot. This method was chosen mainly for its intuitive approach, and for simple implementation. An issue that can occur with this method, is that the robot travel velocity set by the program might not be achieved. This depends on the *zone* set for each target, which in turn decides how close the robot has to be to the current target before moving on to the next. If the robot has to stop at each such intermediate step, it is obvious that the actual robot travel speed will be lower than the programmed one.

We needed a way to simulate the same environment for the two control systems, in order to make it possible to compare their performance. The solution in this report was to assume a connection between the tool power consumption and the interaction forces between the robot and the environment. This is an assumption that carries with it some uncertainty regarding its validity, and therefore a number of different such connections was tested. Simulations using a linear and two non-linear connections are presented in Section 6.2. The reason for testing both linear and non-linear connections was that this is quite a big assumption, and therefore we wanted to test a wide range of options.

From the simulations presented in Section 6.2, we saw that the behavior changed quite a bit when the assumed connection was changed. The shape of the vertical robot path is not affected very much by a change in the connection, but the actual cutting depth into the material varies a lot and the amplitudes of the oscillations are also changed. With the linear connection the Outer FC controller made to robot cut too deep into the smallest coke bump and too little into the large one, both with P and PI control action in C_F.

When the connection is assumed non-linear, the overall performance of the Outer FC is better. Compared to the linear case the robot path is very similar in the beginning, with the big difference occurring over the two bumps of coke. Here the non-linear cases reach more desirable cutting levels. This can be because these non-linear connections are more correct than the linear one, or because the control system is more suited to the signals generated by these non-linear connections.

By adding integral action into the C_F term of the Outer FC, we can expect to achieve less steady state error. What was observed from the simulations was that the general result was a quicker response and a vertical position closer to the desired one. This is because the integral term adds previous errors to the output generation, and is therefore strengthening the reaction.

The process time for each connection, with and without integral control action in C_F, was recorded during the simulation. In Table 6.2 we could see that the industrial PID control approach was the fastest, with all the other simulations following about 2.6-4.0% behind. From the graphs of the simulations in Section 6.2 we can find the probable solution to this. As we can see, with the PID control the robot does not oscillate around the desired value, and therefore spends less time going up and down.

7.2 Conclusions

This thesis proposes an extension to an approach to force control already found in the literature, by adding velocity control through a PID controller. In this case velocity control can be regarded as a secondary object, introduced to enhance the final end-product, and it is not likely to be sufficient on its own. This report also presents the necessary adaptations needed to implement this proposed control system on an industrial robot manipulator.

The choice of control structure was influenced by the conditions imposed on the robot cell to be controlled. It was assumed to consist of an industrial ABB robot with a standard robot controller, and no additional sensors or hardware could be utilized. If a force/torque sensor had been present, a different force control system could have been chosen. In the current situation it is likely that, for instance, a hybrid force/motion control system with a force/torque sensor could produce better results than the ones obtained by the control system proposed here.

By developing a control approach on a platform consisting of both existing literature and practical information about an industrial robot, we end up with an approach that is ready to be implemented on most industrial robots. This is a very useful property, as many other control approaches originating from various research are purely theoretical, and impossible to directly implement on an industrial robot.

In order to make the control approach directly implementable on an industrial robot, certain adaptations had to be made. First of all kinematic and dynamic robot models had to be assumed unavailable, and this was dealt with by using the existing robot motion controller more actively. The effect of this change on the system performance is not known accurately, since no basis of comparison exists. Secondly, numerical approximations of derivatives and integrals of tool power consumption had to be made. These changes can be regarded as minor adjustments, and will probably have little influence on the system performance.

Successful simulations, with subsequent comparisons, of both the control system proposed in this report and an industrial control approach have been conducted. For these simulations, the velocity controller was omitted from the proposed control system, in order to facilitate comparison of process time.

From these simulations it became apparent that the industrial control approach performed very well, and out-performed the newly developed control system proposed in this report. The industry approach was able to operate closer to the desired cutting depth, both in transient phase and steady state. Several factors can contribute to the proposed control system is not reaching the expected level of performance, with the most likely being the assumption of a connection between the tool power consumption and the interaction forces. This is an assumption that can possibly have a great impact on the simulation, due to the uncertainty regarding its validity.

With the linear connection, we observed that the robot was cutting too deep on the small coke bump, and also too deep on the large bump. This can point towards an error in the connection, since this is an unexpected result, and that the assumption of a linear connection is wrong. There were no similar observations regarding the non-linear connections, which can indicate that a connection of this type is a more correct assumption.

The C_F term in the Outer FC approach was tested with both pure proportional (P) and proportional and integral (PI) control structure. With the integral action came less steady state error, and also a shorter response time. This was true for both the linear and non-linear connections. This means that for both the linear and non-linear connection a PI control structure in C_F resulted in better overall performance for the control system.

When comparing the process time for the various control approaches, it became clear that the industry PID approach had the shortest process time. The other approaches had between 2.6 and 4.0% longer process time, which is a low percentage. When the overall process time is short, this difference will hardly be noticeable. But on a larger scale, and when a short process time is essential, these results can be more important.

This industry approach benefits from the fact that it utilizes signals from the cutting tool to control the robot position. In this way it can achieve a shorter response time than is possible with the robot motion controller, which can be vital. This advantage is not included in the simulations, so one can assume that this approach has a greater potential than what is shown in the simulations.

7.3 Further Work

As a continuation on the work done for this report, the control system proposed here could be tested on a real robot system. This way the uncertainty surrounding the assumed connection between the tool motor current and the interaction forces could be eliminated. However, this will require that all the necessary equipment is available.

For this thesis, it was assumed that the dynamic model of the robot was unknown. By obtaining or deriving this model, it is possible to test the performance of the originally proposed control system, without the adaptations needed here. These adaptations are possibly contributing to some of the unwanted behavior of the control system, so removing the adaptation process may improve the control system performance.

For the simulations, a way to update the robot position target was introduced, namely to cut the robot path into small pieces. This is not necessarily the optimal solution to this problem, and a different solution might make force control easier to implement on an industrial robot controller.

Finally, if we add a velocity controller to the PID industrial control approach the milling process might produce a better end-result. This would be because a reduced velocity could in some places replace a positional change, and therefore yield a smoother surface of the finished anode.

Bibliography

ABB Robotics. *Product Specification Controller Software IRC5*, 2004.

ABB Robotics. *Application Manual Force Control for Machining*, 2006a.

ABB Robotics. *Product Specification IRB6660-205/1.9*, 2006b.

ABB Robotics. *Product Specification Robot Application Builder 5.10*, 2008a.

ABB Robotics. *Application Manual SoftMove*, 2008b.

Ahmed Abou-El-Ela and Rolf Isermann. Fine motion control of robot manipulators in deburring applications utilizing cutting tool signals. *Advanced Motion Control, 4th International Workshop, vol 1*, pages 86–91, 1996.

Hassan Aref and James W. Phillips, editors. *Mechanics for a New Millennium.* Kluwer Academic Publishers, 2002.

Jan C. Aurich and David Dornfeld, editors. *Burrs - Analysis, Control and Removal.* Springer Berlin Heidelberg, 2010.

A. Blomdell, G. Bolmsjo, and T. Brogardh et. al. Extending an industrial robot controller: implementation and applications of a fast open sensor interface. *Robotics and Automation Magazine, IEEE, Vol. 12, No. 3*, pages 85–94, 2005.

Neville Hogan. Impedance control of industrial robots. *Robotics and Computer-Integrated Manufacturing, Vol. 1, No. 1*, pages 97–113, 1984.

H. Kazerooni, J.J. Bausch, and B.M. Kramer. Automated deburring by robot manipulators. *American Control Conference*, pages 1749–1755, 1986.

Wisama Khalil and Etienne Dombre. *Modeling, Identification and Control of Robots.* Hermes Penton Ltd., 2002.

Simeon P. Patarinski and Roumen G. Botev. Robot force control: A review. *Mechatronics Vol. 3, No. 4*, pages 377–398, 1993.

M.H. Raibert and J.J. Craig. Hybrid force/position control of manipulators. *ASME J. Dyn. Syst. Meas. Control 102*, pages 126–133, 1981.

A. Robertsson, T. Olsson, and R. Johansson et. al. Implementation of industrial robot force control case study: High power stub grinding and deburring. *Intelligent Robots and Systems, IEEE/RSJ Int. Conf.*, pages 2743–2748, 2006.

RobotNorge AS. System description - removing coke from anode. (In Norwegian), 2009.

Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer-Verlag Berlin Heidelberg, 2008.

Bruno Siciliano and Kimon P. Valavanis, editors. *Control Problems in Robotics and Automation*. Springer London, 1998.

Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Modelling, Planning and Control*. Springer London, 2009.

Antonio Visioli, editor. *Practical PID control*. Springer-Verlag London, 2006.

T. Yoshikawa. Force control of robot manipulators. *Int. Conf. on Robotics and Automation, Vol. 1*, pages 220–226, 2000.