

Discrete-time sliding mode control of the RRR-robot

Citation for published version (APA):

van der Zalm, G. M. (2003). *Discrete-time sliding mode control of the RRR-robot*. (DCT rapporten; Vol. 2003.082). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2003

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Discrete-time Sliding Mode Control of the RRR-robot

G.M. van der Zalm

DCT – 2003.82

Supervision

**M.Sc. Dragan Kostic
dr. ir. Bram de Jager**

**Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology**



Contents

1	Introduction	5
2	Discussion of the control algorithm	7
2.1	System definition in discrete time	7
2.2	Control strategy	7
2.2.1	Definition of the sliding manifold	7
2.2.2	Control law	8
2.2.3	Design of sliding manifold	8
2.3	Reaching law	9
2.4	Simulation	9
2.4.1	No disturbance	9
2.4.2	Disturbance	10
3	Implementation on the RRR-robot	11
3.1	Reference trajectory	11
3.2	Identification	11
3.3	Modification of the control law	12
3.4	Observer design	13
3.5	Tuning the controller parameters	13
3.5.1	Choice of α	13
3.5.2	Choice of σ , q and r	15
4	Results	17
4.1	Results of the sliding mode controller	17
4.1.1	Discussion	19
4.2	Comparison with H_∞ controller	20
5	Conclusions & Recommendations	23
A	Coordinate transformation	25
B	MATLAB-algorithms	27
C	Transfer function Kalman filter with controller	29

Chapter 1

Introduction

The theory of Variable structure systems (VSS) with sliding mode control (SMC) has been developed for the last five decades [1-3]. When designed in continuous-time domain, SMC systems can feature some distinguished characteristics: invariance to parameter uncertainties and external disturbances, system order reduction, predictable transient behaviour, etc. One of the underlying assumptions in the design and analysis of VSS with SMC is that the control can be switched from one value to another infinitely fast. In practical systems, however, it is impossible to achieve infinite frequency of switching. This is especially impossible with discrete time implementation of SMC algorithms, when finite time delays for control computation are present. Without switching control at infinite rate, the distinguished features of SMC systems may not be feasible in practice. Additionally, chattering may occur in the sliding and steady-state modes of SMC systems. In the steady-state, chattering appears as a high-frequency oscillation about the desired equilibrium point and may also serve as a source to excite unmodelled high-frequency dynamics of the system. To deal with problems caused by discrete-time implementation of SMC algorithms, several conceptual and methodological solutions have been proposed. The most important conceptual contribution is the development of discrete-time SMC algorithms [4,5] that directly take into account effects of discretization. Methodological solutions are directed at eliminating or reducing effects of chattering [1-3]. A discrete time SMC algorithm, considered in [6-8], utilizes both conceptual and methodological advances of SMC theory. It ensures that the sliding mode is reached in finite time without negative chattering effects. Additionally, above mentioned distinguished characteristics of VSS with SMC are preserved - the algorithm is proven to be robust against parameter uncertainties, as well as external disturbances, it enables nice control over transient behaviour of the controlled system, and the system order is reduced in sliding mode. The exceptional property of the considered SMC algorithm is simplicity of its feedback control law, which can be implemented on-line with a minimum computational effort. In this project, effects of the discrete-time SMC algorithm [6-8] in the robot motion control problem should be investigated. The RRR robot [9-10], installed in the Dynamics and Control Laboratory at the Department of Mechanical Engineering, is the experimental set-up used for this investigation. Because of the interactions between the three joints, the RRR robot is quite difficult for accurate and robust control. To facilitate model-based control, a nonlinear robot dynamic model is derived in closed-form and its parameters are experimentally estimated with sufficient accuracy [11,12]. The model is implemented in Matlab/Simulink for real-time application [13]. Enhanced performance of motion control using the nonlinear dynamic model in combination with a linear H_∞ feedback law has been recognized and reported in [14] already. Drawback of the H_∞ feedback is the relative high-order of the controller, which may look less appealing for use in practice. Purpose of this project is to investigate whether the discrete-time SMC law can provide similar performance as the H_∞ feedback but with less on-line computational effort.

This report presents the results obtained after six weeks work on the project. It is organized as follows. First, the discrete-time SMC algorithm is worked out theoretically, and verified in simulations. Then, identification of the RRR robot dynamics, that are not compensated with the dynamic model, is done. The discrete-time SMC law is designed upon the identified dynamics. The controller is tested in experiments on the RRR-robot, and compared with the results of the H_∞ controller. Finally, conclusions are drawn and recommendation for further research are given.

Chapter 2

Discussion of the control algorithm

The control algorithm is discussed using the example in [8], in which a DC motor is considered as the controlled plant. First, the system will be defined in discrete time. Then, the calculation of the sliding manifold coefficients will be done and the control law will be defined. Finally, simulations are performed to validate the controller.

2.1 System definition in discrete time

The considered plant is a DC motor, which can be described by the following continuous-time model:

$$\left. \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -16x_2 - 680u \end{aligned} \right\} \rightarrow \dot{x} = Ax + bu \quad (2.1)$$

where $x_1 = \theta_d - \theta$, with θ the angular position of the shaft, θ_d the desired constant angle, $x_2 = -\omega$, with ω the rotor velocity, $x = [x_1 \ x_2]^T$, and u the control signal. The discrete model of the system (2.1) can be calculated with [8]:

$$\delta x(kT) = A_\delta(T)x(kT) + b_\delta(T)u(kT), \quad (2.2)$$

where

$$A_\delta(T) = \frac{e^{AT} - I_n}{T}, \quad b_\delta(T) = \frac{1}{T} \int_0^T e^{A\tau} b d\tau \quad (2.3)$$

In this discrete-time representation, $\delta x(kT)$ stands for forward difference, i.e. $\delta x(kT) = \frac{x((k+1)T) - x(kT)}{T}$, and e^{AT} is the fundamental matrix of the system. In the following, the notation (k) stands for (kT) .

2.2 Control strategy

The control strategy is such that the state is driven to a so-called sliding manifold from any initial condition in finite time, and is forced to stay on it. The state x is driven to the sliding manifold in the reaching mode. If the state remains on the sliding manifold, the system is said to be in sliding mode. A desired sliding mode dynamics can be achieved by appropriate design of the sliding manifold. On this manifold, the system dynamics are of lower order, and the sliding mode is very robust with respect to modeling error and disturbances.

2.2.1 Definition of the sliding manifold

Let

$$s = c_\delta(T)x, \text{ where } c_\delta(T) \in \mathbb{R}^{1 \times n}. \quad (2.4)$$

The sliding manifold is then defined as $s = 0$. With the assumption $c_\delta(T)b_\delta(T) = 1$, the relative degree of s with respect to the control signal u is one.

2.2.2 Control law

The reaching law is defined as

$$\delta s = -\Phi(s(k), X(k)) \quad (2.5)$$

where the function $\Phi(s, X)$ will be defined later on, and

$$\begin{aligned} \delta s(k) &= \frac{s(k+1) - s(k)}{T} = c_\delta \delta x(k) \\ X(k) &= \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} = \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix}. \end{aligned} \quad (2.6)$$

By substituting (2.2) into (2.6), and solving for $u(k)$, one obtains the control law

$$u(k) = -c_\delta(T)A_\delta(T)x(k) - \Phi(s(k), X(k)). \quad (2.7)$$

The function $\Phi(s, X)$ will be chosen such that the control law has two modes: a non-linear and a linear one. The linear mode acts in a vicinity of the sliding manifold. That vicinity, denoted by $S(T)$, is defined by:

$$S(T) = \{X \in \mathbb{R}^{2n} : s = |c_\delta(T)x| < \varepsilon T + T\eta_1 \|x\|_1 + T\eta_2 \|\hat{x}\|_1\}, \quad \varepsilon > 0, \quad \eta_1, \eta_2 \geq 0. \quad (2.8)$$

Here, $\|x\|_1$ denotes the 1-norm, i.e. $\|x\|_1 = \sum_{i=1}^n |x_i|$.

2.2.3 Design of sliding manifold

The dynamics of the system on the sliding manifold are governed by the equation $s = c_\delta x = 0$. This means that a desired behavior on the manifold can be obtained by calculating appropriate values for c_δ . In the following, the algorithm for this calculation will be discussed. First, the system (2.2) with the control law (2.7) is transformed into the following regular form:

$$\begin{aligned} \delta \tilde{x}(k) &= \begin{bmatrix} \frac{0_{(n-2) \times 1} \mid I_{n-2}}{-c_1(T)} \end{bmatrix} \tilde{x}(k) + \begin{bmatrix} \frac{0_{(n-2) \times 1}}{1} \end{bmatrix} s(k) = A_{11}(T)\tilde{x}(k) + a_{12}s(k) \\ \delta s(k) &= -\Phi(s(k), X(k)), \end{aligned} \quad (2.9)$$

by the coordinate transformation

$$x = P_1(T)P_2(T) \begin{bmatrix} \tilde{x} \\ s \end{bmatrix}, \quad \tilde{x} \in \mathbb{R}^{n-1}. \quad (2.10)$$

Here, the matrices P_1 and P_2 are given by equations (A.2-A.4), see appendix A. c_δ has to be designed in such a way, that the zero dynamics, i.e. the behavior of $x(k)$ when $s \equiv 0$, is such that x converges to zero. For $s \equiv 0$, the system (2.9) is described by the relation

$$\delta \tilde{x}(k) = A_{11}(T)\tilde{x}(k) \quad (2.11)$$

The eigenvalues of A_{11} are only determined by $c_1(T)$, and are free to choose. The desired eigenvalues δ_i are chosen as

$$\delta_i(T) = \frac{e^{-\alpha_i T - 1}}{T}, \quad \alpha_i > 0, \quad (2.12)$$

Then the elements of $c_1(T)$ are determined by

$$c_i(T) = \frac{1}{(i-1)!} \frac{d^{i-1} \prod_{j=1}^{n-1} (\delta - \delta_j(T))}{d\delta^{i-1}} \Big|_{\delta=0}. \quad (2.13)$$

Finally, the vector c_δ defining the sliding manifold is given by

$$c_\delta(T) = [c_1(T) \mid 1] P_1^{-1}(T). \quad (2.14)$$

2.3 Reaching law

Consider the perturbed system

$$\delta x(k) = A_\delta(T)x(k) + \Delta A_\delta(T)x(k) + b_\delta(T)u(k) + d_\delta(T)f(k), \quad (2.15)$$

where $\Delta A_\delta(T) \in \mathbb{R}^{n \times n}$ is a matrix of uncertainties, $d_\delta(T) \in \mathbb{R}^{n \times 1}$, and $f(k)$ a bounded external disturbance with $|f(k)| \leq \mu \forall k \in N_0$.

Assume that the matching conditions are given by:

$$\begin{aligned} \Delta A_\delta(T) &= b_\delta(T)d, \quad d = [d_1 d_2 \dots d_n], \quad |d_i| \leq d_m < \infty, \quad i = 1, 2, \dots, n, \\ d_\delta(T) &= b_\delta(T). \end{aligned} \quad (2.16)$$

It is proven in [8], that the following conditions on Φ are sufficient for x to reach the sliding manifold in finite time and stay on it for each $x_0 \in \mathbb{R}^n$:

$$\begin{aligned} (1) \quad & \Phi(s, X) = \frac{s}{T}, \quad X \in S(T), \\ (2) \quad & \frac{\gamma T(d_m \|x\|_1 + \mu)}{|s|} < |s| < \frac{T\Phi(s, X)}{s} < 1, \quad X \notin S(T), \gamma > 1, \\ (3) \quad & \varepsilon > \mu, \eta_2 > d_m. \end{aligned} \quad (2.17)$$

The first condition constitutes the linear mode of the control law, where the second and third condition are sufficient for s to reach S in finite time, and stay in it. Since the mapping Φ is a continuous one in the vicinity S , the control law is also a continuous function, which results in a chattering free system.

If the reaching function is chosen as

$$\Phi(s, X) = \min\left(\frac{|s|}{T}, \sigma + q|s| + r\|X\|_1 \operatorname{sgn}(s)\right), \quad 0 \leq qT < 1, \quad r \geq d_m \gamma, \quad \sigma > \gamma \mu, \quad (2.18)$$

the conditions (2.17) are satisfied. With this reaching function, the vicinity S is defined by:

$$S(T) = \left\{ X \in \mathbb{R}^{2n} : s = |c_\delta(T)x| < \frac{\sigma T + rT\|X\|_1}{1 - qT} \right\}. \quad (2.19)$$

If no disturbances are present, s equals zero one time step after it enters S , since the equation for δs becomes:

$$\delta s(k) = \frac{s(k+1) - s(k)}{T} = -\Phi = -\frac{s(k)}{T} \rightarrow s(k+1) = 0. \quad (2.20)$$

The second and third condition of (2.17) guarantee that s stays in S , in face of the presence of disturbances.

2.4 Simulation

The control algorithm is validated by simulations in Simulink. First, a simulation without disturbance is done. Then, the robustness properties of the controller are investigated by performing simulations with a perturbed model.

2.4.1 No disturbance

The numerical values for the discrete-time system matrices with $T = 0.4$ ms are

$$A_\delta(T) = \begin{bmatrix} 0 & 0.9968 \\ 0 & -15.9489 \end{bmatrix}, \quad b_\delta(T) = \begin{bmatrix} -0.1357 \\ -677.8286 \end{bmatrix} \quad (2.21)$$

The desired eigenvalue of A_{11} is chosen as $\delta_i = -15$. Then, the sliding manifold coefficients c_δ as calculated by the MATLAB-routine `calc_cd`, see appendix B, are:

$$c_\delta = \begin{bmatrix} -0.0221 & -0.0015 \end{bmatrix} \quad (2.22)$$

Simulation results with $\sigma = 8$, $q = 0$ and $r = 0$ are shown in figure 2.1. It can be concluded that the sliding manifold is reached in finite time, and that the control input is chattering free.

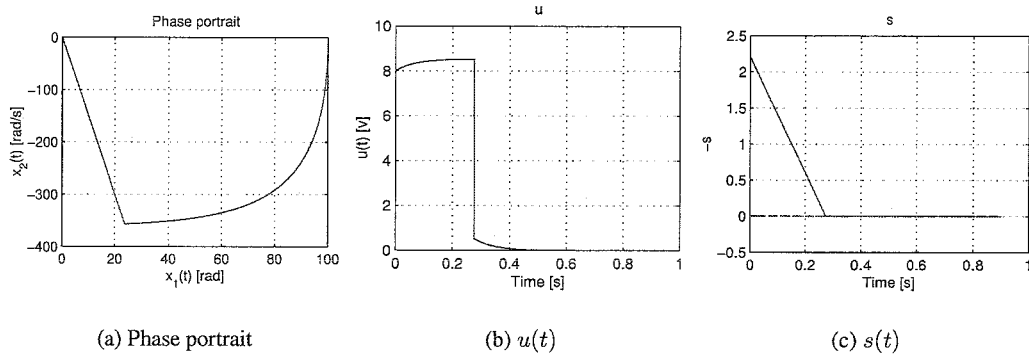


Figure 2.1: Simulation with $\sigma = 8, q = r = 0$ and initial conditions $x_0 = [100 \ 0]^T$

2.4.2 Disturbance

The perturbed system (2.15) is used, with the disturbance being a normally distributed white noise f . Taken into account the matching conditions (2.16), the model becomes:

$$\delta x(k) = A_\delta(T)x(k) + b_\delta(T)(u(k) + f(k) + dx(k)), \quad |d_i| \leq d_m < \infty, \quad |f(k)| \leq \mu. \quad (2.23)$$

The noise amplitude $\mu = 0.7$ and the upperbound for the vector d is $d_m = 0.01$. Then, according to equation 2.18, $r \geq d_m \gamma$ and $\gamma > 1$, r is set to 0.011. The influence of sampling time T on the simulation results for s and θ is shown in figure 2.2. It is clear that a smaller sampling time improves the performance

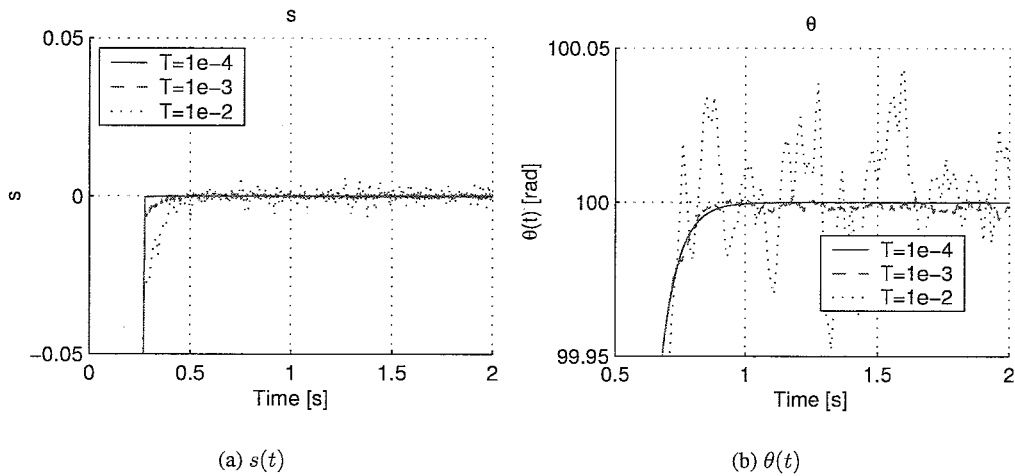


Figure 2.2: Influence of sampling time T on s and θ

of the controller.

Chapter 3

Implementation on the RRR-robot

3.1 Reference trajectory

The goal of this project was to confront performance of the discrete-time SMC with the results shown in [14] and [15]. For a good comparison, the same reference trajectory is used, a so-called letter-writing trajectory. The robot has to perform the writing task shown in figure 3.1. The reference trajectory is such that this trajectory is followed in 10 seconds.

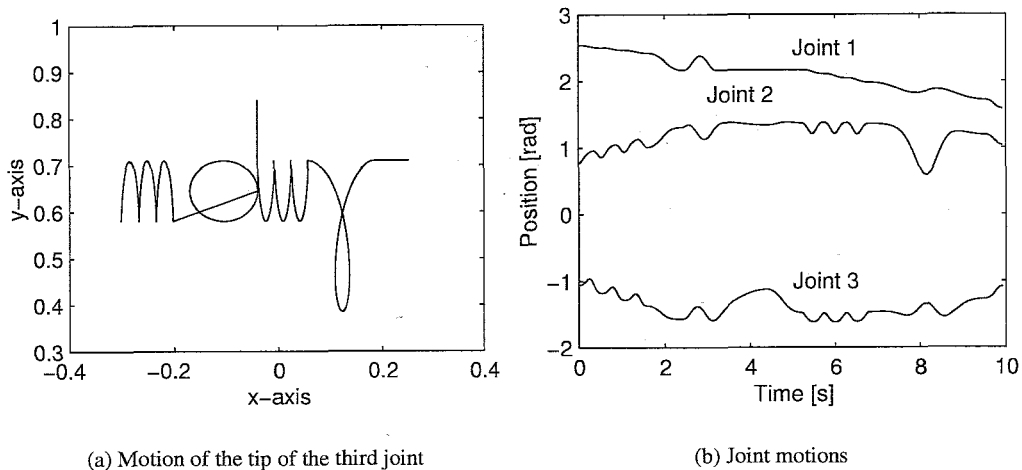


Figure 3.1: Writing task

3.2 Identification

A decoupling input-output linearizing controller is already implemented in the experimental setup [14], which yields for each joint the following input-output relation:

$$J\ddot{q} = u, \quad (3.1)$$

with \ddot{q} the acceleration of the joint, J a constant equal to 1, and u the control input. To check whether this model is valid, an experiment is done. In this experiment, the relevant joint is moving with constant speed,

and white noise f is added to the control input u . The other two joints are held at a fixed position. A weak PD-controller is used to control the motion of the robot. Then, the process sensitivity can be computed by:

$$S = \frac{1}{1 + PC} = \frac{f}{u}, \quad (3.2)$$

with P the plant, and C the controller. From the sensitivity, the plant dynamics can be computed, since the controller has a known transfer function. Results of this procedure for each joint are shown in figure 3.2. The estimated model in this figure is the model of equation (3.1). For joint 1, this model approximates

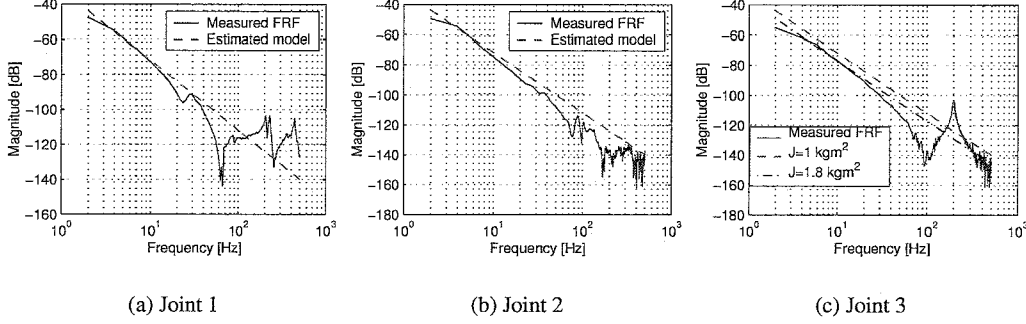


Figure 3.2: Frequency response function measurements

the measured frequency response rather well for frequencies lower than 20 Hz. At this frequency, a small anti-resonance occurs. For joint 2, the estimated model is approximately correct for frequencies lower than 30 Hz. The estimated model for joint 3 deviates more than for joint 1 and 2. Apparently, an error is made in the part of the controller that feedback-linearizes the robot dynamics of joint 3. No attempt to correct the feedback-linearizing controller is made, but the model mismatch is solved by setting the constant $J_3 = 1.8$. Further, a large resonance peak occurs at 200 Hz.

3.3 Modification of the control law

To improve the tracking performance if the robot has to track a reference trajectory, a modification of the control law (2.7) is done. Assume that the reference trajectory q_{ref} can be realized, i.e. $\exists u_{ref} = u_{ref}(kt)$ (bounded), $\exists q_{ref}(0)$ so that for all $kt > 0$

$$\delta x_{ref}(k) = A_\delta x_{ref}(k) + b_\delta u_{ref}(k) \quad (3.3)$$

Define

$$e(k) = q(k) - q_{ref}(k), \quad \dot{e}(k) = \dot{q}(k) - \dot{q}_{ref}(k), \quad (3.4)$$

and the state $x(k) = [e(k) \ \dot{e}(k)]^T$, then

$$\delta x(k) = A_\delta x(k) + b_\delta (u(k) - u_{ref}(k)) \quad (3.5)$$

The control law (2.7) is modified in the following way:

$$u(k) = u_{ref}(k) - c_\delta(T)A_\delta(T)x(k) - \Phi(s(k), E(k)), \quad (3.6)$$

with $E(k) = [e(k) \ e(k-1)]^T$. This control law includes the feedforward component u_{ref} , which, according to (3.1), equals $J\ddot{q}_{ref}$.

3.4 Observer design

The performance of a sliding mode controller depends strongly on the quality of the measurements of the available states. The presence of noise can cause the state to jump from one side of the sliding manifold to the other, which leads to chattering. Since noise-free measurements are impossible in practice, an observer is designed, which reconstructs the state from the measurement. The observer is designed using the model (3.1), with the uncertainty on the system equations and the measurement taken into account:

$$\begin{aligned}\ddot{e}(t) &= u(t) + v(t) \\ \dot{v}(t) &= w(t) \\ \tilde{y}(t) &= e(t) + \eta(t)\end{aligned}\quad (3.7)$$

Here, $v(t)$ represents the model error, which is modeled as white noise $w(t)$ filtered by a single integrator. Due to this filtering, emphasis is placed on the estimation of the low-frequent behaviour, which is important due to the presence of Coulomb friction in the robot. $\tilde{y}(t)$ represents the measured position error, and $\eta(t)$ the measurement noise. The discrete equivalent of (3.7) is:

$$\begin{aligned}x(k+1) &= E(T)x(k) + f(T)u(k) + g(T)w(k) \\ \tilde{y}(k) &= cx(k) + \eta(k)\end{aligned}, \quad (3.8)$$

where

$$x(k) = [e(k) \quad \dot{e}(k) \quad v(k)]^T, \quad (3.9)$$

$$E(T) = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, f(T) = \begin{bmatrix} T^2/2 \\ T \\ 0 \end{bmatrix}, g(T) = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, c = [1 \quad 0 \quad 0]. \quad (3.10)$$

A Kalman observer provides an optimal reconstruction of the state in the presence of the modeling uncertainty and the measurement noise:

$$\begin{aligned}\hat{x}(k+1) &= E(T)\bar{x}(k) + fu(k) \\ \bar{x}(k) &= \hat{x}(k) + K(\tilde{y}(k) - c\hat{x}(k)),\end{aligned}\quad (3.11)$$

where \bar{x} denotes the updated estimate of all states and K the optimal Kalman gain, as computed by the MATLAB-routine `dlqe`. Each time step, the estimate is updated based on the difference between the measured and the predicted error. In the experimental setup, a time delay of approximately two time steps is present. Therefore, this time delay is compensated by extending the observer state with two states $e(k-2)$ and $e(k-1)$, and treat the current measurement as $e(k-2)$. Then, the observer predicts $e(k)$, which can be used in the control law.

3.5 Tuning the controller parameters

3.5.1 Choice of α

The parameter α determines the behaviour inside the vicinity S . According to equations (2.12-2.14), the parameter α determines c_δ . When s is in the vicinity S of the sliding manifold, the reaching law equals $\frac{s}{T}$, and the control law is given by

$$u = u_{ref} - c_\delta(T)A_\delta(T) - \frac{s}{T} = u_{ref} - c_\delta(T)(A_\delta(T) + \frac{1}{T}I_n)x(k). \quad (3.12)$$

This control law consists of a feedforward part and a linear state feedback. For small αT , c_1 approaches α , since

$$\left. \begin{aligned} \delta &= \frac{e^{-\alpha T} - 1}{T} \approx \frac{-\alpha T}{T} = -\alpha, \\ c_1 &= -\delta = \alpha \\ c_\delta &= [c_1 \quad 1]P_1^{-1}(T) \approx [c_1 \quad 1] \end{aligned} \right\} \rightarrow c_\delta \approx [\alpha \quad 1]. \quad (3.13)$$

Further, for small T , the term A_δ in equation (3.12) is negligible, and the control law becomes

$$u = u_{ref}(k) + \frac{1}{T}[\alpha \ 1]x(k). \quad (3.14)$$

The linear feedback actually is a PD-action, which approximately consists of the proportional action $\frac{\alpha}{T}$ and the derivative action $\frac{1}{T}$. This means that the choice for α could be done by tuning a PD-controller on the measured frequency response of each joint. The smaller the sampling time T , the higher the feedback gain, and thus the smaller the influence of disturbances. The approximate gain $\frac{\alpha}{T}$ is limited due to the high-frequency behaviour of the system, as can be seen in figure 3.2. Exciting the higher frequencies can result in chattering or even instability.

Tuning a PD-controller on the measured frequency response is not possible in the current setup, since a Kalman filter is used to estimate the states. In this report, the Kalman filter is considered as a part of the controller. The transfer function of the Kalman filter with the controller is derived in appendix C, and shown in figure C.1. The choice for α will be done experimentally, because the combined transfer function of the controller and the Kalman filter is not very convenient for tuning. The performance criterion is the position tracking error, combined with the constraint that s should stay in S . The values for α , together with the corresponding P and D action, that give the best results are given in table 3.1. In all experiments, a sampling time $T = 1 \cdot 10^{-3}$ is used. The values for joint 1 and 2 give very high feedback gains in the

Joint	α_i	P	D
1	200	$181 \cdot 10^{-3}$	1091
2	200	$181 \cdot 10^{-3}$	1091
3	5	7830	1573

Table 3.1: Final values of tuning parameter α_i and corresponding PD-actions

vicinity S , without leading to instability. The gains can become this high because the error predicted by the Kalman filter is used, which gives a phase lead. The maximum allowable value for α_3 that yields a stable system in the vicinity S is much lower. Using the controller transfer as calculated in appendix C, the open loop transfer of each joint can be calculated. These are given in figure 3.3. From this figure, it

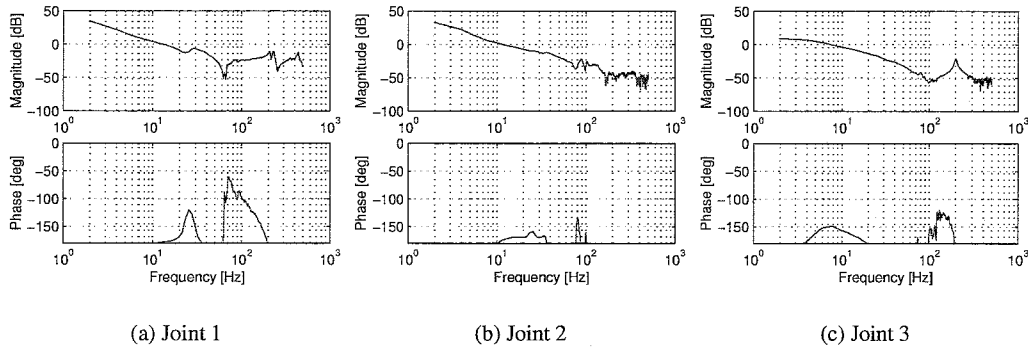


Figure 3.3: Open loop transfers for each joint

can be concluded that the phase margins for joint 1 and 2 are very small. This doesn't lead to instability, because the controller switches to another mode if the switching function s leaves the vicinity S . From figure 3.3 (c), it can be argued that the gain of the controller for joint 3 could be raised. However, this leads to chattering during the experiments. The reason for this apparent contradiction is that the large resonance peak is amplified with a high control gain, and thus easily excites vibrations and consequently leads to control chattering.

3.5.2 Choice of σ , q and r

The parameters σ , q and r determine the behaviour in the reaching phase, and according to equation (2.19), the width of the vicinity S . The width of the vicinity S is mostly determined by σ . The parameters q and r have less influence on the width, but are useful for finetuning. To guarantee stability, σ has to be larger than the model uncertainty. This model uncertainty is estimated online in the Kalman filter with the parameter v . Table 3.2 gives the maximum of the estimated model uncertainty for each joint, and the chosen value for σ . The parameter r increases the width of the vicinity S when the error increases, so that s stays in S . This is especially useful in the parts of the trajectory with a high-frequency content, when the model is not very accurate. The parameter q is important for a fast response in the reaching phase. The experimentally determined values of the parameters q and r are also listed in table 3.2.

Joint	$\max(v)$	σ	q	r
1	5	15	15	20
2	9	25	15	40
3	40	60	40	60

Table 3.2: Model uncertainty and final tuning parameters

Chapter 4

Results

4.1 Results of the sliding mode controller

The controller is designed to control the predicted estimated error $\hat{e}(k+2)$. It is assumed that this is a good estimate of the real tracking error. The results for each joint are shown in a phase portrait, since it gives information about the position error e_i , the velocity error \dot{e}_i , and the switching function s . Further, the position error as a function of time is given, and the applied control voltage for each joint.

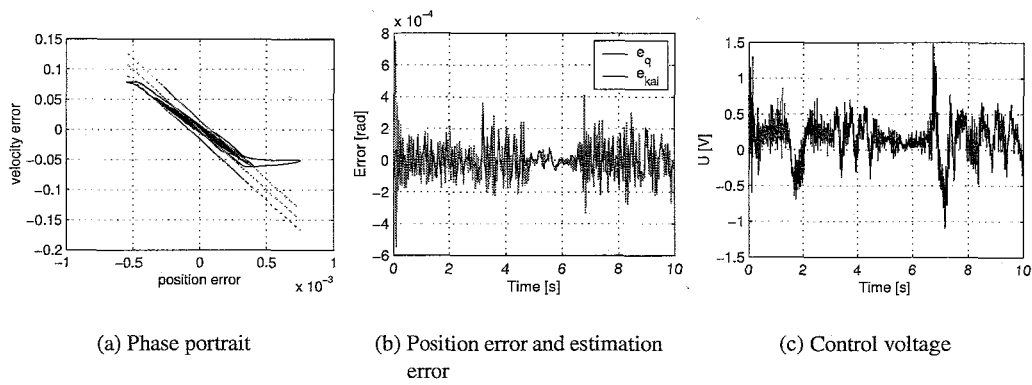


Figure 4.1: Experimental results joint 1

From figures 4.1-4.3, it can be concluded that s stays within S for all three joints, except for a few peaks. These peaks coincide with peaks in the reference trajectory in joint coordinates for joint 3, shown in figure 3.1. Further, it can be seen that for joint 3 the error is much larger than for joint 1 and 2. This is caused by the lower feedback gain of the controller of joint 3. The mean value of the absolute errors for the joints are $0.9 \cdot 10^{-4}$, $1.8 \cdot 10^{-4}$ and $9.4 \cdot 10^{-4}$ respectively. Figures 4.1- 4.2 (b) show that for joint 1 and 2 the position error and the estimation error, i.e. $e_{kal} = q(k-2) - \hat{q}(k-2)$, are very similar. It is obvious that the controller cannot control better than the quality of the estimation.

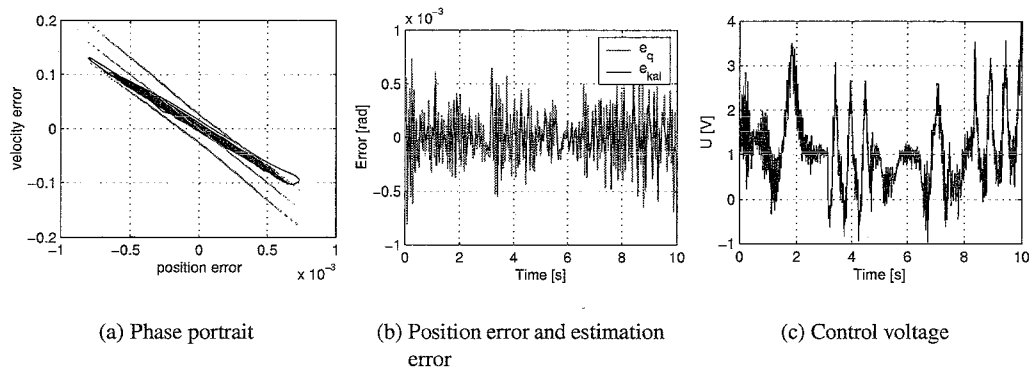


Figure 4.2: Experimental results joint 2

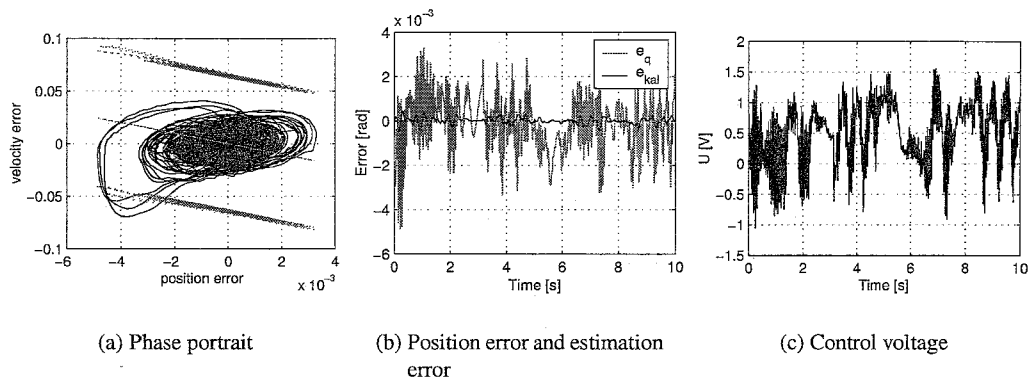


Figure 4.3: Experimental results joint 3

4.1.1 Discussion

The switching function is based on the predicted error, thus the controlled variable is this error. Unfortunately, the assumption that the predicted error is a good estimate for the actual measured error is false, as can be seen in figure 4.4. The actual tracking error is approximately the same as the sum of the estimation

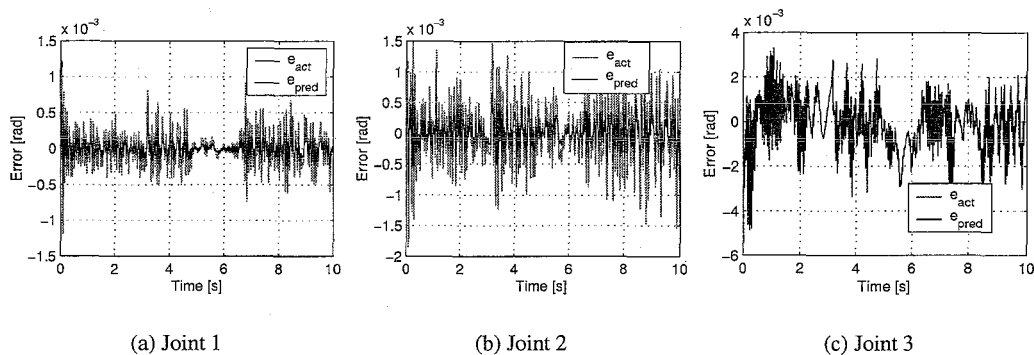


Figure 4.4: Comparison actual and predicted error

error and the predicted error. This means that for joint 1 and 2 the actual error is approximately twice as large as the predicted error, whereas for joint 3 the actual error is approximately the same as the predicted error. Reducing the estimation error also reduces the predicted error, and therefore has a double effect on the actual error. Thus, to improve the results for the actual tracking error, the Kalman filter should be re-designed to give a better estimate of the tracking error. The filter should rely less on the model, and more on the measurement. This however, will lead to the presence of more noise in the state estimate, and therefore to a lower gain for the controller to avoid chattering. This in turn will lead to a larger tracking error. Thus, to achieve considerable improvements in the performance, further tuning of both the Kalman filter and the controller for all joints is needed to achieve good results. The design of the filter and the controller cannot be done separately, since the interactions between them are too large to be neglected in the design.

4.2 Comparison with H_∞ controller

The results of the H_∞ controller discussed in [15] are compared with the actual measured error of the sliding mode controller in figures 4.5-4.7. The errors are plotted for the same part of the reference trajectory. From these figures, it can be concluded that for each joint, the sliding mode controller achieves a tracking

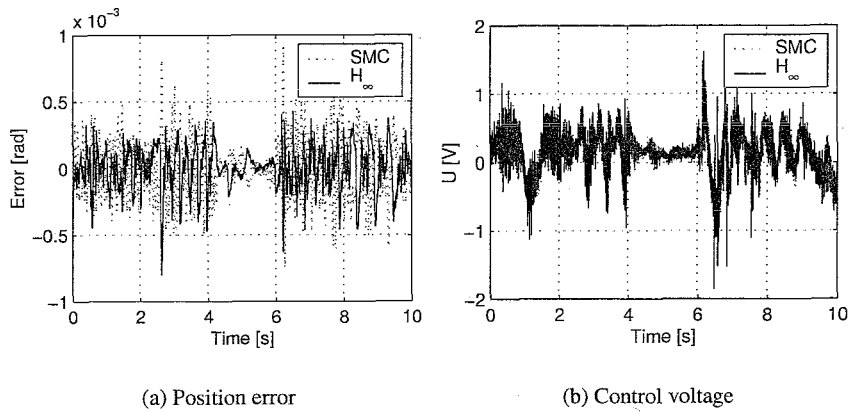


Figure 4.5: Comparison H_∞ (black) and SMC (red dotted) results joint 1

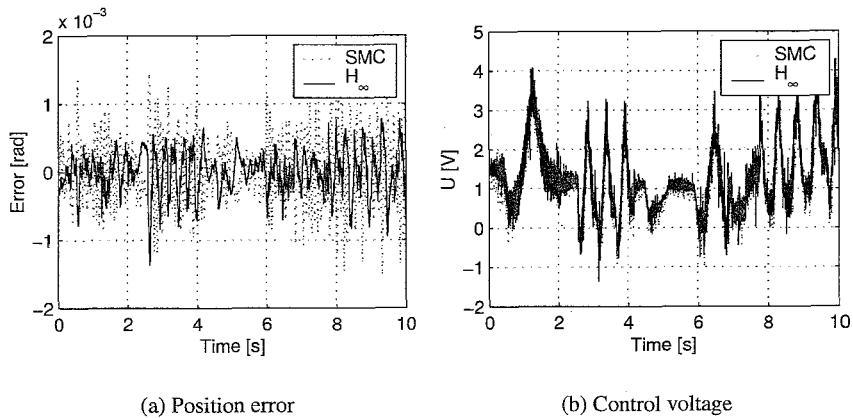
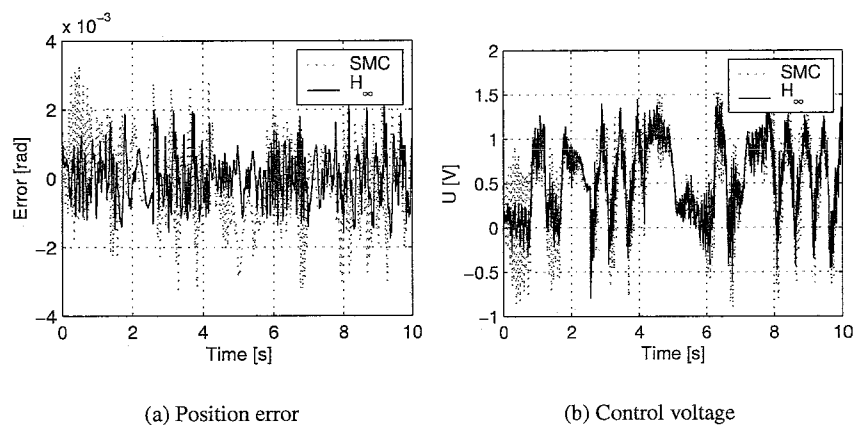


Figure 4.6: Comparison H_∞ (black) and SMC (red dotted) results joint 2

error which is greater than the tracking error of the H_∞ controller. The predicted error (not shown in the figures) is in the same order of magnitude as the error of the H_∞ controller. An advantage of the sliding mode controller is that the computational costs together with the Kalman filter are considerably lower. Taking into account that the error is limited by the quality of the Kalman filter, it can be concluded that the sliding mode concept is very promising.

Figure 4.7: Comparison H_∞ (black) and SMC (red dotted) results joint 3

Chapter 5

Conclusions & Recommendations

In this report, a discrete time sliding mode algorithm is designed and validated on the experimental setup of a RRR-robot. The design of the controller in the sliding mode can be done by tuning a PD-controller on the measured frequency response. The performance in terms of the tracking error is less than with an H_∞ controller, but the computational costs for the control effort are considerably lower. The tracking performance is limited by the quality of the Kalman filter and high-frequent model uncertainties. Further tuning of both Kalman filter and controller is needed to achieve a performance at the same level as the H_∞ controller.

Appendix A

Coordinate transformation

$$x = P_1(T)P_2(T) \begin{bmatrix} \tilde{x} \\ s \end{bmatrix}, \tilde{x} \in \mathbb{R}^{n-1}. \quad (\text{A.1})$$

P_1 is given by:

$$P_1(T) = \begin{bmatrix} b_\delta & \dots & A_\delta^{n-1} b_\delta \end{bmatrix} \begin{bmatrix} a_1 & \dots & \dots & a_n \\ a_2 & \dots & a_n & 0 \\ \vdots & \ddots & \vdots & \vdots \\ a_n & \dots & 0 & 0 \end{bmatrix}, \quad (\text{A.2})$$

where a_i are the characteristic polynomial coefficients:

$$|\delta I_n - A_\delta| = a_n \delta^n + a_{n-1} \delta^{n-1} + \dots + a_1 \delta + a_0. \quad (\text{A.3})$$

P_2 is given by:

$$P_2(T) = \left[\begin{array}{c|c} I_{n-1} & 0_{(n-1) \times 1} \\ \hline -c_1(T) & 1 \end{array} \right], \quad (\text{A.4})$$

and

$$c(T) = c_\delta(T)P_1(T) = \begin{bmatrix} c_1(T) & 1 \end{bmatrix}. \quad (\text{A.5})$$

Appendix B

MATLAB-algorithms

```
function [c_d, Ad, bd, A, b]=calc_cd(Ac,bc,T,alpha)
% [c_d, Ad, bd, A, b] = calc_cd(Ac,bc,T,alpha)
%
% calculate sliding manifold coefficients for the
% discrete-time equivalent of the continuous system  $\dot{x} = Ax+bu$ 
% T = sample time
% alpha = desired decay rate eigenvalues if  $s=0$ 
% c_d = sliding manifold coefficients,  $s=c_d x_d$ 
% further output are the system matrices of:
%  $\Delta x_d(k)=A_d(T)x_d(k)+b_d(T)u(k)$ 
%  $x(k+1)=A(T)x(k)+b(T)u(k)$ 

n=length(Ac);

%%% convert A,b to discrete equivalents acc. to eq. (3)
sys=ss(Ac,bc,eye(n),0);
sysd=c2d(sys,T);
[A,b,c,d]=ssdata(sysd);
Ad=(A-eye(n))/T;
bd=b/T;

%%% determine coeff of char pol  $\det(\Delta I-Ad)$ 
[num,cp]=ss2tf(Ad,bd,c,d)

%%% calculate coordinate transformation matrices P1 acc. to eq. (9)
P1a=[];
P1b=[];

for i=1:n
    P1a=[P1a Ad^(i-1)*bd];
end
P1b=hankel(cp(1:end-1)*hankel([zeros(n-1,1);ones(1,1)]));

P1=P1a*P1b;

%%% desired eigenvalues for All asympt. stable acc. to eq. (14)
for i=1:n-1
    eval(['delta_' num2str(i) ']=(exp(-alpha(' num2str(i) ')*T)-1)/T;'])
end
```

```
%%% calculate elements of c1 acc. to eq. (15)
c1=[];

% calculate product function in eq. (15)
syms delta

p=1;
for j=1:n-1
    eval(['p=(delta-delta_' num2str(j) ') *p;'])
end

% calculate elements c_i of vector c1
for i=1:n-1
    eval(['c_' num2str(i) '=1/factorial(i-1)*diff(p,i-1);'])
    eval(['c_' num2str(i) '=subs(c_' num2str(i) ',delta,0);'])
    eval(['c1=[c1 c_' num2str(i) '];'])
end

%%% calculate vector c_d, defining sliding manifold, acc to eq. (16)
c_d=double([c1 1]*inv(P1));
```

Appendix C

Transfer function Kalman filter with controller

We are interested in the transfer from measurement y to the control input u . This transfer can be calculated analytically. The equations for the Kalman filter are:

$$\begin{aligned}\hat{x}(k+1) &= E(T)(I_n - Kc)\hat{x}(k) + fu(k) + EKy(k) \\ \bar{x}(k) &= (I_n - Kc)\hat{x}(k) + Ky(k)\end{aligned}\quad (C.1)$$

so

$$\begin{aligned}\hat{x}(k) &= H_{\hat{x}y}y + H_{\hat{x}u}u \\ \bar{x}(k) &= [(I_n - Kc)H_{\hat{x}y} + K]y + [(I_n - Kc)H_{\hat{x}u}]u\end{aligned}\quad (C.2)$$

The control law for $s \in S$ is

$$u(k) = c_\delta \begin{bmatrix} e(k) \\ \dot{e}(k) \end{bmatrix} = \begin{bmatrix} \frac{e^{-\alpha T} - 1}{T^2} & 1 \end{bmatrix} \begin{bmatrix} e(k) \\ \dot{e}(k) \end{bmatrix} = H_{SMC}\bar{x},\quad (C.3)$$

with $\bar{x} = [e(k-2) \ e(k-1) \ e(k) \ \dot{e}(k) \ v(k)]^T$. Combining (C.2) and (C.3) gives

$$\bar{x}(k) = [(I_n - Kc)H_{\hat{x}y} + K]y + [(I_n - Kc)H_{\hat{x}u}]H_{SMC}\bar{x}\quad (C.4)$$

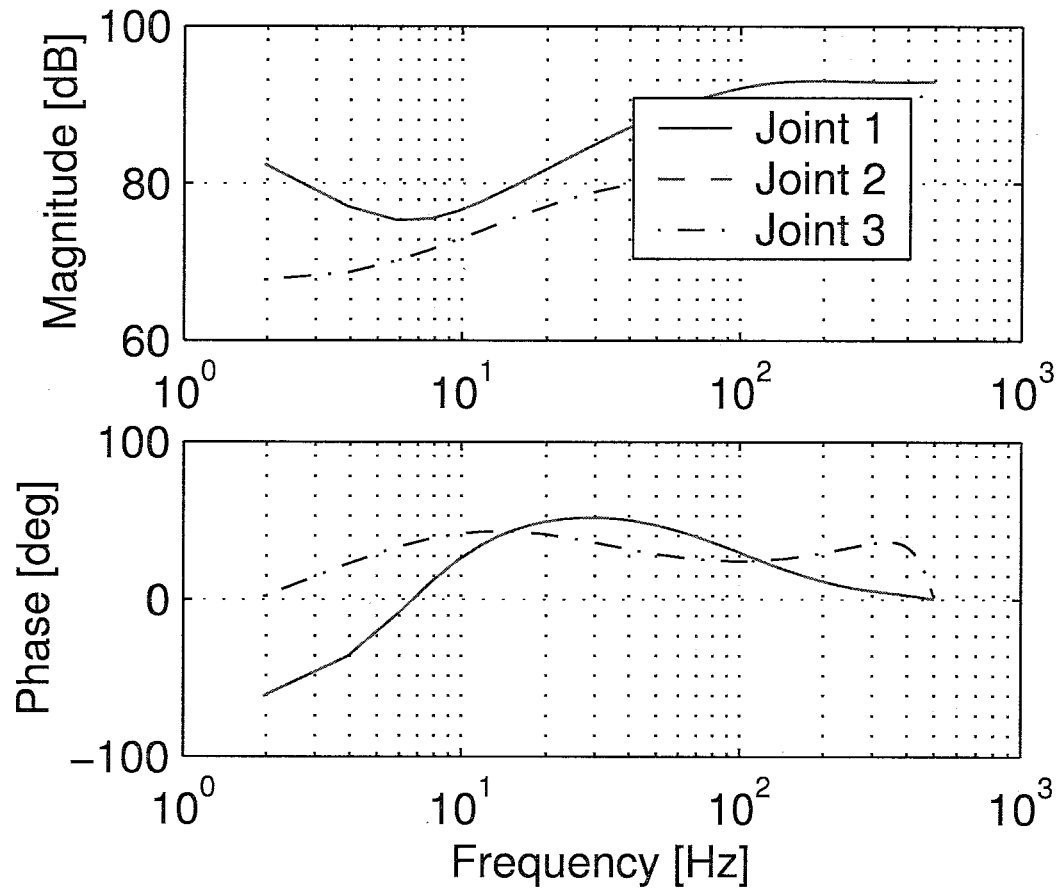
So the transfer $\frac{\bar{x}}{y}$ can be expressed as:

$$\frac{\bar{x}}{y} = \frac{[(I_n - Kc)H_{\hat{x}u} + K]}{I_n - (I_n - Kc)H_{\hat{x}u}H_{SMC}}\quad (C.5)$$

This gives for the transfer from measurement to control input

$$\frac{\bar{u}}{y} = H_{SMC}\frac{\bar{x}}{y} = H_{SMC}\frac{[(I_n - Kc)H_{\hat{x}u} + K]}{I_n - (I_n - Kc)H_{\hat{x}u}H_{SMC}}\quad (C.6)$$

The bode plots of this transfer for each joint, using the final controller settings from table 3.5.1, are given in figure C.1.

Figure C.1: Transfers Kalman filter and SMC controller for $s \in S$

Bibliography

- [1] J.Y. Hung, W. Gao, J.C. Hung, "Variable Structure Control: A Survey," *IEEE Transactions On Industrial Electronics*, Vol. 40, No. 1, pp. 2-22, February 1993.
- [2] V.I. Utkin, "Sliding Mode Control Design Principles and Applications to Electric Drives," *IEEE Transactions On Industrial Electronics*, Vol. 40, No. 1, pp. 23-36, February 1993.
- [3] W. Gao, J.C. Hung, "Variable Structure Control of Nonlinear Systems: A New Approach," *IEEE Transactions On Industrial Electronics*, Vol. 40, No. 1, pp. 45-55, February 1993.
- [4] Č. Milosavljević, "General Conditions for the Existence of a Quasi-sliding Mode on the Switching Hyperplane in Discrete Variable Structure Systems," *Automatic Remote Control*, Vol. 46, pp. 307-314, 1985.
- [5] W. Gao, Y. Wang, A. Homaifa, "Discrete-time Variable Structure Control Systems," *IEEE Transactions on Industrial Electronics*, Vol. 42, No. 2, pp. 117-122, April 1995.
- [6] G. Golo, Č. Milosavljević, "Parabolic and Triangular Wave Oscillator With Sliding Mode," *Electronic Letters*, Vol. 32, No. 17, pp. 1535-1536, August 1996.
- [7] G. Golo, Č. Milosavljević, "Two-phase Triangular Wave Oscillator Based on Discrete-time Sliding Mode Control," *Electronic Letters*, Vol. 33, No. 22, pp. 1838-1839, October 1997.
- [8] G. Golo, Č. Milosavljević, "Robust Discrete-time Chattering Free Sliding Mode Control," *Systems & Control Letters*, Vol. 41, pp. 19-28, 2000.
- [9] A.M. van Beek, RRR-robot: Instruction manual, WFW report 98.012, Eindhoven University of Technology, Faculty of Mechanical Engineering, Section Systems & Control, March 1998.
- [10] A.M. van Beek, RRR-robot: Design of an Industrial-like Test Facility for Nonlinear Robot Control, WFW report 98.014, Eindhoven University of Technology, Faculty of Mechanical Engineering, Section Systems & Control, May 1998.
- [11] D. Kostić, R. Hensen, B. de Jager, M. Steinbuch, "Modeling and Identification of an RRR-robot," in *Trans. of IEEE Conference On Decision and Control*, pp. ,Orlando, Florida, December 4-7, 2001.
- [12] D. Kostić, R. Hensen, B. de Jager, and M. Steinbuch, "Closed-form Kinematic and Dynamic Models of an Industrial-like RRR Robot," in *Trans. IEEE International Conference on Robotics and Automation*, pp. 1309-1314, Washington D.C., USA, May, 2002.
- [13] A. den Hamer, "Adaptive Control of RRR-manipulators," report following 6-weeks trainingship at DCT group in 2002.
- [14] D. Kostić, R. Hensen, B. de Jager, and M. Steinbuch, "Experimentally Supported Control Design for a Direct Drive Robot," in *Trans. IEEE Conference on Control* 186-191, Glasgow, United Kingdom, September, 2002.
- [15] D. Kostić, B. de Jager, and M. Steinbuch, "Control Design for Robust Performance of a Direct-drive Robot," accepted for *IEEE Control Appl. Conf.*, Istanbul, June 23-25, 2003.