Discrete-Time Sliding Mode Control

Govert Monsees

Stellingen

behorende bij het proefschrift

Discrete-Time Sliding Mode Control

Govert Monsees

Delft, 4 juni 2002

1. De kracht van discrete-time sliding mode control t.o.v. continuous-time sliding mode control is dat de onrealistische aanname om het ingangssignaal met oneindige frequentie te schakelen, reeds bij het ontwerp van de regelaar niet meer meegenomen dient te worden.

2. Een zwakte van discrete-time sliding mode control is dat de praktische onhaalbaarheid van fenomenen als sliding mode, ordereductie en invariantie t.o.v. sommige verstoringen niet door theoretische aannamen verborgen worden, hetgeen bij continuous-time sliding mode control wel het geval is.

3. Voor relatief hoge bemonsteringsfrequentie geven een continuous-time sliding mode controller en een discrete-time sliding mode controller gelijke prestaties. Voor een relatief lage bemonsteringsfrequentie is een discrete-time sliding mode controller duidelijk te prefereren boven een continuous-time sliding mode controller zoals men intuïtief reeds zou verwachten. (Dit proefschrift)

4. Het samenvoegen van regeltechniek groepen vanuit verschillende faculteiten, zoals binnen de Technische Universiteit Delft gaande is, illustreert eens te meer het multidisciplinaire werkgebied van de regeltechniek, hetgeen als motivatie kan dienen voor de vorming van een faculteit regeltechniek.

5. Het stemrecht in een democratie dient men te benutten om voor een bepaald beleid te stemmen en niet tegen een beleid zoals helaas de trend van de laatste tijd in Nederland is.

6. Wanneer iedereen op tijd zou komen zou menig vergadering 15 minuten later gepland kunnen worden.

7. In verre, voor ons vreemde, landen ware het handig de volgorde van bestellen, eten, en betalen te veranderen naar; (1) Betalen, (2) Eten, (3) Bestellen.

8. Oploskoffie is geen oplossing.

9. Vakantiedagen zijn er om opgenomen te worden.

10. Voor de reizende fotograaf zou een geurrecorder een nuttige toevoeging aan een fototoestel zijn.

1. The strength of discrete-time sliding mode control, compared to continuous-time sliding mode control, is that the impractical theoretical foundation of continuous-time sliding mode control (i.e. infinite switching speed) is not an issue during the design phase of the controller.

2. A weakness of discrete-time sliding mode control, compared with continuous-time sliding mode control, is not that properties such as the sliding mode, order reduction, and invariance against certain disturbances no longer apply, but that they so obviously do not apply.

3. The performance of a continuous-time sliding mode controller and a discrete-time sliding mode controller are approximately the same for a relatively large sampling frequency. For relatively low sampling frequencies, the discrete-time sliding mode controller performs better, as could intuitively be expected. (This thesis).

4. Combining control groups from different faculties into one single control group, which is happening within the Delft University of Technology, illustrates the multi-disciplinary field of application of control engineering, providing an argument for the creation of a control-engineering faculty.

5. In a democracy, the right to vote should be used to vote in favour of a policy and not against a policy, which unfortunately is the recent trend in The Netherlands.

6. If everybody would be in time, many meetings could be scheduled 15 minutes later.

7. In some countries, the sequence of ordering, eating, and paying could better be rescheduled to: (1) paying, (2) eating, and (3) ordering.

8. Instant coffee is no solution.

9. Vacation time exists to be used.

10. For the travelling photographer, a scent recorder would be a useful feature on a camera.

Discrete-Time Sliding Mode Control

Govert Monsees

Discrete-Time Sliding Mode Control

Proefschrift

Ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J. T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 4 Juni 2002 om 10:30 uur
door Govert MONSEES
elektrotechnisch ingenieur
geboren te Uden.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. M.H.G. Verhaegen

Toegevoegd promotor:
Dr.ir. J.M.A. Scherpen

Samenstelling Commissie:
Rector Magnificus, Voorzitter
Prof.dr.ir. M.H.G. Verhaegen, Technische Universiteit Delft, promotor
Dr.ir. J.M.A. Scherpen, Technische Universiteit Delft, toegevoegd promotor
Prof.dr. S.K. Spurgeon, University of Leicester
Prof.ir. O.H. Bosgra, Technische Universiteit Delft
Prof.dr. A.A. Stoorvogel, Technische Universiteit Delft
Prof.dr. H. Nijmijer, Technische Universiteit Eindhoven
Dr.ir. J. Swevers, Katholieke Universiteit Leuven

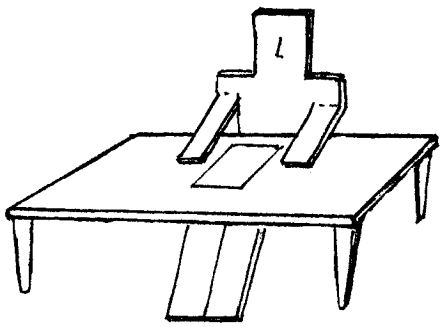
Prof.dr. R. Babuška, Technische Universiteit Delft (reserve lid)

ISBN 90-77017-83-6

Contents

Glossary

I_m	Identity matrix of size $m \times m$
A^T	Transpose of the matrix A
$\lambda(A)$	Eigenvalues of the matrix A
$\mathbb{R}$	Collection of real numbers
$\|x\|$	2-Norm of the vector x defined as $\sqrt{x^T x}$
$\|A\|$	2-Norm of the matrix A defined as $\max \frac{\|Ax\|^2}{\|x\|^2}$
$\|x(t)\|$	Absolute value of the scalar x at time t
$\|\|x(t)\|\|$	2-norm of the vector x at time t
$\dot{x}(t)$	Time derivative of the vector $x(t)$, i.e. $\dot{x}(t) = dx(t)/dt$
$\ddot{x}(t)$	Second time derivative of the vector $x(t)$, i.e. $\ddot{x}(t) = d^2x(t)/dt^2$
AD	Analog to Digital
BPMD	Bounded Preview Method of Dichotomies
BPMDSM	Bounded Preview Method of Dichotomies with Sliding Mode Control
CSMC	Continuous-Time Sliding Mode Control
DA	Digital to Analog
DSMC	Discrete-Time Sliding Mode Control
FIR	Finite Impulse Response
GCCF	Generalized Controller Canonical Form
LMI	Linear Matrix Inequality
MD	Method of Dichotomies
MIMO	Multiple Input, Multiple Output
OSMC	Output Based Sliding Mode Control
OCSMC	Output Based Continuous-Time Sliding Mode Control
ODSMC	Output Based Discrete-Time Sliding Mode Control
RMS	Root Mean Square
SAZC	Steering Along the Zeros Control method
SCOOP	Seat COmfort OPtimization
SDI	Stable Dynamic Inversion
SISO	Single Input, Single Output
SMC	Sliding Mode Control
SSMC	State Based Sliding Mode Control
SCSMC	State Based Continuous-Time Sliding Mode Control
SDSMC	State based Discrete-Time Sliding Mode Control
SVCTP	Seat Vibrational Comfort Testing Procedure
VAF	Variance Accounted For, computed as $VAF(r_1, r_2) = \left(1 - \frac{Var(r_1 - r_2)}{Var(r_1)}\right) 100\%$



1

Introduction

Examples of control engineering can be found in many applications. For example the operation of a HiFi-amplifier can be seen as a control problem. The loudspeaker has to be controlled such that some reference trajectory (the music) is replicated. Another example is the generation of some desired voltage by a static inverter, in which case transistors are controlled to maintain some desired output voltage. For both examples, the controller can be composed out of continuous-time, electronic, components. In a way, these electronic controllers can be called continuous-time computers since they compute the control action based on measurements. Another example of continuous-time computers can be found in the earlier air data computers. For example the F-104 Starfighter, a military aircraft manufactured in the 1960s, had a mechanical computer to retrieve the required information from the measured air data. Although, as the previous examples illustrate, it is possible to construct complex controllers out of continuous-time components, digital computers are better suited for the implementation of controllers. Nowadays computers are cheap and flexible. They can perform millions of operations every second and are a lot easier to implement then their mechanical and electrical, continuous-time, counterparts. Another advantage is that they can be incorporated within for example a maintenance monitoring system and can easily be operated from miles away.
The main difference between digital computers and continuous-time computers is the fact that measurements are sampled to be used in the digital computer. A possible block scheme of a discrete-time computer controlling a continuous-time system is sketched in Figure 1.1. In this figure, the controller is composed out of three parts. In the center of the controller block is the control algorithm which is a mathematical description relating the control signal $u[k]$ with the measured output values $y[k]$. The AD-block represents the Analog to Digital (i.e. AD) converter. An AD-converter converts continuous-time signals into discrete-time signals, a procedure which is called *sampling*. This sampling is typically performed at equidistant time instances, separated by what is called the sampling time. The last part of the controller is the DA-converter which stands for Digital to Analog converter. It is assumed in this thesis that the output signal of a DA-converter is kept constant

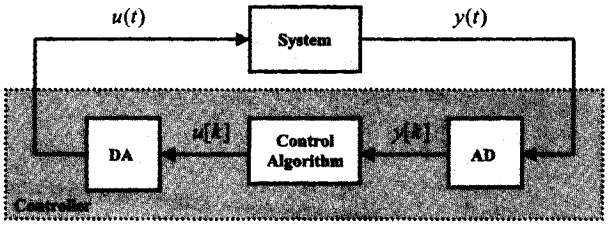


Fig. 1.1: Closed-loop system formed by a discrete-time controller and a continuous-time system.

over the sampling period, a procedure which is called *hold* since the discrete-time value is held (maintained) until the next discrete-time value appears one sampling period later. The combined procedure of AD-conversion and DA-conversion, as described above, is known as *sample-and-hold*. In the previous text it was assumed that the system under control is continuous-time. In some applications the system is discrete-time as well. In those case the AD- and DA-converter are no longer present. However, most systems which have to be controlled are continuous-time thereby necessitating the sample-and-hold operation. The combination of a continuous-time system and a discrete-time controller is often called a sampled-data system.
The design of a controller for a sampled-data system can be performed in many ways, in (Chen and Francis 1996) the following design options are given:

I Sampled-data implementation of a continuous-time controller.

II Discrete-time design using a discretised representation of the continuous-time system.

III Direct sampled-data design.

This thesis studies the design of one specific control technique according to the second method. This control technique known as Sliding Mode Control (SMC) has its roots in (continuous-time) relay control. It originated in the Soviet Union somewhere in the late 1950s, but was not published outside the Soviet Union until the publications (Itkis 1976) and (Utkin 1977). In the years thereafter, the sliding mode research community expanded rapidly and the number of publications grew correspondingly. SMC is inherently a continuous-time control technique since it relies on an infinite switching frequency of the input signal. This assumption can never be met in practice. Especially for discrete-time controllers, the input signal can only be changed at the sampling instances thereby limiting the switching frequency to the sampling frequency. However, in many applications the assumption of an infinite switching frequency can more or less be justified. If the sampling is performed much faster than the dynamics of the system under control, the influence of the bounded switching frequency will be limited. It is therefore a common approach to design sliding mode controllers in the continuous-time domain, even though the system is computer-controlled (Young and Özgüner 1999), corresponding with the earlier mentioned design option *I*. We call the result of this design

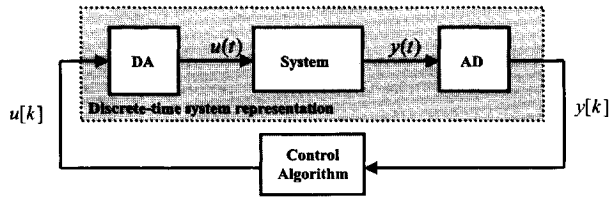


Fig. 1.2: Closed-loop system formed by a discrete-time controller and a continuous-time system. The DA- and AD-converters have been combined with the continuous-time system illustrating the discrete-time description of the system.

procedure a Continuous-Time Sliding Mode Controller (CSMC) since it is based on a continuous-time model of the system and ignores the sampling issue. The success of the derived controller will amongst many other things strongly depend on the sampling frequency. The faster the sampling is performed, the smaller the effect of the sampling will be. For a relatively low sampling frequency, the limited switching frequency can lead to undesirable effects in the input signal or even instability of the closed-loop system. An alternative sliding mode controller, specifically designed for sampled-data or discrete-time systems is considered in this thesis, resulting in a Discrete-Time Sliding Mode Controller (DSMC). This approach, which corresponds with the earlier mentioned design option *II*, can be illustrated by the block-scheme presented in Figure 1.2. Comparing Figure 1.2 with Figure 1.1 we see that the DA- and AD-converters are combined with the continuous-time system. The dotted box in Figure 1.2 behaves as it were a discrete-time system, at least as far as the controller can "see" since the controller only has information of the system at the sampling instances. Therefore, the dotted box in Figure 1.2 is modeled by a discrete-time model to be used by the controller. Of course, precious (continuous-time) information of the system could be lost by this operation. However, in many cases there is no continuous-time model of the system available in the first place. Often, a system description is retrieved from (discrete-time) measurements naturally leading to a discrete-time model.

While a discrete-time model, instead of a continuous-time model, is used for the design of the controller, the design of the sliding surface is similar to the original design procedure. The main difference between a CSMC and a DSMC is, apart from the model of the system under control, the implementation of the control law. A CSMC typically has a discontinuous control term which starts switching with infinite frequency once the closed-loop system is driven in the sliding mode. It is shown in this thesis that for discrete-time and sampled-data systems it is best to eliminate this discontinuous control term from the control signal. Instead, a control law is used which is entirely linear. This eliminates the chattering behaviour which can be caused by a discontinuous control signal and improves the accuracy. Apart from defining a State-Based Discrete-Time Sliding Mode Controller, special attention is given to the problem of designing an Output-Based Discrete-Time Sliding Mode Controller (ODSMC) as well. While ODSMC is considered in some publications,

see for example (Sha and Bajić 2000), (Chan 1999), or (Kaynak and Denker 1993), these are all based on a transfer function representation of the system. In this thesis a design procedure is presented for ODSMC based on a state-space representation of the system to be controlled thereby improving the application of ODSMC to multiple input control problems. Both the output-based and state-based DSMCs are introduced for the special case where the system has to be stabilized in the origin of the state-space. However, for many control problems some time varying state- or output-signal needs to be replicated by the system. Therefore, special controller structures are introduced in this thesis which solve this tracking problem using DSMC. For those cases where the DSMC is combined with a feedforward control signal the problem of model inversion is considered as well. The procedure of model inversion can be employed to generate the desired state signal for the case where only the desired output signal is specified. Several simulation studies and laboratory tests are presented which demonstrate the applicability of the proposed controllers.

The research presented in this thesis has been largely inspired by the Seat COmfort OPtimization (SCOOP) project. The main objective of this project is to develop a Seat Vibrational Comfort Testing Procedure (SVCTP). The reason for this project is that few test procedures currently exist in industry, and no international standards define a complete procedure. Fundamental issues such as how to choose mission signals, how to treat the complex behaviour of the person/seat system and how to choose human test subjects are completely lacking from all existing national and international standards. Vehicle and seat system manufacturers currently cannot compare test results obtained by different laboratories or exchange data. This situation is economically damaging because it adds large additional costs to vehicle and seat development programs by eliminating chances for concurrent engineering. In order to enhance reproducibility and comparability of tests performed at different times and/or places standard road profiles are defined (city, rural, and highway) which have to be replicated on a test setup with high accuracy. The test setup can either be formed by a car placed on shakers, as shown in Figure 1.3.a, or a car-seat placed on a moving platform as shown in Figure 1.3.b. Using such a test setup and standardized road profiles greatly enhances the reproducibility compared to tests performed on the road since all circumstances are under control. An important part of the SCOOP project was the development of controllers which give the best possible tracking of the defined (output) profiles with a minimum of iterations. The current state of the art controllers for these test benches typically require several iterations to achieve their best performance. Since these iterations strongly influence the perception of the test persons and require quite some time, there is a need for faster converging control techniques. Therefore, research within the SCOOP project was focused to:

1. Identification. In order to control a system, a model is required describing the essential dynamics. Although models can be constructed from first principles, a procedure called white box modeling, this is quite labor intensive. The other path to follow is called black box modeling in which case a model is con-

(a) (b)

Fig. 1.3: Seat comfort test setups. Figure (a) represents a car placed on four vertical actuators, Figure (b) a seat with even more degrees of freedom.

structed from measured input and output data. Within the SCOOP project, the identification of linear as well as nonlinear models was investigated.

2. Feedforward control. An estimate of the required control signal can be made once a model of the system has been constructed. Of course, the performance of the system under control of this feedforward control signal strongly depends on the quality of the obtained model, stressing the importance of the identification of a good model. The generation of a feedforward control signal leading to perfect tracking by the model is by no means a trivial task. Simple model inversion often leads to an unstable feedforward control signal necessitating special attention to circumvent this problem.

3. Feedback control. Finally, a feedforward controller can be added to reduce the sensitivity to modeling errors and hence improve the performance of the closed-loop system. Since the prime objective of the feedback controller is to optimize the tracking performance, the research has been focused to a control technique which is well known for its robustness properties.

Like most control problems, the test setups considered in the SCOOP project are (digital-) computer controlled making the closed-loop system a sampled-data system. This thesis addresses the second and the third problem (i.e. feedforward control and feedback control), thereby using a stable inversion technique and DSMC. A topic not addressed in this thesis is the inter-sampling behaviour of the continuous-time system under the control of a discrete-time controller. As stated before, the design approach used in this thesis corresponds with the earlier introduced design option *II*, i.e. a discrete-time model for the continuous-time system is obtained which is used for the controller design. Therefore, only the values of the system variables at the sampling instances are considered. Nothing is guaranteed for these signals between the sampling instances, i.e. the inter-sampling behaviour of the closed-loop system. Also the quantization effect of sampling is not considered. In

this thesis it is assumed that, although discrete in time, the sampled signal exactly matches the true signal at the sampling instances. Also the problem of designing a DSMC to control a system for which a nonlinear discrete-time model is available is not considered.

1.1 Outline of the Thesis

In Chapter 2 an overview of linear, continuous-time, sliding mode control is presented. Basic principles as the sliding mode, reaching mode, reaching law, order reduction, matched- and unmatched uncertainties, and invariance against the latter, are introduced to the reader. Special attention is given to the switching term which can lead to chattering in practical applications. This problem can be reduced by the use of a softened switching gain and by using an adaptive switching gain.
Chapter 3 is devoted to linear, discrete-time, sliding mode control. Several reaching laws are considered and a design procedure for state-based and output-based sliding mode controllers is presented. A varying reaching law as well as several adaptive switching gain implementations are considered.
Chapter 4 considers the feedforward control problem for linear, possibly nonminimum phase, discrete-time systems. Three methods are considered and special attention is paid to the case where only a finite preview of the desired output signal is available.
Chapter 5 treats the problem of tracking. Two strategies are presented. First the problem of replicating some desired output trajectory perfectly is considered, leading to a target tracking controller. Then the problem of tracking some desired dynamics is treated by the introduction of a model reference controller.
Chapter 6 describes some practical tests in a laboratory environment of the proposed discrete-time sliding mode controllers. Also a comparison between a continuous-time sliding mode controller and a discrete-time sliding mode controller is presented.
In Chapter 7 the problem of controlling both the fuel injector and the throttle position of a spark ignition engine is considered.
Finally, Chapter 8 presents the conclusions and recommendations for further research.

1.2 Contributions

The main contributions of this thesis are:

- *Adaptive switching gain definition for continuous-time sliding mode control (Chapter 2).* Although large switching gains (theoretically) lead to extremely strong robustness properties, practical limitations rule out this choice. Large switching gains may lead to extreme wear in gear-boxes, extreme power consumption, and the excitation of unmodeled fast dynamics. Therefore, the switching gain should be chosen as small as possible while still guaranteeing the required robustness. Another possibility is to define an adaptive switching gain, which automatically tunes the switching gain to the actual circumstances. Two methods available in the control literature are reviewed and

compared. The proof given in the literature for one of these methods is found to be incorrect, an alternative proof is presented.

- *A general, systematic, design procedure for linear discrete-time sliding mode control for both the state-based and the output-based case (Chapter 3).* For the cases where the full system state is accessible or only the system outputs are available, a general design procedure is presented. The design procedure is based on a state-space representation of the system and can be applied to multiple input systems.

- *Disturbance estimation (Chapter 3).* Since in the presented approach the switching function (instead of the normally used system state) is used for the disturbance estimation, the procedure can be used for state-based and output-based sliding mode controllers. Incorporating the estimated disturbance reduces the quasi sliding mode band considerably for relatively slow varying disturbances.

- *Varying reaching law for discrete-time sliding mode control (Chapter 3).* The reaching law has an important effect on the performance of a discrete-time sliding mode controller. It is shown that a linear reaching law, possibly in combination with disturbance estimation, gives the best performance. A varying reaching law is proposed to increase the precision close to the sliding surface.

- *Adaptive switching gain definition for discrete-time sliding mode control (Chapter 3).* The two methods found in the literature for continuous-time sliding mode are applied to discrete-time sliding mode control. To overcome the disadvantages they possess a new adaptation law is introduced.

- *Feedforward control (Chapter 4).* A finite preview time feedforward control technique for linear, discrete-time, nonminimum phase systems is proposed. Furthermore, special attention to zeros on the unit circle is given.

- *Tracking control (Chapter 5).* The problem of tracking is solved in two different ways, both using of discrete-time sliding mode control. The first strategy aims at perfect tracking of some desired output trajectory while the second strategy aims at tracking some specified desired closed-loop dynamics specified in the form of a reference model. For both strategies the state-based as well as the output-based approach are considered.

- *Experimental evaluation of discrete-time sliding mode control (Chapter 6).* The proposed control techniques are evaluated on a laboratory test setup. Also a comparison between discrete-time sliding mode control and continuous-time sliding mode control is presented for a relatively fast and slow sampling frequency.

- *Automotive engine control (Chapter 7).* The problem of controlling the fuel injection system and the throttle position of a spark ignition engine is solved using sliding mode control. A so-called mean value model is used to describe

the dynamics. Using this model, two continuous-time, nonlinear, sliding mode controllers are designed to solve the control problem. The controllers are demonstrated by the use of a highly accurate industrial simulation model.

1.3 Publications

- **Part of book:**

  Monsees, G. and J. M. A. Scherpen (2002) , "Nonminimum phase output tracking using sliding modes", *To appear in Nonlinear and Adaptive Control, editors A. Zinober and F. Labnabhi-Lagarrique.*

- **In journals:**

  Monsees, G., S. K. Spurgeon and P. Puleston (2001), "Air fuel ratio control using sliding modes", *Systems Science* **26**(3), 97–108.

  Puleston, P. F., S. K. Spurgeon and G. Monsees (2001), "Automotive engine speed control: A robust nonlinear control framework", *IEE Proceedings - Control Theory and Applications* **148**(1), 81–87.

  Monsees, G. and J. M. A. Scherpen "Adaptive switching gain for a discrete-time sliding mode controller", International Journal of Control **75**(4), 242–251.

  Puleston, P., G. Monsees and S. K. Spurgeon, "Air/fuel ratio and speed control for low emission vehicles based on sliding mode techniques", *To appear in the Journal of System and Control Engineering.*

  Monsees, G. and J. M. A. Scherpen, "Discrete-time sliding mode control with disturbance estimation", *Submitted.*

  Monsees, G. and J. M. A. Scherpen, "Discrete-Time Output-Based Target Tracking using Sliding Modes", *Submitted.*

- **In conference proceedings:**

  Monsees, G. and J. M. A. Scherpen, (2001) "Automobile road vibration reproduction using sliding modes", *Proceedings of the 7th International Conference & Exhibition, Firenze, Italy.*

  Monsees, G. and J. M. A. Scherpen (2001), "Discrete-time sliding mode control with a disturbance estimator", *Proceedings of the European Control Conference 2001, Porto, Portugal* pp. 3270–3275.

  Monsees, G. and J. M. A. Scherpen (2001), "Output tracking using a discrete-time sliding mode controller with reduced-order state-error estimation", *Proceedings of the 1st IFAC Symposium on System Structure and Control, Prague, Czech Republic.*

  Monsees, G. and J. M. A. Scherpen (2001), "Adaptive Switching Gain for a Discrete-Time Sliding Mode Controller", *Proceedings of the American Control Conference, Chicago, U.S.A.* pp. 1639–1643.

Monsees, G., S. K. Spurgeon and P. F. Puleston (2000), "A combined feedforward/feedback approach to air/fuel control using sliding modes", *Proceedings of the International Conference on Systems Engineering, Coventry, U.K.*

Monsees, G., K. George, J. M. A. Scherpen and M. Verhaegen (1999), "A feedforward-feedback interpretation of a sliding mode control law", *Proceedings of the 7th IEEE Mediterranean Conference on Control and Automation, Haifa, Israel* pp. 2384–2398.

▶ **Report:**

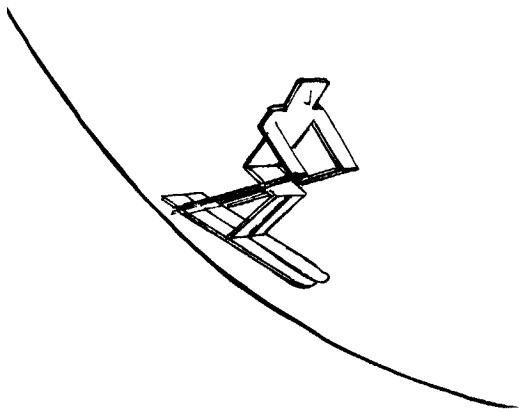
Monsees, G., K. George, J. M. A. Scherpen and M. Verhaegen (2000), Deliverable 10b, SCOOP-project, Technical report, Delft University of Technology.

▶ **Abstracts:**

Monsees, G. and J. M. A. Scherpen (2001), "Discrete-time output-based sliding mode control", *Proceedings of the 20th Benelux Meeting on Systems and Control, Houfalize, Belgium.*

Monsees, G., K. George, J. M. A. Scherpen and M. Verhaegen (1999), "A feedforward-feedback interpretation of a sliding mode control law", *Proceedings of the 18th Benelux Meeting on Systems and Control, Houthalen, Belgium.*

2



Continuous-Time Sliding Mode Control

SMC has its roots in (continuous-time) relay control. It originated in the Soviet Union somewhere in the late 1950s, but was not published outside the Soviet Union until the publications (Itkis 1976) and (Utkin 1977). After these publications, the list of publications concerning SMC grew rapidly and SMC obtained a solid position in the field of both linear- and nonlinear control theory. To obtain a good overview of continuous-time sliding mode we refer to the standard work (Utkin 1992). Other good publications are for example (DeCarlo et al. 1988), (Hung et al. 1993), (Edwards and Spurgeon 1998), and (Utkin et al. 1999), where the latter publications describes several practical applications.

This Chapter presents the basics of well-known continuous-time SMC (CSMC) in order to provide a setting for developing a theory for discrete-time SMC (DSMC) in the next chapter. The principle of CSMC is to define a new output, with the same number of entries as the input (i.e. a square system) and relative degree one, such that the zero dynamics corresponding to the newly defined output are stable. Once this is accomplished, full attention can be given to the stabilization of the newly defined output. Whether the new output is defined on the system's state (i.e. state-based CSMC, SCSMC) or the systems' outputs (i.e. Output-based CSMC, OCSMC) is irrelevant, in either case the result is, if possible, a square system with stable zero dynamics. To illustrate the basic concepts of CSMC we consider the driven pendulum, sketched in Figure 2.1, as an example. After this informal treatment of CSMC, Sections 2.1 and 2.2 present a rigorous design procedure for a SCSMC and a OCSMC respectively.

Let us consider the problem of stabilizing the driven pendulum in the equilibrium point $\theta(t) = 0$ and $\dot{\theta}(t) = 0$. The dynamics of a pendulum with viscous friction can

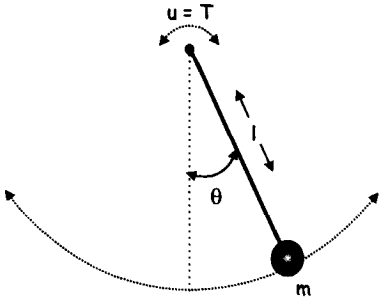


Fig. 2.1: Sketch of the pendulum.

be described by:

$$\ddot{\theta}(t) = -\frac{1}{g}\sin(\theta(t)) - c_f\dot{\theta}(t) + \frac{1}{ml^2}u(t) \tag{2.1}$$

where θ is the angle of the pendulum, g is the gravity constant, m the mass, l the length of the pendulum, and the input u is the applied torque. We now define the new output $\sigma(t)$, called the *switching function*, as

$$\sigma(t) = S_1\theta(t) + S_2\dot{\theta}(t) \tag{2.2}$$

(S_1 and S_2 being scalars where S_2 is nonzero) and we define the control input as

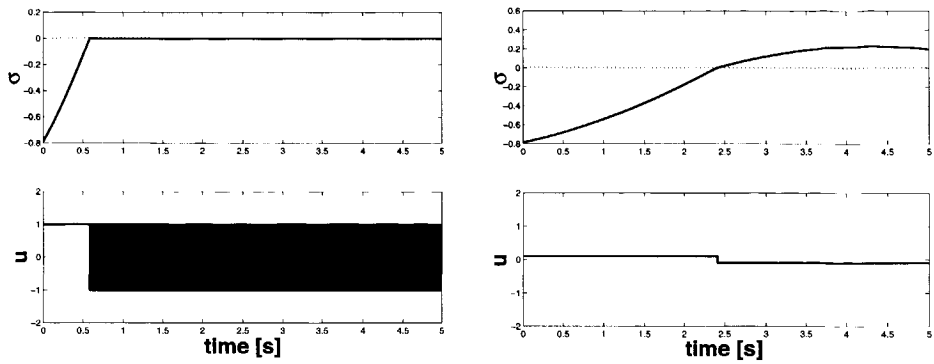
$$u(t) = -K_s\text{sign}(\sigma(t)) \tag{2.3}$$

where the sign function is defined as

$$\text{sign}(\sigma(t)) = \begin{cases} 1 & \text{if} \quad \sigma(t) \geq 0 \\ -1 & \text{if} \quad \sigma(t) < 0 \end{cases} \tag{2.4}$$

Then for the parameters $l = 1$, $m = 1$, $c_f = 0.1$, $K_s = 1$, $S_1 = 1$, $S_2 = 1$, and the gravitational constant being $g = 9.81$, we obtain the simulation results presented in Figure 2.2.a. After approximately 0.5 s, the newly defined output $\sigma(t)$ becomes zero. From that moment on, the input signal starts switching with infinite frequency. For this reason the newly defined output $\sigma(t)$ is called the switching function: the switching function switches the input if it crosses zero. From a mathematical point of view, the differential equation (2.1) can no longer be solved because the input signal is no longer unique. Due to the infinite switching frequency, the control input is both $+K_s$ and $-K_s$. The solution can be found by the approach of (Filippov 1964) who considers the case of differential equations with discontinuous right hand side. Using this work, the problem is solved by approaching the discontinuous point from both directions and forcing the solution to be tangent to the line $\sigma = 0$ (Utkin 1992). While $\sigma(t) = 0$ the control input can also be determined from the equation:

$$\dot{\sigma}(t) = S_1\dot{\theta}(t) + S_2\ddot{\theta}(t) = 0 \tag{2.5}$$



(a) (b)

Fig. 2.2: (a) Plot of the input u (bottom) and the newly defined output σ (top) for $K_s = 1$. (b) Plot of the input u (bottom) and the newly defined output σ (top) for $K_s = 0.1$.

Using the above equation and the system description (2.1) the control input can be determined to be:

$$u_{eq}(t) = \frac{ml^2}{g}\sin(\theta(t)) + \frac{ml^2}{S_2}\left(S_2 c_f - S_1\right)\dot{\theta}(t) \tag{2.6}$$

Applying the above control signal to the system at the moment that $\sigma(t) = 0$ results in exactly the same behavior as applying the hard switching signal presented in Figure 2.2.a. It is therefore called the *equivalent control.* The equivalent control is in fact the low frequency content of the switching signal, or in other words, low pass filtering the infinitely fast switching control input results in the equivalent control as well, i.e the system acts as a low pass filter since it cannot react to the infinitely high switching frequency.
The same experiment is repeated with $K_s = 0.1$, the result is presented in Figure 2.2.b. The first thing to notice is that it takes longer before the switching function is zero. After 2.5 seconds, instead of 0.5 seconds for the previous case, the switching function becomes zero. At that time the input switches (from $u = 0.1$ to $u = -0.1$), it is however not strong enough to keep the closed-loop system at $\sigma(t) = 0$. Apparently the switching gain was chosen too small in this case to stabilize the newly defined output $\sigma(t)$ under the given circumstances. This phenomenon can be studied by constructing a Lyapunov function of the form:

$$V(\sigma) = \frac{1}{2}\sigma^2(t) \tag{2.7}$$

From Lyapunov's stability theory we know that the system (2.1) reaches $\sigma(t) = 0$ in finite time if the above Lyapunov function satisfies:

$$\dot{V}(\sigma) \leq -\eta|\sigma(t)| \tag{2.8}$$

for some strictly positive constant $\eta \in \mathbb{R}$. By differentiating $V(\sigma)$ and realizing that $\sigma(t) = |\sigma(t)| \text{ sign}(\sigma(t))$ the above stability condition reduces to:

$$\dot{\sigma}(t) \leq -\eta \text{ sign}(\sigma(t)) \tag{2.9}$$

Since condition (2.9) gives a condition to reach $\sigma(t) = 0$ in finite time, it is called the *reaching law*. Realizing that $\dot{\sigma}(t) = S_1\dot{\theta}(t) + S_2\ddot{\theta}(t)$ and using the equation for $\ddot{\theta}(t)$ we obtain:

$$-\frac{S_2}{g}\sin(\theta(t)) + (S_1 - S_2c_f)\dot{\theta}(t) + \frac{S_2}{ml^2}u(t) \leq -\eta \text{ sign}(\sigma(t)) \tag{2.10}$$

Substituting the equation for $u(t)$ in the above equation and rearranging terms leads to:

$$K_s \text{ sign}(\sigma(t)) \geq \frac{ml^2}{S_2}\left\{\eta \text{ sign}(\sigma(t)) - \frac{S_2}{g}\sin(\theta(t)) + (S_1 - S_2c_f)\dot{\theta}(t)\right\} \tag{2.11}$$

Whenever condition (2.11) is met, the system will move monotonically towards $\sigma(t) = 0$ and once $\sigma(t) = 0$ is reached it will stay zero. Consequently, to guarantee stability within some region $\theta(t) < \theta_{max}$ and $\dot{\theta}(t) < \dot{\theta}_{max}$, the condition for the switching gain K_s can be determined by the application of the triangle inequality:

$$K_s \geq \frac{ml^2}{S_2}\left\{\eta + \frac{S_2}{g}\sin(\theta_{max}) + (S_1 - S_2c_f)\dot{\theta}_{max}\right\} \tag{2.12}$$

for some positive η. Condition (2.12) gives a worst case condition on the switching gain K_s. However, in many practical applications it is not desirable to have a high amplitude, high frequency, control signal. This might in fact lead to excessive wear in mechanical structures and high currents in electrical circuits. Therefore, we seek a control structure where the condition (2.12) can be made less conservative, which can be achieved if, besides $\theta(t)$ and $\dot{\theta}(t)$, also the parameters of the pendulum are known. In this case an estimate of the equivalent control can be made. We therefore define the control law as:

$$u(t) = u_c(t) + u_d(t) \tag{2.13}$$

where the subscripts c and d stand for continuous and discontinuous respectively. The structure of the discontinuous control part is identical to control law (2.13) and the continuous control part is taken as the equivalent control, i.e.:

$$u_d(t) = -K_s\text{sign}(\sigma(t)) \tag{2.14}$$

$$u_c(t) = \frac{ml^2}{g}\sin(\theta(t)) + \frac{ml^2}{S_2}\left(S_2c_f - S_1\right)\dot{\theta}(t) \tag{2.15}$$

It is noted however, that the continuous control component is not the same as the equivalent control. Despite the fact that equation (2.15) is identical to equation (2.6), their interpretation is different. Where the equivalent control gives a continuous interpretation of the *total control signal* (i.e. $u_c(t) + u_d(t)$), equation (2.15)

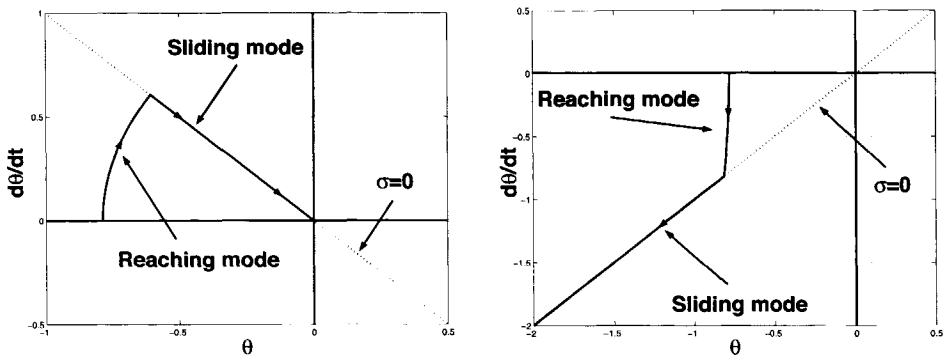


Fig. 2.3: (a) Phase plot for the pendulum with $S = [1 \ 1]$. (b) Phase plot for the pendulum with $S = [1 \ -1]$.

should be seen as an approximation of the required signal to maintain sliding mode. Only in the case of a perfect model, perfect measurements and no disturbances, a perfect estimate of the required equivalent control can be made. In all other case, the discontinuous control term (equation (2.14)) is still required to drive the system into sliding mode.
Again using reaching condition (2.9) to determine the condition on the switching gain results in the following:

$$K_s \geq \eta \tag{2.16}$$

The value of η is defined by the control engineer. It specifies the speed of convergence of the closed-loop system to $\sigma(t) = 0$. Therefore, to combine a small switching gain with fast convergence, the discontinuous control term could be extended with a proportional feedback term as:

$$u_d(t) = -K_s \text{sign}(\sigma(t)) - K_p \sigma(t) \tag{2.17}$$

with some strictly positive constant $K_p \in \mathbb{R}$. It can be determined that reaching condition (2.9) is satisfied for the control law $u(t) = u_c(t) + u_d(t)$ if:

$$K_s > \eta \tag{2.18}$$

Of course, this is identical to the case without a proportional feedback term in the discontinuous control part. However, the speed of convergence towards $\sigma(t) = 0$ is increased considerably.
So far we have concentrated on the stability of the newly defined output $\sigma(t)$, another important issue is of course the internal dynamics of the closed-loop system. To illustrate these dynamics, Figure 2.3 represents the phase plane plot for the obtained simulation results. Figure 2.3.a depicts the phase plane plot for $S = [1 \ 1]$ and $K_s = 1$. As can be seen in the figure, two separate modes can be identified. The first mode, the *reaching mode*, represents the initial phase where the switching

function is not equal to zero. However, as can be seen in the figure, the system is moving in the direction of $\sigma(t) = 0$. In the second mode, the *sliding mode*, the switching function is zero and remains zero. While $\sigma(t) = 0$, the zero dynamics associated with the newly defined output $\sigma(t)$ can be seen as the movement along the line $\sigma(t) = 0$, in this case towards the origin. This explains the name sliding mode, the closed-loop system slides along the switching function. The line $\sigma(t) = 0$ is therefore named the *sliding surface.*

Figure 2.3.b shows the phase plane plot for the case where $S = [1 \quad -1]$. The same modes as for the previous case can be identified. The sliding surface is reached in finite time (reaching mode). After reaching the sliding surface, the closed-loop system remains in sliding mode. Unlike the case for $S = [1 \quad 1]$, the system is now moving away from the origin. Apparently the zero dynamics associated with the newly defined output $\sigma(t)$ are unstable. These dynamics can be easily obtained from $\sigma(t) = S_1\theta(t) + S_2\dot{\theta}(t) = 0$:

$$\dot{\theta}(t) = -\frac{S_1}{S_2}\theta(t) \tag{2.19}$$

From the above equation it can easily be seen that the internal dynamics in the sliding mode are stable if $\frac{S_1}{S_2} > 0$. This condition is not met for $S = [1 \quad -1]$ which resulted in the unstable behavior along the sliding surface. Another important implication of the above given dynamics is that they do not depend on any system parameters at all. The dynamics in the sliding mode are purely specified by the ratio of S_1 and S_2. This property is known as *invariance* since the dynamics in the sliding mode are invariant to any system parameters and consequently the dynamics in the sliding mode are invariant against parameter variations as well.

In the following sections, a more formal introduction of sliding mode control is given. In Section 2.1 the state-based, continuous-time, sliding mode controller (SCSMC) is presented, Section 2.2 is devoted to the output-based, continuous-time, sliding mode controller (OCSMC).

2.1 State-Based Continuous-Time Sliding Mode Control

Several design procedures exist to obtain a CSMC for linear systems. Roughly these methods can be divided into the transfer function based method and the state space based method. This section is focused on the latter method, for which the following system is considered:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.20}$$

with $x \in \mathbb{R}^n$ being the system state, $u \in \mathbb{R}^m$ being the system input, and the matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. The matrix B is assumed to have full rank and the pair (A, B) is assumed to be controllable. The design procedure for a state-based continuous-time sliding mode controller (SCSMC) can now be divided into two parts:

- **Step 1.** Finding the matrix $S \in \mathbb{R}^{m\times n}$ which defines the switching function:

$$\sigma_x(t) = Sx(t) \tag{2.21}$$

($\sigma_x \in \mathbb{R}^m$) such that the internal dynamics in the sliding mode are stable. The subscript $_x$ is used to indicate that the switching function is based on the system state.

- **Step 2.** Designing a controller which ensures that the sliding mode is reached and subsequently maintained.

2.1.1 State-Based Sliding Surface design

Here we introduce a design procedure for Step 1, i.e. finding a switching function with stable zero-dynamics. It is well known that for the controllable system (2.20) there exists an invertible transformation $T_{r_x} \in \mathbb{R}^{n\times n}$, defined as:

$$\begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \end{bmatrix} = T_{r_x} x(t) \tag{2.22}$$

which brings the system into the so called *regular form* (Utkin 1992):

$$\dot{\bar{x}}_1(t) = \bar{A}_{11}\bar{x}_1(t) + \bar{A}_{12}\bar{x}_2(t) \tag{2.23}$$

$$\dot{\bar{x}}_2(t) = \bar{A}_{21}\bar{x}_1(t) + \bar{A}_{22}\bar{x}_2(t) + \bar{B}_2 u(t) \tag{2.24}$$

where $\bar{x}_1 \in \mathbb{R}^{n-m}$, $\bar{x}_2 \in \mathbb{R}^m$, $\bar{A}_{11} \in \mathbb{R}^{(n-m)\times(n-m)}$, $\bar{A}_{12} \in \mathbb{R}^{(n-m)\times m}$, $\bar{A}_{21} \in \mathbb{R}^{m\times(n-m)}$, $\bar{A}_{22} \in \mathbb{R}^{m\times m}$, and $\bar{B}_2 \in \mathbb{R}^{m\times m}$. Furthermore, $\bar{B}_2$ has full rank. In these new coordinates the switching function (2.21) becomes:

$$\sigma_x(t) = \bar{S}_1\bar{x}_1(t) + \bar{S}_2\bar{x}_2(t) \tag{2.25}$$

with $\bar{S}_1 \in \mathbb{R}^{m\times(n-m)}$ and $\bar{S}_2 \in \mathbb{R}^{m\times m}$. The matrices $\bar{S}_1$ and $\bar{S}_2$ are the design parameters which define the sliding surface, they should be chosen such that in the case that $\sigma_x(t) = 0$ all remaining dynamics are stable. In other words, the matrices $\bar{S}_1$ and $\bar{S}_2$ should be designed such that the zero-dynamics of $\sigma_x(t)$ are stable. Assuming for the moment that there exists a controller which forces the system into sliding mode (i.e. $\sigma_x(t) = 0$), we can find (under the restriction that we choose $\bar{S}_2$ invertible) from equation (2.25):

$$\bar{x}_2(t) = -\bar{S}_2^{-1}\bar{S}_1\bar{x}_1(t) \tag{2.26}$$

For simplicity we use the variable $M = \bar{S}_2^{-1}\bar{S}_1$. Substituting the above equation into equation (2.23) results in:

$$\dot{\bar{x}}_1(t) = \underbrace{\left(\bar{A}_{11} - \bar{A}_{12}M\right)}_{=A_{sm}}\bar{x}_1(t) \tag{2.27}$$

The above equation describes all dynamics of the closed-loop system in sliding mode. Therefore, stability in the sliding mode is ensured when all eigenvalues of the matrix $\left(\bar{A}_{11} - \bar{A}_{12}M\right)$ are within the unit circle. The problem of finding the design matrix M is in fact a classical state feedback problem. It can be found that if the pair (A, B) is controllable, then the pair $(\bar{A}_{11}, \bar{A}_{12})$ is controllable as well (Edwards and

Spurgeon 1998). Therefore the matrix M can be determined by for example pole pole placement or LQR-design.
Remembering that state component $\bar{x}_2(t)$ can be found from $\bar{x}_1(t)$ by the algebraic relation (2.26), stability of $\bar{x}_1(t)$ implies stability of the closed-loop system in sliding mode. Furthermore it follows that the dynamics in the sliding mode depend on the matrix M. Hence, the matrix $\bar{S}_2$ only acts as a scaling factor and can be chosen arbitrarily as long as it is invertible. $\bar{S}_2$ can therefore be chosen as $\bar{S}_2 = \bar{B}_2^{-1}$. In this way, each entry of the switching function is coupled to one entry of the input vector. The matrix S in the original coordinates (see transformation (2.22)) can then be found from:

$$S = \bar{S}_2[M \quad I_m]T_{r_x} \tag{2.28}$$

Summarizing we have redefined the original control problem (2.20) as:

$$\begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ \sigma_x(t) & = & Sx(t) \end{array} \tag{2.29}$$

where S is designed in such a way that the zero dynamics corresponding to $\sigma_x(t)$ are stable. The remaining step (Step 2) is to stabilize $\sigma_x(t)$ at zero. In the next section a controller is presented which stabilizes $\sigma_x(t)$ at zero.

2.1.2 State-Based Controller Implementation

Now we continue with design step 2, which aims at stabilizing $\sigma_x(t)$ at zero. Let us now assume that the system is subject to a disturbance and/or modeling error $f(x,t) \in \mathbb{R}^n$. The system dynamics are then given by:

$$\dot{x}(t) = Ax(t) + Bu(t) + f(x,t) \tag{2.30}$$

In order to find a controller which ensures sliding mode of the above system we define the transformation matrix $T_\sigma \in \mathbb{R}^{n \times n}$ by:

$$T_\sigma = \begin{bmatrix} I & 0 \\ \bar{S}_1 & \bar{S}_2 \end{bmatrix} \tag{2.31}$$

where the matrices $\bar{S}_1$ and $\bar{S}_2$ have been determined in the previous section. The above transformation matrix is invertible since the matrix $\bar{S}_2$ was invertible by design choice. Then applying the transformation:

$$\begin{bmatrix} \bar{x}_1(t) \\ \sigma_x(t) \end{bmatrix} = T_\sigma T_{r_x} x(t) \tag{2.32}$$

(with transformation T_{r_x} as defined in (2.22)) brings the system in the following form:

$$\dot{\bar{x}}_1(t) = \tilde{A}_{11}\bar{x}_1(t) + \tilde{A}_{12}\sigma_x(t) + f_u(x,t) \tag{2.33}$$

$$\dot{\sigma}_x(t) = \tilde{A}_{21}\bar{x}_1(t) + \tilde{A}_{22}\sigma_x(t) + u(t) + f_m(x,t) \tag{2.34}$$

where $\tilde{A}_{11} \in \mathbb{R}^{(n-m)\times(n-m)}$, $\tilde{A}_{12} \in \mathbb{R}^{(n-m)\times m}$, $\tilde{A}_{21} \in \mathbb{R}^{m\times(n-m)}$, $\tilde{A}_{22} \in \mathbb{R}^{m\times m}$, $[f_u^T(x,t)\ f_m^T(x,t)]^T = T_\sigma T_{r_x} f(x,t)$, $f_u(x,t) \in \mathbb{R}^{n-m}$, $f_m(x,t) \in \mathbb{R}^m$, and (by design choice) $\bar{S}_2\bar{B}_2 = I_m$. The subscript m of $f_m(x,t)$ stems from the word *matched* since it is a matched uncertainty. As can be seen in the above equations, the term $f_m(x,t)$ appears in the same line as the input and hence can be cancelled out directly by the input. The subscript u of $f_u(x,t)$ on the other hand stems from the word *unmatched.*

The control law is chosen similar to (2.13) as:

$$u(t) = u_c(t) + u_d(t) \tag{2.35}$$

where the continuous control component u_c and the discontinuous control component u_d are given by:

$$u_c(t) = -\tilde{A}_{21}\bar{x}_1(t) - \tilde{A}_{22}\sigma_x(t) \tag{2.36}$$

$$u_d(t) = -K_s\frac{\sigma_x(t)}{\|\sigma_x(t)\|} - K_p\sigma_x(t) \tag{2.37}$$

To study stability the following Lyapunov function is defined:

$$V(\sigma_x) = \frac{1}{2}\sigma_x(t)^T\sigma_x(t) \tag{2.38}$$

to which the reaching law should apply:

$$\dot{V}(\sigma_x) = \sigma_x^T(t)\dot{\sigma}_x(t) \leq -\eta\|\sigma_x(t)\| \tag{2.39}$$

for some strictly positive scalar η. Substituting control laws (2.35), (2.36), (2.37) into equation (2.34) results in:

$$\dot{\sigma}_x(t) = -K_s\frac{\sigma_x(t)}{\|\sigma(t)\|} - K_p\sigma_x(t) + f_m(x,t) \tag{2.40}$$

Combining equation (2.40) with the reaching law (2.39) results in:

$$-K_s\frac{\sigma_x^T(t)\sigma_x(t)}{\|\sigma(t)\|} - K_p\sigma_x^T(t)\sigma_x(t) + \sigma_x^T(t)f_m(x,t) \leq -\eta\|\sigma_x(t)\| \tag{2.41}$$

Since $\|\sigma_x(t)\|^2 = \sigma_x^T(t)\sigma_x(t)$ and $\sigma_x^T(t)f_m(x,t) \leq \|\sigma_x(t)\|\|f_m(x,t)\|$, condition (2.41) is always satisfied if:

$$K_s + K_p\|\sigma_x(t)\| \geq \eta + \|f_m(x,t)\| \tag{2.42}$$

Let us assume that we have some knowledge of the disturbance, for example:

$$\|f(x,t)\| \leq k_1\|x\| + k_2 \tag{2.43}$$

with $0 < k_1 \in \mathbb{R}$ and $0 < k_2 \in \mathbb{R}$. Then applying the transformations T_σ and T_{r_x} and realizing that $\|x\| \leq c_1\|\bar{x}_1\| + c_2\|\sigma_x\|$, for some $0 < c_1 \in \mathbb{R}$ and $0 < c_2 \in \mathbb{R}$, leads to:

$$\|f_m(x,t)\| \leq k_{m,\bar{x}_1}\|\bar{x}_1\| + k_{m,\sigma_x}\|\sigma_x\| + k_{m,2} \tag{2.44}$$

with some $k_{m,\bar{x}_1} < c_1 \in \mathbb{R}$, $0 < k_{m,\sigma_x} \in \mathbb{R}$, $k_{m,2} < c_1 \in \mathbb{R}$. Substituting the previous equation into equation (2.42) results in the following condition:

$$K_s + K_p\|\sigma_x(t)\| \geq \eta + k_{m,\bar{x}_1}\|\bar{x}_1\| + k_{m,\sigma_x}\|\sigma_x\| + k_{m,2} \tag{2.45}$$

leading to the conditions:

$$K_p \geq k_{m,\sigma_x} \tag{2.46}$$

$$K_s \geq \eta + k_{m,\bar{x}_1}\|\bar{x}_1\| + k_{m,2} \tag{2.47}$$

If the above conditions are met, then the closed-loop system will reach the sliding mode in finite time. The dynamics in the sliding mode are represented by the *reduced order dynamics*:

$$\dot{\bar{x}}_1(t) = \tilde{A}_{11}\bar{x}_1(t) + f_u(x,t) \tag{2.48}$$

Equation (2.48) describes all dynamics in the sliding mode, the state component can be obtained by the algebraic relation $\bar{x}_2(t) = -\bar{S}_2^{-1}\bar{S}_1\bar{x}_1(t)$ and therefore does not contain any independent dynamics. Therefore, the order of the dynamics is given by the order of equation (2.48), which is $(n-m)$. In general it can be concluded that the dynamics in the sliding mode have order $(n-m)$ instead of the original order n.
From equation (2.48) it can be concluded as well that once the system is in the sliding mode, it is invariant against matched uncertainties. The dynamics of the closed-loop system are fully described by equation (2.48) which are only driven by the unmatched uncertainties. By the boundedness of $f(x,t)$ (condition (2.43)), and hence the boundedness of $f_u(x,t)$, it follows that $\bar{x}_1(t)$ remains bounded as well. Furthermore, in the special case where $f_u(x,t) = 0$, the closed-loop system will converge towards the origin of the state-space along the sliding surface according to the dynamics:

$$\dot{\bar{x}}_1(t) = \tilde{A}\bar{x}_1(t) \tag{2.49}$$

despite any matched uncertainties.

2.2 Output-Based Continuous-Time Sliding Mode Control

This section is focused on the design of a CSMC which only requires information of the (measured) outputs, whereas the previous section assumed that there was full knowledge of the systems' state. Of course, the system's state can be reconstructed by the use of an observer in which case the state-based CSMC can be used. However, in this section a different approach is considered. Instead of defining the switching function on the full system state, we now only base the switching function on the available outputs. Also the controller which enforces the sliding mode will be defined in terms of the outputs only, thereby omitting the need for an observer.
The design procedure for an OCSMC is quite similar to a SCSMC. However, the limited availability of the system state (only measured outputs are to be used) poses the designer to some restrictions. The most serious restriction for the design

of an OCSMC is that it is limited to the class of minimum phase systems. For nonminimum phase systems, it is not possible to construct an output-based sliding surface which results in a stable behavior in the sliding mode. The reason for this is that the systems' zeros are part of the poles in the sliding mode. Another problem encountered in OCSMC is that it is no longer possible to make an estimate of the equivalent control term. The final limitation of OCSMC is the fact that it is only applicable to systems where the matrix product CB is invertible . Extensions to systems with a higher relative degree could be made by the use of higher order sliding mode control, for which we refer to for example (Levant 1997).
We consider the system:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.50}$$
$$y(t) = Cx(t) \tag{2.51}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^p$, $u \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$. Furthermore, it is assumed that the triple (A, B, C) is both controllable and observable, and $\text{Rank}\{CB\} = m$. Similar to the design of the SCSMC, the design of an OCSMC consists of two steps:

- **Step 1.** Finding the matrix $S \in \mathbb{R}^{m\times n}$ which defines the switching function:

$$\sigma_y(t) = Sy(t) \tag{2.52}$$

  such that the internal dynamics in sliding mode are stable. The subscript $_y$ is used to indicate that the switching function is based on the system outputs, in contrast to SCSMC of Section 2.1.

- **Step 2.** Designing a controller which ensures that sliding mode is reached and subsequently maintained.

2.2.1 Output-Based Sliding Surface design

Here, the design procedure for Step 1, introduced in (Edwards and Spurgeon 1998), for an output-based sliding surface for the system (2.50) is presented. By the nonsingular transformation

$$\begin{bmatrix} \bar{x}_0(t) \\ \bar{x}_1(t) \\ \bar{y}_1(t) \\ \bar{y}_2(t) \end{bmatrix} = T_{r_y} x(t) \tag{2.53}$$

(with $T_{r_y} \in \mathbb{R}^{n\times n}$) we can bring the system (2.50) into the following form:

$$\begin{bmatrix} \dot{\bar{x}}_0(t) \\ \dot{\bar{x}}_1(t) \\ \dot{\bar{y}}_1(t) \\ \dot{\bar{y}}_2(t) \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} & \bar{A}_{14} \\ 0 & \bar{A}_{22} & \bar{A}_{23} & \bar{A}_{24} \\ 0 & \bar{A}_{32} & \bar{A}_{33} & \bar{A}_{34} \\ \bar{A}_{41} & \bar{A}_{42} & \bar{A}_{43} & \bar{A}_{44} \end{bmatrix} \begin{bmatrix} \bar{x}_0(t) \\ \bar{x}_1(t) \\ \bar{y}_1(t) \\ \bar{y}_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \bar{B}_2 \end{bmatrix} u(t) \tag{2.54}$$

$$y(t) = [0 \;\; T_y] \begin{bmatrix} \bar{x}_0(t) \\ \bar{x}_1(t) \\ \bar{y}_1(t) \\ \bar{y}_2(t) \end{bmatrix} \tag{2.55}$$

where $\bar{x}_0 \in \mathbb{R}^r$, $\bar{x}_1 \in \mathbb{R}^{n-p-r}$, $\bar{y}_1 \in \mathbb{R}^{p-m}$, $\bar{y}_2 \in \mathbb{R}^m$, all matrices $\bar{A}_{ij}$ $\forall\, i,j = 1 \ldots 4$ of appropriate size, and the matrix $\bar{B}_2 \in \mathbb{R}^{m \times m}$ has full rank. The transformation matrix T_{r_y} has been chosen such that the partition $\bar{A}_{11}$ contains the stable invariant zeros. The switching function (2.52) in the new coordinates is given by:

$$\sigma_y(t) = \bar{S}_1 \bar{y}_1(t) + \bar{S}_2 \bar{y}_2(t) \tag{2.56}$$

where we choose $\bar{S}_2$ invertible. Once the system is in the sliding mode (i.e. $\sigma_y(t) = 0$), we can write for $\bar{y}_2(t)$ from the above equation:

$$\bar{y}_2(t) = -\bar{S}_2^{-1} \bar{S}_1 \bar{y}_1(t) \tag{2.57}$$

For simplicity the matrix $M = \bar{S}_2^{-1} \bar{S}_1$ is introduced. Substituting the above equation into equation (2.54) leads to the dynamics in the sliding mode:

$$\begin{bmatrix} \dot{\bar{x}}_0(t) \\ \dot{\bar{x}}_1(t) \\ \dot{\bar{y}}_1(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & (\bar{A}_{13} - \bar{A}_{14}M) \\ 0 & \bar{A}_{22} & (\bar{A}_{23} - \bar{A}_{24}M) \\ 0 & \bar{A}_{32} & (\bar{A}_{33} - \bar{A}_{34}M) \end{bmatrix}}_{=A_{sm}} \begin{bmatrix} \bar{x}_0(t) \\ \bar{x}_1(t) \\ \bar{y}_1(t) \end{bmatrix} \tag{2.58}$$

the above description clearly shows that the poles of the closed-loop system in the sliding mode are given by:

$$\lambda(A_{sm}) = \lambda(\bar{A}_{11}) \cup \lambda\left(\begin{bmatrix} \bar{A}_{22} & (\bar{A}_{23} - \bar{A}_{24}M) \\ \bar{A}_{32} & (\bar{A}_{33} - \bar{A}_{34}M) \end{bmatrix} \right) \tag{2.59}$$

It is well known that the matrix $\bar{A}_{11}$ contains the invariant zeros of the open-loop system. Since the location of these zeros cannot be altered by output feedback control, OCSMC is limited to the class of minimum phase systems. A system is called minimum phase if it has all its invariant zeros in the left half section of the complex plane. Assuming that the system is minimum phase, the problem of designing a stable output-based sliding surface reduces to placing the eigenvalues of the following matrix into the left half plane:

$$\begin{bmatrix} \bar{A}_{22} & (\bar{A}_{23} - \bar{A}_{24}M) \\ \bar{A}_{32} & (\bar{A}_{33} - \bar{A}_{34}M) \end{bmatrix} \tag{2.60}$$

The problem of finding the matrix M is equivalent to solving the classical static output feedback problem $u(t) = -My(t)$ for the system $(\bar{A}_o, \bar{B}_o, \bar{C}_o)$, where the matrices are defined by:

$$\bar{A}_o = \begin{bmatrix} \bar{A}_{22} & \bar{A}_{23} \\ \bar{A}_{32} & \bar{A}_{33} \end{bmatrix} \quad \bar{B}_o = \begin{bmatrix} \bar{A}_{24} \\ \bar{A}_{34} \end{bmatrix} \quad \bar{C}_o = [0 \;\; I_m] \tag{2.61}$$

A matrix M, which stabilizes the triple $(\bar{A}_o, \bar{B}_o, \bar{C}_o)$, can be found if the so-called Kimura-Davison condition holds (Edwards and Spurgeon 1998). However, since the triple $(\bar{A}_o, \bar{B}_o, \bar{C}_o)$ is stabilizable and detectable (which follows from the fact that the triple (A, B, C) is controllable and observable), the solution can be found by adding more dynamics to the closed-loop system for which we refer to (Edwards and Spurgeon 1998). For a rigorous method to obtain the matrix M we refer to (Edwards and Spurgeon 2001).
Like the state-based case the choice of $\bar{S}_2$ is arbitrary as long as it is invertible. For simplicity we choose $\bar{S}_2 = \bar{B}_2^{-1}$. The matrix S can then be found from:

$$S = \bar{S}_2 \begin{bmatrix} M & I_m \end{bmatrix} T_y^{-1} \tag{2.62}$$

The above design procedure has redefined the control problem as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ \sigma_y(t) &= SCx(t) \end{aligned} \tag{2.63}$$

where S was defined in such a way that the zero dynamics corresponding to the output $\sigma_y(t)$ are stable.

2.2.2 Output-Based Controller Implementation

Now we continue with Step 2, i.e. the stabilization of the switching function. Just as for the state-based controller implementation (Section 2.1.2) let us assume that the system is subject to the disturbance $f(x,t) \in \mathbb{R}^n$. The system dynamics are then given by:

$$\dot{x}(t) = Ax(t) + Bu(t) + f(x,t) \tag{2.64}$$

$$y(t) = Cx(t) \tag{2.65}$$

Successively applying the transformation T_{r_y} (bringing the above system in the same form as equation (2.54)) and the transformation $T_\sigma \in \mathbb{R}^{n \times n}$ defined by:

$$T_\sigma = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & \bar{S}_1 & \bar{S}_2 \end{bmatrix} \tag{2.66}$$

(T_σ is nonsingular since $\bar{S}_2$ has full rank) we can bring system (2.64) into the following form:

$$\begin{aligned} \begin{bmatrix} \dot{\bar{x}}_0(t) \\ \dot{\bar{x}}_1(t) \\ \dot{\bar{y}}_1(t) \\ \dot{\sigma}_y(t) \end{bmatrix} &= \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} & \tilde{A}_{14} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} & \tilde{A}_{34} \\ \tilde{A}_{41} & \tilde{A}_{42} & \tilde{A}_{43} & \tilde{A}_{44} \end{bmatrix} \begin{bmatrix} \bar{x}_0(t) \\ \bar{x}_1(t) \\ \bar{y}_1(t) \\ \sigma_y(t) \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_m \end{bmatrix} u(t) + \begin{bmatrix} f_{\bar{x}_0}(x,t) \\ f_{\bar{x}_1}(x,t) \\ f_{\bar{y}_1}(x,t) \\ f_m(x,t) \end{bmatrix} \end{aligned} \tag{2.67}$$

where it was used that by design choice $\bar{S}_2\bar{B}_2 = I_m$. By assumption the partition $\tilde{A}_{11}$ is stable, by design the partition $[\tilde{A}_{22} \quad \tilde{A}_{23}; \tilde{A}_{32} \quad \tilde{A}_{33}]$ is stable. The vectors $f_{\bar{x}_0}(x,t)$, $f_{\bar{x}_1}(x,t)$, and $f_{\bar{y}_1}(x,t)$ represent the unmatched disturbances. The vector $f_m(x,t)$ represents the matched disturbance.
Similar to the SCSMC we construct a control law which consists of a continuous control component $u_c(t)$ and a discontinuous control component $u_d(t)$, i.e.:

$$u(t) = u_c(t) + u_d(t) \tag{2.68}$$

where:

$$u_c(t) = -\tilde{A}_{43}\bar{y}_1(t) - \tilde{A}_{44}\sigma_y(t) \tag{2.69}$$

$$u_d(t) = -K_s\frac{\sigma_y(t)}{\|\sigma_y(t)\|} - K_p\sigma_y(t) \tag{2.70}$$

Now the Lyapunov function defined by $V(\sigma_y) = \frac{1}{2}\sigma_y^T(t)\sigma_y(t)$ should satisfy the reaching law :

$$\sigma_y^T(t)\dot{\sigma}_y(t) \leq -\eta\|\sigma_y(t)\| \tag{2.71}$$

for some strictly positive scalar η. Using equation (2.67) and the control equations (2.68), (2.69), and (2.68) we can rewrite the above reaching law to:

$$\sigma_y^T(t)\left(\tilde{A}_{41}\bar{x}_0(t) + \tilde{A}_{42}\bar{x}_1(t) - K_s\frac{\sigma_y(t)}{\|\sigma_y(t)\|} - K_p\sigma_y(t) + f_m(x,t)\right) \leq -\eta\|\sigma_y(t)\| \tag{2.72}$$

Since:

$$\sigma_y^T(t)f_m(x,t) \leq \|\sigma_y(t)\|\|f_m(x,t)\| \tag{2.73}$$

$$\sigma_y^T(t)\tilde{A}_{41}\bar{x}_0(t) \leq \|\sigma_y(t)\|\|\tilde{A}_{41}\bar{x}_0(t)\| \tag{2.74}$$

$$\sigma_y^T(t)\tilde{A}_{42}\bar{x}_1(t) \leq \|\sigma_y(t)\|\|\tilde{A}_{42}\bar{x}_1(t)\| \tag{2.75}$$

condition (2.72) is satisfied if:

$$K_s + K_p\|\sigma_y(t)\| \geq \eta + \|f_m(x,t)\| + \|\tilde{A}_{41}\bar{x}_0(t)\| + \|\tilde{A}_{42}\bar{x}_1(t)\| \tag{2.76}$$

Writing the bound on the disturbance vector $\|f_m(x,t)\| \leq k_1\|x(t)\| + k_2$ into the new coordinates as:

$$\|f_m(x,t)\| \leq k_{m,xy}\left\|[\bar{x}_0(t) \quad \bar{x}_1(t) \quad \bar{y}_1(t)]^T\right\| + k_{m,\sigma}\|\sigma_y(t)\| + k_{m,2} \tag{2.77}$$

this results in the following conditions on the gains K_s and K_p:

$$K_s \geq \eta + k_{m,xy}\left\|[\bar{x}_0(t) \quad \bar{x}_1(t) \quad \bar{y}_1(t)]^T\right\| + k_{m,2} \tag{2.78}$$

$$K_p \geq k_{m,\sigma} \tag{2.79}$$

If the above conditions are met, then the closed-loop system will reach sliding mode in finite time. The dynamics in sliding mode are represented by:

$$\begin{bmatrix} \dot{\bar{x}}_0(t) \\ \dot{\bar{x}}_1(t) \\ \dot{\bar{y}}_1(t) \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} \end{bmatrix} \begin{bmatrix} \bar{x}_0(t) \\ \bar{x}_1(t) \\ \bar{y}_1(t) \end{bmatrix} + \begin{bmatrix} f_{\bar{x}_0}(x,t) \\ f_{\bar{x}_1}(x,t) \\ f_{\bar{y}_1}(x,t) \end{bmatrix} \tag{2.80}$$

From the above equation it can be concluded that once the system is in the sliding mode, it is invariant against matched uncertainties. The dynamics of the closed-loop system are fully described by equation (2.80) which are only driven by the unmatched uncertainties represented by $f_{\bar{x}_0}(x,t)$, $f_{\bar{x}_1}(x,t)$, and $f_{\bar{y}_1}(x,t)$. By the boundedness of $f(x,t)$ (condition (2.43)), and hence the boundedness of the unmatched uncertainty components ($f_{\bar{x}_0}(x,t)$, $f_{\bar{x}_1}(x,t)$, and $f_{\bar{y}_1}(x,t)$), it follows that $\bar{x}_0(t)$, $\bar{x}_1(t)$, and $\bar{y}_1(t)$ remain bounded as well. Another important issue to mention is that the invariant zeros of the system are still present in reduced order dynamics given by equation (2.80). These zeros correspond to the eigenvalues of $\tilde{A}_{11}$.

2.3 Practical Considerations

The strong properties of CSMC, such as invariance against matched uncertainties, are the result of the high frequency switching term. While in some applications such switching is inherent (for example switched networks such as electrical converters (Utkin et al. 1999)), in other applications the high frequency switching is undesirable. First of all the actuator bandwidth is often limited, therefore an infinite switching frequency cannot be attained. Furthermore, the high frequency input signal leads to strong current peaks in electrical actuators and high wear in mechanical gear boxes making the high frequency signal undesirable from the system's point of view. In addition the high frequency signal might excite unmodeled (fast) dynamics making it an undesirable signal from the control engineer's point of view as well. The above effects will result in a chattering behavior along the switching surface which is illustrated in Figure 2.4. In principle chattering can be reduced by:

- Reducing the frequency of the switching term.
- Reducing the amplitude of the switching term.

Reducing the frequency can be accomplished in several ways. The most straightforward way would be to filter the control signal in a low pass filter in order to protect the system against the high frequency control actions. However, while limiting the frequency of the applied control signal, this could lead to a chattering behaviour around the sliding surface. Better alternatives are creating a dynamic reaching condition (Sira-Ramírez 1993, Bartolini et al. 1998) which ensures smooth reaching of the switching surface, or using an observer to estimate the equivalent control (Bondarev et al. 1985). Because of its simplicity, we limit ourselves to the approach which uses a continuous approximation of the switching term, which can be found in Section 2.3.1.

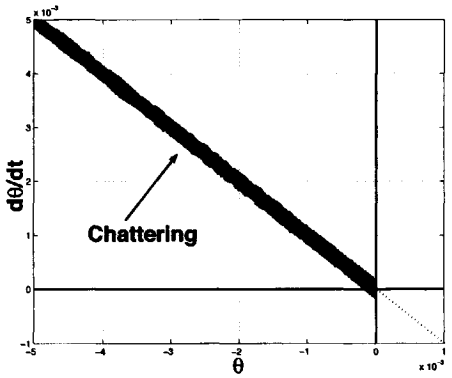


Fig. 2.4: Chattering behavior around the switching surface.

The other way of reducing chattering is to decrease the amplitude of the switching term. Comparing conditions (2.12) and (2.18) it can be immediately seen that the introduction of the continuous control $u_c(t)$ (equation (2.15)) term and the proportional feedback term $-K_p\sigma(t)$ (equation (2.17)) reduced the condition on the switching gain considerably. In Section 2.3.2 a method is given which adapts the switching gain to the actual conditions, without any knowledge of the disturbance. In the following sections the SCSMC of Section 2.1 is considered. It can be determined with equations (2.34), (2.35), (2.36), (2.39), and (2.37) that the sliding mode is reached in finite time if there exists some strictly positive constant η satisfying:

$$\sigma_x^T(t)\left(u_d(t) + f_m(x,t)\right) \leq -\eta\|\sigma_x(t)\| \tag{2.81}$$

The above condition is used in the next sections to study the stability of the introduced discontinuous control part.

2.3.1 Softened Discontinuous Control

The method of softening the discontinuous control part by a continuous approximation is very attractive for its simplicity. Two possible candidates are the ("standard") saturation function (Slotine 1984, Slotine and Li 1991) and the saturation function with continuous derivative (Edwards and Spurgeon 1998). The discontinuous control term with saturation, or with continuous derivative saturation, are successively defined in the following two definitions.

Definition 2.3.1 *The discontinuous control term with saturation is defined by:*

$$u_d(t) = -K_s sat\left(\frac{\sigma(t)}{\delta}\right) = \begin{cases} -K_s\frac{\sigma(t)}{\|\sigma(t)\|} & \|\sigma(t)\| \geq \delta \\ -K_s\frac{\sigma(t)}{\delta} & \|\sigma(t)\| < \delta \end{cases} \quad 9 \tag{2.82}$$

for some, preferably small, $\delta > 0$.

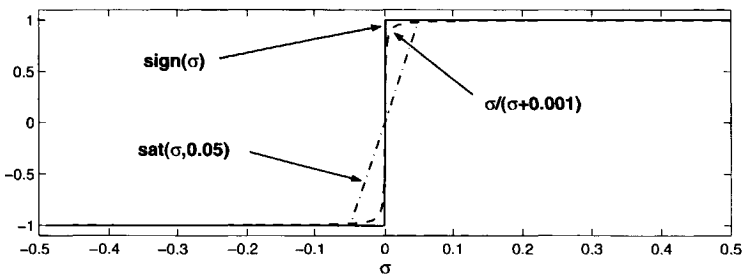


Fig. 2.5: Continuous approximations of the sign-function.

Definition 2.3.2 *The discontinuous control term with continuous derivative saturation is defined by:*

$$u_d(t) = -K_s \frac{\sigma(t)}{\|\sigma(t)\| + \delta} \tag{2.83}$$

for some, preferably small, $\delta > 0$.

Despite the fact that both definitions define the so-called discontinuous control term, they are no longer discontinuous. However, because they perform the role of the traditional discontinuous control term we maintain the terminology. Both approximations are illustrated for the single input case in Figure 2.5. Using the above approximations, the reaching law will still be satisfied for $\|\sigma(t)\| > \Delta$ with some, preferably small positive $\Delta \in \mathbb{R}$. Once the system is driven into the region $\sigma(t) \leq \Delta$ the systems is said to be in a *quasi sliding mode* or *pseudo sliding mode.* The region $\sigma(t) \leq \Delta$ is called the *quasi sliding mode band* or *pseudo sliding mode band.* The value of Δ is determined by the parameters δ and K_s.
The following two lemma's present the quasi sliding mode band for the saturated discontinuous control part and the continuous derivative saturated discontinuous control part as defined in Definitions 2.3.1 and 2.3.2 respectively, for a SCSMC.

Lemma 2.3.3 *The closed-loop system formed by system (2.20) with controller (2.35), (2.36) and the discontinuous control part defined in Definition 2.3.1 will converge to the quasi sliding mode band with* Δ *given as:*

$$\Delta = \delta \frac{\hat{f}_m}{K_s} \tag{2.84}$$

if $K_s > \hat{f}_m$ *with* $\hat{f}_m = \max\{\|f_m(x,t)\|\}$.
Proof: *The proof of this lemma follows from condition (2.81). We will evaluate this condition for* $\sigma_x \geq \delta$ *and* $\sigma_x < \delta$ *independently to determine the region of* σ_x *for which the condition (2.81) is fulfilled.*

- *For* $\|\sigma_x(t)\| \geq \delta$ *substituting (2.82) into condition (2.81) results in:*

$$\sigma_x^T(t) \left(-K_s \frac{\sigma_x(t)}{\|\sigma_x(t)\|} + f_m(x,t) \right) \leq -\eta \|\sigma_x(t)\| \tag{2.85}$$

Condition (2.85) can be rearranged as:

$$K_s\|\sigma_x(t)\| \geq \eta\|\sigma_x(t)\| + \sigma_x^T(t)f_m(x,t) \tag{2.86}$$

Then, by the triangle inequality $\sigma_x^T(t)f_m(t,x) \leq \|\sigma_x(t)\|\hat{f}_m$ *the condition (2.86) is satisfied if:*

$$K_s \geq \eta + \hat{f}_m \tag{2.87}$$

The constant η *can be chosen arbitrarily small and by assumption* $K_s > \hat{f}_m$. *Hence, condition (2.87) is satisfied for sufficiently small* η.

▶ *For* $\|\sigma_x(t)\| < \delta$ *substituting (2.82) into condition (2.81) results in:*

$$\sigma_x^T(t)\left(-K_s\frac{\sigma_x(t)}{\delta} + f_m(x,t)\right) \leq -\eta\|\sigma_x(t)\| \tag{2.88}$$

While $\|\sigma_x(t)\| \neq 0$ *condition (2.88) is fulfilled, following the same motivation as for* $\|\sigma_x(t)\| \geq \delta$, *in:*

$$\|\sigma_x(t)\| \geq \frac{\delta}{K_s}\left(\eta + \hat{f}_m\right) \tag{2.89}$$

The parameter $\eta > 0$ *can be taken arbitrary small. Using this knowledge condition (2.89) can be reduced to:*

$$\|\sigma_x(t)\| > \delta\frac{\hat{f}_m}{K_s} \tag{2.90}$$

Since the reaching law (2.81) is fulfilled for $\|\sigma_x(t)\| > \delta\frac{\hat{f}_m}{K_s}$, *the closed-loop system will be driven into the quasi sliding mode band* $\Delta \geq \frac{\delta}{K_s}\bar{f}_m$. □

Lemma 2.3.4 *The closed-loop system of system (2.20) and controller (2.35), (2.36) and the discontinuous control part defined in Definition 2.3.2 will converge to the quasi sliding mode band with* Δ *given as:*

$$\Delta = \frac{\hat{f}_m}{K_s - \hat{f}_m}\delta \tag{2.91}$$

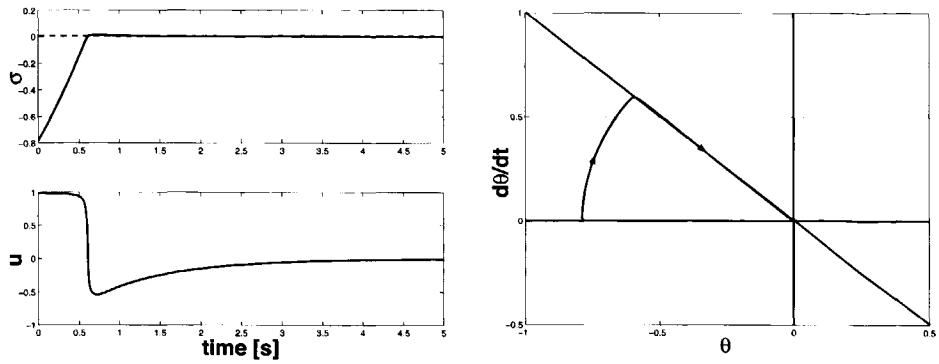
if $K_s > \hat{f}_m$ *with* $\hat{f}_m = \max\{\|f_m(x,t)\|\}$.

Proof: *The proof of this lemma follows from condition (2.81). We will evaluate this condition to determine the region of* σ_x *for which the condition (2.81) is fulfilled. Substituting (2.83) into condition (2.81) results in:*

$$\sigma_x^T(t)\left(-K_s\frac{\sigma_x(t)}{\|\sigma_x(t)\| + \delta} + f_m(x,t)\right) \leq -\eta\|\sigma_x(t)\| \tag{2.92}$$

Condition (2.92) can be rearranged as:

$$K_s\frac{\|\sigma_x(t)\|^2}{\|\sigma_x(t)\| + \delta} \geq \eta\|\sigma_x(t)\| - \sigma_x^T(t)f_m(x,t) \tag{2.93}$$



(a) (b)

Fig. 2.6: Forced pendulum example with discontinuous control part as defined in Definition 2.3.2. Figures (a) present the switching function (top) and the input signal (bottom), figure (b) presents the phase plane plot.

Then, assuming that $\|\sigma_x(t)\| \neq 0$, *by the triangle inequality* $\sigma_x^T(t) f_m(t,x) \leq \|\sigma_x(t)\| \hat{f}_m$ *condition (2.93) is satisfied if:*

$$\|\sigma_x(t)\| \geq \frac{\eta + \hat{f}_m}{K_s - \eta - \hat{f}_m}\, \delta \tag{2.94}$$

The parameter $\eta > 0$ *can be taken arbitrary small. Using this knowledge condition (2.94) reduces to:*

$$\|\sigma_x(t)\| > \frac{\hat{f}_m}{K_s - \hat{f}_m}\, \delta \tag{2.95}$$

Having shown that the reaching condition (2.81) is fulfilled for $\|\sigma_x(t)\| > \frac{\hat{f}_m}{K_s - \hat{f}_m}\, \delta$, *we know that the closed-loop system is driven into the quasi sliding mode band* $\Delta = \frac{\hat{f}_m}{K_s - \hat{f}_m}\delta$. □

The effect of the saturation function will be illustrated in the next example.

Example 2.3.5 *To illustrate the effect of the softened discontinuous control the input signal for the pendulum example at the beginning of this chapter is now defined as:*

$$u(t) = -K_s \frac{\sigma(t)}{|\sigma(t)| + \delta} \tag{2.96}$$

The simulation results are depicted in Figure 2.6 with $\delta = 1e^{-2}$. *Comparing the results to those presented at the beginning of this chapter , Figure 2.2.a, the improvement is remarkable. The plots of the switching function* σ *seem to be identical, while the control signal has changed drastically. The high frequency component has disappeared. Another effect of the smooth discontinuous control term can be seen by*

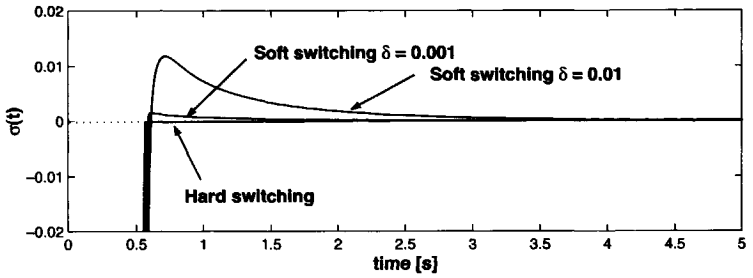


Fig. 2.7: Switching function close to $\sigma(t) = 0$ for the forced pendulum example with the discontinuous control as the hard switching sign function and the soft switching discontinuous control as defined in Definition 2.3.2.

zooming $\sigma(t)$ close to the sliding surface as presented in Figure 2.7. In Figure 2.7 it can be clearly seen that after reaching the sliding surface the closed-loop system does not stay on it. The reason for this phenomenon is that the reaching law is no longer satisfied within some boundary region around the sliding surface. This boundary region, which is described in Lemma 2.3.4, is called the quasi sliding mode band. Figure 2.7 also clearly shows that $\sigma(t)$ stays closer to zero for smaller values for δ, which is in correspondence with Lemma 2.3.4.

2.3.2 Adaptive Switching Gain

In the previous section the switching gain was softened to reduce chattering. In this section the case of a variable switching gain is considered, i.e. the discontinuous control part is now defined as:

$$u_d(t) = -K_s(t)\frac{\sigma_x(t)}{\|\sigma_x(t)\|} \tag{2.97}$$

Still stability of the closed-loop system is satisfied if condition (2.81) is satisfied. If we assume that the matched uncertainty vector $f_m(x,t)$ is bounded by:

$$\|f_m(x,t)\| < k_{m,1}\|x(t)\| + k_{m,2} \tag{2.98}$$

then it can easily be shown that condition (2.81) is satisfied if the following switching gain is selected:

$$K_s(t) = c_1\|x(t)\| + c_2 \tag{2.99}$$

with $c_1 \geq k_{m,1}$ and $c_2 \geq k_{m,2}$. For this solution some knowledge of the disturbance vector $f_m(x,t)$, and of course the system state x, is required to determine the coefficients c_1 and c_2. In cases where there is no a priori knowledge of the disturbance these coefficients should be determined by experimentation. Another way to obtain a variable switching gain is to make it adaptive. The switching gain is then optimized according to some criterion.

The most straightforward adaptation mechanism for the switching gain can be found in (Leung et al. 1991, Su et al. 1991, Wang and Fan 1993, Jiang et al. 1994, Roh and Oh 2000) and is defined as follows:

Definition 2.3.6 *The Method I adaptation law is defined as:*

$$\tilde{K}_s(t) = \tilde{K}_s(t_o) + \int_{\tau=t_o}^{t} \|\sigma(\tau)\| d\tau \tag{2.100}$$

This adaptation law is based on the fact that once the switching gain is sufficiently large, the system will be forced to the switching surface $\sigma(t) = 0$ in which case the switching gain will remain constant.

Lemma 2.3.7 *The closed-loop system formed by system represented by the equations (2.33) and (2.34), and controller (2.35), (2.36), (2.97) and switching gain as defined in Definition 2.3.6 will reach sliding mode in finite time. There exists some bounded $\tilde{K}_{s,max}$ such that:*

$$\hat{f}_m \leq \tilde{K}_s(\infty) < \tilde{K}_{s,max} \tag{2.101}$$

with $\hat{f}_m = \max\{f_m(x,t)\}$, $\int_{\tau=t_o}^{\infty} \max\{0, (\|f(x,t)\| - (1-\epsilon)\hat{f}_m)\} d\tau = \infty$ with $0 < \epsilon \in \mathbb{R}$ arbitrarily small and t_o being the initial time.

Proof: *For $\tilde{K}_s(t_o) > \hat{f}_m$ the closed-loop system will be driven in the sliding mode in finite time thereby bounding $\tilde{K}_s(t)$ from above. Together with the fact that the gain $\tilde{K}_s(t)$ is non-decreasing (2.101) immediately follows.*
For $\tilde{K}_s(t_o) \leq \hat{f}_m$ it is assumed, without loss of generality, that $\tilde{K}_s(t_o) = 0$. Then substituting equation (2.100) into condition (2.81) it follows that:

$$\sigma_x^T(t) \left(-\frac{\sigma_x(t)}{\|\sigma_x(t)\|} \int_{\tau=t_o}^{t} \|\sigma_x(\tau)\| d\tau + f_m(x,t) \right) \leq -\eta \|\sigma_x(t)\| \tag{2.102}$$

which, in the case of $\|\sigma_x(t)\| \neq 0$, can be rewritten as:

$$\|\sigma_x(t)\| \int_{\tau=t_o}^{t} \|\sigma_x(\tau)\| d\tau \geq \eta \|\sigma_x(t)\| + \sigma_x^T(t) f_m(x,t) \tag{2.103}$$

Using the property that $\sigma_x^T(t) f_m(x,t) \leq \|\sigma_x(t)\| \|f_m(x,t)\|$ and taking η infinitely small, condition (2.103) is satisfied if:

$$\int_{\tau=t_o}^{t} \|\sigma_x(\tau)\| d\tau \geq \|f_m(x,t)\| \tag{2.104}$$

While condition (2.104) is not satisfied, $\|\sigma_x(t)\|$ will only be driven towards zero while $\tilde{K}(t) > f_m(x,t)$. By the assumption $\int_{\tau=t_o}^{\infty} \max\{0, (\|f_m(x,t)\| - (1-\epsilon)\hat{f}_m)\} d\tau = \infty$ with $0 < \epsilon \in \mathbb{R}$ arbitrarily small it is ensured that over the infinite time horizon the equation $\|f_m(x,t)\| = \hat{f}_m$ holds infinitely long. Therefore, $\sigma_x(t)$ will be sufficiently long nonzero to ensure that $\tilde{K}_s(t)$ will satisfy $\tilde{K}_s(t) > \hat{f}_m$ after some, possibly infinite, time t.
The boundedness of $\tilde{K}_s(t)$ follows instantly from the fact that the reaching law (condition (2.81)) is satisfied for $\tilde{K}_s(t) > \hat{f}_m$, in which case the closed-loop system is driven into the sliding mode in finite time. Once the system is driven into sliding mode and $\tilde{K}_s(t) > \hat{f}_m$, $\sigma_x(t)$ will remain zero thereby stopping the adaptation process. □

Despite the fact that Lemma 2.3.7 shows that the switching gain is increased until the system is driven into sliding mode, Method I has some severe practical disadvantages:

1. In case of a large initial distance from the sliding surface, the switching gain $\tilde{K}_s(t)$ will increase quickly due to this error and not because of a model-mismatch. This may result in a switching gain which is significantly larger then necessary.

2. Noise on the measurements will prevent $\sigma_x(t)$ attaining zero exactly so the adaptive gain will continue to increase.

3. The adaptation law can only increase the gain but never decrease it. Thus if the circumstances change such that a smaller switching gain is permitted, the adaptation law is not able to adapt to these new circumstances.

A more sophisticated way of determining the switching gain is by the adaptation law introduced in (Lenz et al. 1998) for the single input case. In (Wheeler et al. 1998) a similar, but more advanced adaptation procedure is used. A straightforward extension to the multiple input case is presented here. The discontinuous control component is taken as defined in Definition 2.3.2. It is known that, for some sufficiently large $\tilde{K}_s(t)$, the controller steers the closed-loop system into the quasi sliding mode band $\|\sigma_x(t)\| < \Delta$. If we demand no more then just steering the closed-loop system into the quasi sliding mode band, then the switching gain can be decreased within this band and increased outside it. This can be translated into a adaptation law as presented in the next definition.

Definition 2.3.8 *The Method II adaptive switching gain is defined as:*

$$\tilde{K}_s(t) = \left| \tilde{K}_s(t_o) + \int_{\tau=t_o}^{t} (\|\sigma_x(\tau)\| - \psi)\, d\tau \right| \tag{2.105}$$

where $\psi \in \mathbb{R}$ is a positive constant satisfying $\psi < \Delta$.

Remark 2.3.9 *In (Lenz et al. 1998) it was shown for the single input case that the adaptive gain $\tilde{K}_s(t)$ converges to an optimal gain K_o which is defined as the maximum value of the disturbance. However, in the case that the disturbance remains smaller then the maximum value for some time, the system would be driven closer to the switching surface. In this case the term $(\|\sigma_x(\tau)\| - \psi)$ in equation (2.105) becomes negative thereby reducing the switching gain $\tilde{K}_s(t)$. Therefore, in general the switching gain $\tilde{K}_s$ does not converge to this so-called optimal gain.*

Having observed in remark 2.3.9 that the existing proof of convergence for the Method II adaptive switching gain in (Lenz et al. 1998) is not correct, it is not easy to prove that the closed-loop system is driven into a quasi sliding mode and remains in a quasi sliding mode thereafter. The following theorem shows, however, that a quasi sliding mode is reached in finite time. It does not guarantee that the system remains in a quasi sliding mode. However, whenever the system is driven out of the quasi sliding mode, it will return into a quasi sliding mode in finite time.

Theorem 2.3.10 *The closed-loop system formed by system represented by the equations (2.33) and (2.34), with controller (2.35), (2.36), (2.97) and the adaptive switching gain of Definition 2.3.8 reaches a quasi sliding mode with some bounded $\Delta > \delta$ in finite time.*

Proof : *Substituting (2.105) (with $\tilde{K}_s(t_o) = 0$) into condition (2.81) it follows that:*

$$\sigma_x^T(t)\left(sat\left(\frac{\sigma_x(t)}{\delta}\right)\left|\int_{\tau=t_o}^{t}(\|\sigma_x(\tau)\|-\psi)\,d\tau\right|+f_m(x,t)\right)\leq-\eta\|\sigma_x(t)\| \quad (2.106)$$

For $\|\sigma_x(t)\| \geq \Delta$ condition (2.106) becomes:

$$\|\sigma_x(t)\|\int_{\tau=t_o}^{t}(\|\sigma_x(\tau)\|-\psi)\,d\tau\geq\eta\|\sigma_x(t)\|+\sigma_x^T(t)f_m(x,t) \quad (2.107)$$

By the triangle inequality $\sigma_x^T(t)f_m(x,t) \leq \|\sigma_x(t)\|\|f_m(x,t)\|$ condition (2.107) is satisfied if:

$$\int_{\tau=t_o}^{t}(\|\sigma_x(\tau)\|-\psi)\,d\tau\geq\eta+\|f_m(x,t)\| \quad (2.108)$$

Taking the worst case $\|f_m(x,\tau)\| = \hat{f}_m \ \forall\ \tau \in [t_o,t]$, $\hat{f}_m = \max\{\|f_m(x,t)\|\}\ \forall\ t$, and $\|\sigma_x(\tau)\| = \Delta\ \forall\ \tau \in [t_o,t]$, then it can be concluded that condition (2.108) is satisfied if:

$$\int_{\tau=t_o}^{t}(\delta-\psi)\,d\tau\geq\eta+\hat{f}_m \quad (2.109)$$

which is identical to:

$$(\delta-\psi)\,t\geq\eta+\hat{f}_m \quad (2.110)$$

Condition (2.110) is satisfied for:

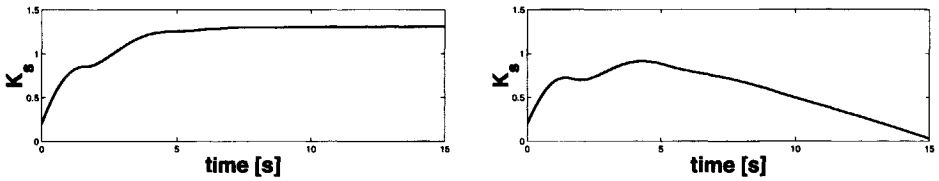
$$t>\frac{\eta+\hat{f}_m}{\delta-\psi} \quad (2.111)$$

Hence, the reaching condition is satisfied in finite time, which proofs the theorem.□

While Theorem 2.3.10 proofs that the closed-loop system is driven in a quasi sliding mode, is does not guarantee that the closed-loop system remains in a quasi sliding mode. However, the closed-loop system will return in a quasi sliding mode whenever it is driven out of it. Comparing Method II with the drawbacks of Method I we see that:

1. In the case of a large initial error, the switching gain $\tilde{K}_s(t)$ will increase quickly due to this error, but once the system has reached the quasi sliding band $\|\sigma_x(t)\| < \psi$ the switching gain will decrease again.

2. Noise on the measurements does not disturb the adaptation procedure if the quasi sliding band is chosen sufficiently large.



(a) (b)

Fig. 2.8: Figure (a) represents the Method I adaptive switching gain, figure (b) the Method II adaptive switching gain.

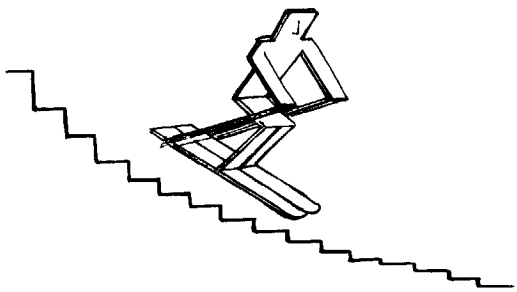
3. Where the Method I adaptation law adjusted the switching gain according to the worst case circumstances (Method I has a non-decreasing switching gain), Method II automatically decreases the gain once the system is within the region $\sigma_x(t) < \psi$.

Both adaptation schemes are illustrated in the next example.

Example 2.3.11 *Figure 2.8 represents the switching gain obtained from the adaptive gains defined in Definitions 2.3.6 and 2.3.8 (with $\psi = 0.1$) respectively for the pendulum example used at the beginning of this chapter. The only difference is that the switching gain presented in Figure 2.8.b is decreasing slowly after reaching the maximum value.*

2.4 Summary

This chapter introduces the basic concepts of continuous-time sliding mode control. The pendulum example at the beginning of this chapter was largely inspired by (Edwards and Spurgeon 1998). The Sections 2.1 and 2.2 successively present a design procedure for state-based and output based continuous-time sliding mode controllers. These methods can be found in for example the books (Utkin 1992, Edwards and Spurgeon 1998, Utkin et al. 1999). In Section 2.3 the effect of chattering and the quasi sliding mode concept are discussed. It is mentioned that several methods to reduce or even eliminate chattering exist. Two options are discussed in detail. The first option is to replace the discontinuous control part by a continuous approximation such as for example the saturation function as can be found in (Slotine 1984, Slotine and Li 1991). The second option to reduce chattering is to introduce an adaptive switching gain which adapts the gain according to the circumstances. Two adaptation methods known from literature are introduced and compared. Method I can be found in (Leung et al. 1991, Su et al. 1991, Wang and Fan 1993, Jiang et al. 1994, Roh and Oh 2000). Method II was introduced in (Lenz et al. 1998) for single input systems. In (Wheeler et al. 1998) a similar method is introduced for multiple input systems. It is shown that under practical circumstances, where perfect sliding mode cannot be attained, the Method I adaptation law is unstable. This in contrast to the Method II adaptation law, which for a suitable choice of parameters results in a bounded switching gain.



3

Discrete-Time Sliding Mode Control

As described in the previous chapter, continuous-time sliding mode control (CSMC) has a strong theoretical background. However, a large class of continuous-time systems are in fact computer controlled. In these cases information about the system (measurements) are only available at specific time instances and control inputs can only be changed at these time instances. In some cases it is in fact hard to get around sampling. In (Åstrom and Wittenmark 1997) examples of systems which are inherently sampled are given, they are divided in:

- Sampling due to the measurement system, such as Radar systems, the use of analytical instruments, economic systems, and magnetic flow meters.
- Sampling due to pulsed operation, such as Thyristor control, biological systems, internal-combustion engine, and partical accelerators.

In this thesis sampled, continuous-time systems are called discrete-time systems. The inter-sample behaviour of the continuous-time system will not be considered, only the values at the sample instances are considered. Furthermore, it is assumed that the sampling takes place by sample-and-hold, i.e. the signal remains constant within the sampling period. For these systems it is often assumed that the sampling frequency is sufficiently high to assume that the closed-loop system is continuous-time (Young and Özgüner 1999). However, the actual closed-loop system cannot be driven into (true) sliding mode but quasi sliding mode which was defined in Section 2.3.1. Obviously, the most apparent difference between a discrete-time system and a continuous-time system is the limited switching speed of the discontinuous control part. However, not only the switching frequency is limited for a discrete-time system. Also the exact timing of a change of input constrained to the sampling instances denoted by $t = kT_s$ or simply by k. Therefore, a perfect sliding mode is only possible in the case of perfect model knowledge and the absence of disturbances. In all other cases, the best a CSMC can achieve is keeping the system close to the switching surface.
This chapter explores the field of Discrete-Time Sliding Mode Control (DSMC). The main difference between a CSMC and a DSMC is the fact that a DSMC

is based on a discrete-time model of the system. In this way, the controller is automatically constrained to the sampling frequency by which we mean that the control signal inevitable only changes at the sample instances. It is no longer assumed that the control signal can switch with infinite frequency once sliding mode has been obtained, thereby being a more realistic design procedure. CSMC has already received a lot of attention within the control community. DSMC first appeared in the control literature in the mid 80s (Milosavljević 1985) to be followed by a growing list of publications, see for example (Sarpturk et al. 1987), (Aly and All 1990), (Furuta 1990), (Sira-Ramírez 1991), (Spurgeon 1992), (Hung et al. 1993), (Bartolini et al. 1995), (Gao et al. 1995), (Bartoszewicz 1996), (Jung and Tzou 1996), (Iordanou and Surgenor 1997), (Bartoszewicz 1998), (Hui and Żak 1999), and (Golo and Milosavljević 2000). As for CSMC, the design procedure of the controller is split into two steps:

- **Step 1.** Computation of a sliding surface $\sigma[k] = Sx[k]$ which has stable internal dynamics.
- **Step 2.** Establishing a control law which steers the closed-loop system towards the sliding surface and ensures the system trajectories to stay as close as possible to the surface.

The first step of the design procedure is exactly the same as the design procedure presented for the CSMC. It is assumed that the closed-loop system is kept close enough to the sliding surface to approximate the switching function $\sigma[k]$ by zero. Therefore, for determining the sliding surface the reader is referred to Sections 2.1.1 and 2.2.1 for the state-based and output based case respectively.
The second step of the design procedure is different for CSMC and DSMC. Due to the limited switching frequency, the closed-loop system cannot be driven into a sliding mode. Also the definition of a reaching law is not as straightforward as for the continuous time case. Section 3.1 explores the field of State based Discrete-time Sliding mode control (SDSMC). Several reaching laws are presented together with the control laws to satisfy these conditions. In Section 3.2, three possible Output-based Discrete-time Sliding Mode Controllers are proposed. Finally, Section 3.3 presents some practical considerations.

3.1 State-Based Discrete-Time Sliding Mode Control

This section considers Step 2 of the design procedure for a SDSMC. For Step 1 of the design procedure the reader is referred to Section 2.1.1. The following controllable system is considered:

$$x[k+1] = Ax[k] + Bu[k] + f(x,k) \tag{3.1}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $f \in \mathbb{R}^n$ uniformly bounded, and the matrices (A, B) of appropriate size. The switching function is defined by:

$$\sigma_x[k] = Sx[k] \tag{3.2}$$

By defining the transformation:

$$\begin{bmatrix} \bar{x}_1[k] \\ \sigma_x[k] \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{N}(S)^T \\ S \end{bmatrix}}_{=T_\sigma} x[k] \tag{3.3}$$

(the columns of $\mathcal{N}(S)$ span the nullspace of S and S has been chosen in the Step 1 design process such that $SB = I_m$) the following system representation is obtained:

$$\bar{x}_1[k+1] = \tilde{A}_{11}\bar{x}_1[k] + \tilde{A}_{12}\sigma_x[k] + f_u(x,k) \tag{3.4}$$

$$\sigma_x[k+1] = \tilde{A}_{21}\bar{x}_1[k] + \tilde{A}_{22}\sigma_x[k] + u[k] + f_m(x,k) \tag{3.5}$$

Furthermore, the matrix S has been chosen such that the dynamics in the sliding mode are stable, i.e. all eigenvalues of $\tilde{A}_{11}$ are within the unit circle. The problem to solve now is the construction of a control law which drives the above system into the sliding mode. Several control laws can be constructed to achieve this. They are presented in the next sections.

3.1.1 Discretized Continuous-Time Reaching Law

As can be seen in the previous chapter, a control law for a CSMC can be attained by the construction of a Lyapunov function. Starting from this Lyapunov function, a reaching law is constructed from which the control law automatically follows. Therefore, it seems attractive to discretize this reaching law to facilitate the design process for a DSMC. Using a first order forward approximation of the derivative $\dot{\sigma}_x(t)$ the discrete-time reaching law turns out to be:

$$(\sigma_x[k+1] - \sigma_x[k]) \leq -\eta \frac{\sigma_x[k]}{\|\sigma_x[k]\|} \tag{3.6}$$

By the substitution of equation (3.5) into condition (3.6), a control law can easily be constructed. However, the above reaching law does not guarantee convergence towards the sliding surface. It is true that, at any time instant k, the system is forced to move towards the sliding surface. However, a control input leading to the sequence $\sigma_x[k] = \{c(-1)^k\}$, with c arbitrary large, satisfies the above reaching law. Therefore, fulfilling the above reaching law does not impose a sufficient condition for convergence.

3.1.2 Sarpturk Reaching Law

A much better reaching law is given by (Sarpturk et al. 1987):

$$|\sigma_{x,i}[k+1]| < |\sigma_{x,i}[k]| \tag{3.7}$$

where the notation $\sigma_{x,i}$ stands for the i^{th} entry of the vector σ_x. In this case not only the direction of the closed-loop system is specified (namely towards the

sliding surface), but also the norm of the switching function is defined to be strictly decreasing. Reaching law (3.7) can also be formulated as:

$$[\sigma_{x,i}[k+1] - \sigma_{x,i}[k]] \ \text{sign}(\sigma_{x,i}[k]) \ < \ 0 \tag{3.8}$$

$$[\sigma_{x,i}[k+1] + \sigma_{x,i}[k]] \ \text{sign}(\sigma_{x,i}[k]) \ > \ 0 \tag{3.9}$$

The first condition implies that the closed-loop system should be moving in the direction of the sliding surface, whereas the second condition implies that the closed-loop system is not allowed to go too far in that direction. In other words, condition (3.8) results in a lower bound for the control action and condition (3.9) results in an upper bound. In (Sarpturk et al. 1987) the following control law is proposed:

$$u[k] = -K(x, \sigma_x)x[k] \tag{3.10}$$

where the elements $K_{i,j}(x, \sigma_x) \ \forall \ i = 1 \ldots m, j = 1 \ldots n$ of the gain matrix $K(x, \sigma_x)$ are determined as follows:

$$K_{i,j}(x, \sigma_x) = \begin{cases} K_{i,j}^{+} & \text{if} \quad \sigma_{x,i}[k]x_j[k] > 0 \\ K_{i,j}^{-} & \text{if} \quad \sigma_{x,i}[k]x_j[k] < 0 \end{cases} \tag{3.11}$$

The computation of the coefficients K_{ij}^{+} and K_{ij}^{-} is not an easy task. They can be determined by evaluating conditions (3.8) and (3.9) resulting in an upper and a lower bound for each K_{ij}^{+} and K_{ij}^{-}. Indeed, there are circumstances where they do not exist at all (Sarpturk et al. 1987).

3.1.3 Gao Reaching Law

Another approach to obtain a DSMC was introduced in (Gao et al. 1995). According to (Gao et al. 1995), the closed-loop system should posses the following properties:

R_I Starting from any initial state, the trajectory will move monotonically towards the switching plane and cross it in finite time.

R_{II} Once the trajectory has crossed the switching plane the first time, it will cross the plane again in every successive sampling period, resulting in a zigzag motion about the switching plane.

R_{III} The size of each successive zigzagging step is nonincreasing and the trajectory stays within a specified band.

Despite the fact that the above conditions were stated for a single input system, they can be applied to a multiple input system by applying the three rules to the m entries of the switching function $\sigma_x[k]$ independently. The reaching law, proposed in (Gao et al. 1995) and modified by (Bartoszewicz 1996) for the single input case, can be straightforwardly extended to the multiple input case as:

$$\sigma_x[k+1] = \Phi\sigma_x[k] - \begin{bmatrix} K_{s,1} \ \text{sign}(\sigma_{x,1}[k]) \\ K_{s,2} \ \text{sign}(\sigma_{x,2}[k]) \\ \vdots \\ K_{s,m} \ \text{sign}(\sigma_{x,m}[k]) \end{bmatrix} \tag{3.12}$$

for some diagonal matrix $\Phi \in \mathbb{R}^{m\times m}$ with the entries $0 \leq \Phi_{i,i} < 1 \ \forall \ i = 1 \ldots m$, and the gains $K_{s,i} \in \mathbb{R}$ satisfying $K_{s,i} > 0 \ \forall \ i = 1 \ldots m$. Neglecting the unknown disturbance we can determine the control signal from equations (3.5) and (3.12) to be:

$$u[k] = \left(\Phi - \tilde{A}_{22}\right)\sigma_x[k] - \tilde{A}_{21}\bar{x}_1[k] - \begin{bmatrix} K_{s,1}\ \mathrm{sign}(\sigma_{x,1}[k]) \\ K_{s,2}\ \mathrm{sign}(\sigma_{x,2}[k]) \\ \vdots \\ K_{s,m}\ \mathrm{sign}(\sigma_{x,m}[k]) \end{bmatrix} \tag{3.13}$$

Theorem 3.1.1 *The closed-loop system formed by system (3.4),(3.5) and controller (3.13) satisfy Rules R_I, R_{II}, and R_{III} if:*

$$K_{s,i} \ \geq \frac{1+\Phi_{i,i}}{1-\Phi_{i,i}}\hat{f}_{m,i} \tag{3.14}$$

where $\hat{f}_m = \max\{f_m(x,k)\}$. The closed-loop system will then be driven into a quasi sliding band Δ satisfying:

$$\Delta > 2\sqrt{\sum_{i=1}^{m}\left(\frac{1}{1-\Phi_{i,i}}\hat{f}_{m,i}\right)^2} \tag{3.15}$$

Proof: *The gain condition can be found for the single input case in (Bartoszewicz 1996). By design choice each entry of the switching function was coupled to an individual entry of the control vector. In this way, the multiple input case is reduced to m single input problems (m being the number of inputs). Then, choosing the minimal gain fulfilling the gain condition, it is also given that the single input system will go into the quasi sliding mode band (Bartoszewicz 1996):*

$$\Delta_i > \frac{2}{1-\Phi_{i,i}}\hat{f}_{m,i} \tag{3.16}$$

Δ_i denoting the i^{th} component of Δ. Constructing the vector Δ of the above entries and taking the Euclidean norm results in the presented quasi sliding mode band. □

Interesting to note is that for $\Phi = 0$ the smallest possible quasi sliding mode band is obtained given by $\Delta = 2\|\hat{f}_m\|$.

3.1.4 Linear Reaching Law

Yet another way of defining the reaching law can be found in for example (Spurgeon 1992, Hui and Żak 1999), given by:

$$\sigma_x[k+1] = \Phi\sigma_x[k] \tag{3.17}$$

for some diagonal $\Phi \in \mathbb{R}^{m\times m}$ with all entries $0 \leq \Phi_{i,1} < 1 \ \forall \ i = 1 \ldots m$. Notice that this reaching law is similar to the reaching law in the previous section (equation (3.12)), however the switching part has been omitted. Also the Sarpturk reaching

law (3.7) is very similar to the above linear reaching law. In fact the above reaching law satisfies the Sarpturk reaching law. However, the above reaching law gives an exact description of the desired trajectory towards the sliding surface. Despite the fact that this trajectory cannot be achieved due to the unknown disturbance, the design of the controller is fairly straightforward. Again using equation (3.5) and neglecting the unknown disturbance term leads to the control law:

$$u[k] = \left(\Phi - \tilde{A}_{22}\right)\sigma_x[k] - \tilde{A}_{21}\bar{x}_1[k] \tag{3.18}$$

Theorem 3.1.2 *System (3.4),(3.5) under the control of (3.18) will be driven arbitrarily close to the quasi sliding mode band* Δ*:*

$$\Delta = \sqrt{\sum_{i=1}^{m}\left(\frac{1}{1-\Phi_{i,i}}\hat{f}_{m,i}\right)^2} \tag{3.19}$$

where $\hat{f}_m = \max\{f_m(x,t)\}$ *and* Φ *has all its eigenvalues within the unit circle.*
Proof: *Inserting the control law (3.18) in equation (3.5) leads to:*

$$\sigma_x[k+1] = \Phi\sigma_x[k] + f_m(x,k) \tag{3.20}$$

Since Φ *is diagonal each entry of* σ_x *can be treated independently. Starting from an arbitrary value* $\sigma_{x,i}[0]$*, the future values of* $\sigma_x[k]$ *can be found to be:*

$$\sigma_{x,i}[k] = \Phi_{i,i}^k\sigma_{x,i}[0] + \sum_{j=0}^{k-1}\Phi_{i,i}^j f_{m,i}(x,j) \tag{3.21}$$

Because $f_{m,i}(x,k) < \hat{f}_{m,i}$ *it can be written:*

$$|\sigma_{x,i}[k]| \quad < \quad |\Phi_{i,i}^k\sigma_{x,i}[0]| + \sum_{j=0}^{k-1}\Phi_{i,i}^j\hat{f}_{m,i} \tag{3.22}$$

For sufficiently large k *the above can be approximated by:*

$$|\sigma_{x,i}[k]| \quad \leq \quad |\Phi_{i,i}^k\sigma_{x,i}[0]| + \sum_{j=0}^{k-1}\Phi_{i,i}^j\hat{f}_{m,i} \tag{3.23}$$

The righthand side of the above inequality will approach $\frac{1}{1-\Phi_{i,i}}\hat{f}_{m,i}$ *arbitrarily close. Evaluating each entry of* σ_x *in the same way and taking the Euclidean norm of the vector* Δ *composed of the elements* $\Delta_i = \frac{1}{1-\Phi_{i,i}}\hat{f}_{m,i}$ *results in the specified quasi sliding mode band* Δ*.* □

Like the controller developed in the previous section, the discrete-time sliding mode band is minimized with $\Phi = 0$ leading to $\Delta = \|\hat{f}_m\|$.

3.1.5 Linear Reaching Law with Disturbance Estimation

As can be seen in the previous sections, the quasi sliding mode band is rather large. The smallest quasi sliding mode band is obtained with the linear reaching law where $\Phi = 0$, but still the quasi sliding mode band has the same norm as the upper bound for the disturbance. For the Gao reaching law the minimum quasi sliding mode band is even twice the maximum norm of the disturbance. To overcome this problem a disturbance estimator is introduced. Define $\tilde{d}[k] \in \mathbb{R}^m$ by:

$$\tilde{d}[k] = \tilde{d}[k-1] + \sigma_x[k] - \Phi\sigma_x[k-1] \tag{3.24}$$

where $\tilde{d}[k]$ is the estimation of the disturbance vector projected on σ_x by which it is meant that ideally $\tilde{d}[k] = f_m(x,k)$. The control law (3.18) should then be changed to:

$$u[k] = \left(\Phi - \tilde{A}_{22}\right)\sigma_x[k] - \tilde{A}_{21}\bar{x}_1[k] - \tilde{d}[k] \tag{3.25}$$

Theorem 3.1.3 *The closed-loop system formed by the controllable system (3.4),(3.5) and controller (3.25) with disturbance estimator (3.24) will be driven arbitrarily close to the discrete-time sliding mode band Δ:*

$$\Delta = \sqrt{\sum_{i=1}^{m}\left(\frac{1}{1-\Phi_{i,i}}\hat{\delta}_{f,i}\right)^2} \tag{3.26}$$

where $\hat{\delta}_{f,i}$ is the maximum rate of change of the disturbance vector defined by:

$$|f_{m,i}(x,k+1) - f_{m,i}(x,k)| < \hat{\delta}_{f,i} \qquad \forall \quad k \tag{3.27}$$

and the matrix Δ has all its eigenvalues within the unit circle.

Proof: *Since Φ is diagonal, the entries of σ_x are decoupled and hence can be treated independently. Substituting equation (3.5) and (3.25) into the expression for $\tilde{d}_i[k]$ (equation (3.24)) results in:*

$$\tilde{d}_i[k] = f_{m,i}(x,k-1) \tag{3.28}$$

Then starting from some arbitrary bounded $\tilde{d}_i[0]$ it can be written that:

$$\begin{array}{rcl} \sigma_{x,i}[1] & = & \Phi_{i,i}\sigma_{x,i}[0] + f_{m,i}(0) - \tilde{d}_i[0] \\ \sigma_{x,i}[2] & = & \Phi_{i,i}\sigma_{x,i}[1] + f_{m,i}(1) - f_{m,i}(0) \end{array} \tag{3.29}$$

For an arbitrary time $k > 1$, $\sigma_{x,i}[k]$ can be found to be:

$$\sigma_{x,i}[k] = \Phi_{i,i}^{k-1}\sigma_{x,i}[1] + \sum_{j=0}^{k-2}\Phi_{i,i}^{j}f_{m,i}(j+1) - \sum_{j=0}^{k-2}\Phi_{i,i}^{j}f_{m,i}(j) \tag{3.30}$$

Because $|f_{m,i}(k+1) - f_{m,i}[k]| < \hat{\delta}_{f,i}$, the above equation is bounded by:

$$\sigma_{x,i}[k] < \Phi_{i,i}^{k-1}\sigma_{x,i}[1] + \sum_{j=0}^{k-2}\Phi_{i,i}^{j}\hat{\delta}_{f,i} \tag{3.31}$$

The righthand side of the above inequality will approach $\frac{1}{1-\Phi_{i,i}}\hat{\delta}_{f,i}$ *arbitrarily close. Evaluating each entry of* σ_x *in the same way and taking the Euclidean norm of the vector* Δ *composed of the elements* $\Delta_i = \frac{1}{1-\Phi_{i,i}}\hat{\delta}_{f,i}$ *results in the specified* Δ. □

It should be noted that the above disturbance observer is based on the same idea presented in (Su et al. 2000) where the disturbance is recovered from the equation:

$$f(x,k) = x[k+1] - Ax[k] - Bu[k] \tag{3.32}$$

However, the task of the sliding mode controller is to steer the system towards the sliding surface. Therefore, it suffices to have an estimate of the matched disturbance vector, instead of the full disturbance vector. Another important advantage of this approach is that it is not important whether the switching function is based on the state of the system or its output, which is explored in Section 3.2.

3.1.6 Comparison

In this section the developed controller structures are compared in a simple simulation example (also considered in (Gao et al. 1995)). The system matrices are given by:

$$A = \begin{bmatrix} 1.2 & 0.1 \\ 0 & 0.6 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.33}$$

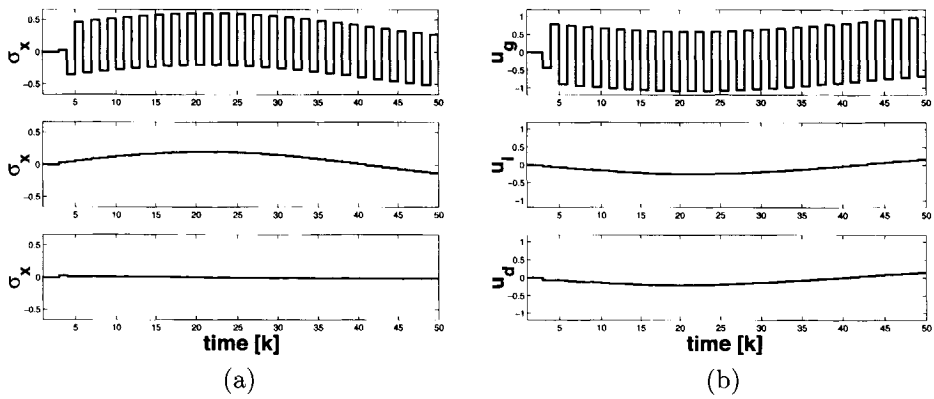
The switching function is defined by $S = [5 \quad 1]$, $\Phi = 0$, and the switching gain for the Gao controller is $K_s = 0.2$. The disturbance $f[k]$ is defined by:

$$f[k] = \begin{bmatrix} 0 \\ 0.2\sin\left(\frac{k}{4\pi}\right) \end{bmatrix} \tag{3.34}$$

The control laws for the simulation results in Figure 3.1.a and 3.1.b are taken as equation (3.13) (top figures), equation (3.18) (middle figures), and equation (3.24) together with equation (3.25) (lower figures). Clearly the figures show that the Gao reaching law results in significant chattering, which is absent for the linear controllers. Besides the absence of chattering for the linear controller, the RMS (Root Mean Square) value of the switching function is for the linear controller significantly better as well. In fact, $\text{RMS}(\sigma_x) = 0.233$ for the Gao controller and $\text{RMS}(\sigma_x) = 0.132$ for the linear controller. Finally, the disturbance estimation controller gives the best results. As for the linear controller, there is no chattering, but now the RMS value of the switching function is reduced to $\text{RMS}(\sigma_x) = 0.012$, a significant improvement compared with the Gao controller and the linear controller.

3.2 Output-Based Discrete-Time Sliding Mode Control

Whereas SDSMC has received quite some attention in the past decade, Output-based Discrete-time Sliding Mode Control (ODSMC) is still relatively new. Although there exist some publications on ODSMC (see for example (Chan 1999)), they are all based on transfer functions. In this section, a state space model is used to obtain an ODSMC. The method is similar to the approach for OCSMC presented



(a) (b)

Fig. 3.1: Figure (a) presents the switching function for the Gao reaching law (top), the linear reaching law (middle) and the linear reaching law with disturbance estimation (bottom). Figure (b) presents the input signal for the Gao reaching law (top), the linear reaching law (middle) and the linear reaching law with disturbance estimation (bottom).

in for example (Edwards and Spurgeon 1998). It is assumed that the dynamics of the closed-loop system in a quasi sliding mode may be approximated by the reduced order dynamics in a "true" sliding mode. Therefore, the sliding surface design is similar as presented for the OCSMC. The only difference is of course that for stability in discrete-time, all poles are restricted to lie within the unit disc of the complex plane, as opposed to the left half plane for the continuous-time case. The following controllable, observable and minimum phase system is considered:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] + f(x,k) \\ y[k] &= Cx[k] \end{aligned} \tag{3.35}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, $f \in \mathbb{R}^n$, and the matrices (A, B, C) are of appropriate size. Furthermore, it is assumed that $p > m$ (i.e. there are more outputs than inputs) and $\text{Rank}\{CB\} = m$. By definition, a linear discrete-time system has all its invariant zeros within the complex unit circle. The switching function is defined by:

$$\sigma_y[k] = Sy[k] \tag{3.36}$$

with $S \in \mathbb{R}^{m \times p}$. Then, by the nonsingular transformation $T_\sigma \in \mathbb{R}^{n \times n}$ given by:

$$\begin{bmatrix} \bar{x}_0[k] \\ \bar{x}_1[k] \\ \bar{y}_1[k] \\ \sigma_y[k] \end{bmatrix} = T_\sigma x[k] \tag{3.37}$$

system (3.35) is brought into the following form:

$$\begin{bmatrix} \bar{x}_0[k+1] \\ \bar{x}_1[k+1] \\ \bar{y}_1[k+1] \\ \sigma_y[k+1] \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} & \tilde{A}_{14} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} & \tilde{A}_{34} \\ \tilde{A}_{41} & \tilde{A}_{42} & \tilde{A}_{43} & \tilde{A}_{44} \end{bmatrix} \begin{bmatrix} \bar{x}_0[k] \\ \bar{x}_1[k] \\ \bar{y}_1[k] \\ \sigma_y[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_m \end{bmatrix} u[k] + \begin{bmatrix} f_{\bar{x}_0}(x,k) \\ f_{\bar{x}_1}(x,k) \\ f_{\bar{y}_1}(x,k) \\ f_m(x,k) \end{bmatrix} \tag{3.38}$$

where it was assumed that S was chosen such that $SB = I_m$. In the previous section it was demonstrated that a linear reaching law, possibly with a disturbance estimator, leads to the smallest quasi sliding mode band while preventing chattering. Therefore, the linear reaching law is selected which is given by:

$$\sigma_y[k+1] = \Phi \sigma_y[k] \tag{3.39}$$

where $\Phi \in \mathbb{R}^{m \times m}$ is, for simplicity, chosen as a diagonal matrix with all entries $0 \leq \phi_i < 1$. From equation (3.38) and (3.39) the feedback control term can be determined to be:

$$u[k] = \left(\Phi - \tilde{A}_{44}\right) \sigma_y[k] - \tilde{A}_{43}\bar{y}_1[k] - \tilde{A}_{41}\bar{x}_0[k] - \tilde{A}_{42}\bar{x}_1[k] - f_m(x,k) \tag{3.40}$$

Applying the above control law to the system (3.38) results in the closed-loop system:

$$\begin{bmatrix} \bar{x}_0[k+1] \\ \bar{x}_1[k+1] \\ \bar{y}_1[k+1] \\ \sigma_y[k+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} & \tilde{A}_{14} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} & \tilde{A}_{34} \\ 0 & 0 & 0 & \Phi \end{bmatrix}}_{=A_{cl}} \begin{bmatrix} \bar{x}_0[k] \\ \bar{x}_1[k] \\ \bar{y}_1[k] \\ \sigma_y[k] \end{bmatrix} + \begin{bmatrix} f_{\bar{x}_0}(x,k) \\ f_{\bar{x}_1}(x,k) \\ f_{\bar{y}_1}(x,k) \\ 0 \end{bmatrix} \tag{3.41}$$

From (3.41) it can be easily seen that the poles of the closed-loop system are given by:

$$\lambda(A_{cl}) = \lambda(\Phi) \cup \lambda\left(\tilde{A}_{11}\right) \cup \lambda\left(\begin{bmatrix} \tilde{A}_{22} & \tilde{A}_{23} \\ \tilde{A}_{32} & \tilde{A}_{33} \end{bmatrix}\right) \tag{3.42}$$

The eigenvalues of Φ are by design choice stable. The eigenvalues of the matrix $\tilde{A}_{11}$ correspond to the invariant zeros of the open-loop system and therefore they are stable. The eigenvalues of the block $[\tilde{A}_{22}\ \tilde{A}_{23}\ ;\ \tilde{A}_{32}\ \tilde{A}_{33}]$ are assumed to be placed within the unit disc by the design procedure for the switching function. Consequently, the closed-loop system is stable.

However, the previous control law is not implementable if $\bar{x}_0[k]$ and $\bar{x}_1[k]$ are unknown. Like the disturbance and modeling error component $f_m(x,k)$, which is assumed to be unknown, also the state-components $\bar{x}_0[k]$ and $\bar{x}_1[k]$ are not known. In the following subsections, three possible controller structures are introduced, based solely on output measurements, to replace the above control law.

3.2.1 Direct Linear Controller

The most straightforward way to come to an applicable control law is to omit the unknown parts, resulting in the control law:

$$u[k] = \left(\Phi - \tilde{A}_{44}\right)\sigma_y[k] - \tilde{A}_{43}\bar{y}_1[k] \tag{3.43}$$

It is noted that this is exactly the structure used for OCSMC, which can be found in for example (Edwards and Spurgeon 1998), with the absence of the discontinuous control component. It can be easily determined by the substitution of (3.43) into equation (3.38) that the closed-loop dynamics are given by:

$$\begin{bmatrix} \bar{x}_0[k+1] \\ \bar{x}_1[k+1] \\ \bar{y}_1[k+1] \\ \sigma_y[k+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} & \tilde{A}_{14} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} & \tilde{A}_{34} \\ \tilde{A}_{41} & \tilde{A}_{42} & 0 & \Phi \end{bmatrix}}_{=A_{cl}} \begin{bmatrix} \bar{x}_0[k] \\ \bar{x}_1[k] \\ \bar{y}_1[k] \\ \sigma_y[k] \end{bmatrix} + \begin{bmatrix} f_{\bar{x}_0}(x,k) \\ f_{\bar{x}_1}(x,k) \\ f_{\bar{y}_1}(x,k) \\ f_m(x,k) \end{bmatrix} \tag{3.44}$$

Clearly, the poles of the above system do not correspond to the designed poles given by (3.42). Therefore it is advisable to check the eigenvalues of the above matrix for stability.

Remark 3.2.1 *It should be noted that the design parameters S, and Φ could be directly determined from the closed loop matrix A_{cl} given in (3.44) by making S explicit. Besides that from the point of view of the sliding mode control methodology this can be considered as reverse engineering, it leads to a more complex design procedure.*

3.2.2 Direct Linear Controller with Disturbance Estimator

Applying the derived control law of the previous section (equation (3.43)) to the system (3.38), $\sigma_y[k+1]$ can be determined to be:

$$\sigma_y[k+1] = \Phi\sigma_y[k] + \tilde{A}_{41}\bar{x}_0[k] + \tilde{A}_{42}\bar{x}_1[k] + f_m(x,k) \tag{3.45}$$

Comparing the above with the desired $\sigma_y[k+1] = \Phi\sigma_y[k]$ then it can be concluded that the error is given by $\tilde{A}_{41}\bar{x}_0[k] + \tilde{A}_{42}\bar{x}_1[k] + f_m(x,k)$, hence the unknown terms $\bar{x}_0[k]$ and $\bar{x}_1[k]$ could be considered as disturbances just like the disturbance $f_m(x,k)$. Therefore, a similar disturbance estimator is employed as presented in Section 3.1.5, equation (3.24), where σ_x is now replaced by σ_y. Then $\tilde{d}[k] \in \mathbb{R}^m$ is given by:

$$\tilde{d}[k] = \tilde{d}[k-1] + \sigma_y[k] - \Phi\sigma_y[k-1] \tag{3.46}$$

It is noted that, despite the fact that the switching function σ_y is now based on the outputs in contrast to the state based case in Section 3.1.5, the structure is identical. The proposed disturbance estimator solely estimates disturbances and uncertainties in the space of the switching function, either in the state-based or the output-based case. The only difference is the fact that in the output-based case, the unknown

zero dynamics form an uncertainty as well and hence are also estimated by the disturbance estimator. Therefore, in the output-based case $\tilde{d}[k]$ ideally represents the disturbance vector $f_m(x,k)$ plus the terms $\tilde{A}_{41}\bar{x}_0[k] + \tilde{A}_{42}\bar{x}_1[k]$. The control law then becomes:

$$u[k] = \left(\Phi - \tilde{A}_{44}\right)\sigma_y[k] - \tilde{A}_{43}\bar{y}_1[k] - \tilde{d}[k] \tag{3.47}$$

In the state-based case it was assumed that the rate of change of the disturbance vector $f_m(x,k)$ was limited. However, in the output-based case $\tilde{d}[k]$ is used to estimate the term $f_m(x,k) + \tilde{A}_{41}\bar{x}_0[k] + \tilde{A}_{42}\bar{x}_1[k]$. Still we can assume that if the sampling frequency is sufficiently high then the term $f_m(x,k) + \tilde{A}_{41}\bar{x}_0[k] + \tilde{A}_{42}\bar{x}_1[k]$ is slow enough.
To obtain the closed-loop dynamics the system state is augmented with the estimated disturbance vector, resulting in:

$$x_{tot}[k] = \begin{bmatrix} \bar{x}_0[k] & \bar{x}_1[k] & \bar{y}_1[k] & \sigma_y[k] & \tilde{d}[k] \end{bmatrix}^T \tag{3.48}$$

with $x_{tot} \in \mathbb{R}^{(n+m)}$. Then with equations (3.38), (3.46) and (3.47) the closed-loop dynamics can be determined to be $x_{tot}[k+1] = A_{cl}x_{tot}[k] + f_{tot}(x,k)$, where:

$$A_{cl} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} & \tilde{A}_{14} & 0 \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} & 0 \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} & \tilde{A}_{34} & 0 \\ \tilde{A}_{41} & \tilde{A}_{42} & 0 & \Phi & -I \\ \tilde{A}_{41} & \tilde{A}_{42} & 0 & 0 & 0 \end{bmatrix} \quad f_{tot}(x,k) = \begin{bmatrix} f_{\bar{x}_0}(x,k) \\ f_{\bar{x}_1}(x,k) \\ f_{\bar{y}_1}(x,k) \\ f_m(x,k) \\ f_m(x,k) \end{bmatrix} \tag{3.49}$$

Just as for the direct linear controller, the dynamics of the closed-loop system do not correspond to the designed dynamics. First of all the order of the closed-loop system has increased by the order of the disturbance estimator. Furthermore, the poles of the closed-loop system do not correspond to the designed poles (3.42). Consequently, similar to the direct linear controller, it is advisable to check the locations of the closed-loop poles. Also Remark 3.2.1 applies to the above control strategy. It is possible to design the parameters S and Φ by making them explicit in the closed-loop matrix A_{cl}. However, since the complexity is even further increased (in relation to the direct linear controller) the complexity of finding the parameters has increased as well.

3.2.3 Direct Linear Controller with Reduced Order State Estimation

The direct linear controller (Section 3.2.1, equation (3.43)), estimates the unknown terms by zero. However, a better estimation for these terms can be obtained with an observer. This method is inspired by the sliding mode observer design methodology given in (Haskara et al. 1998). A reduced order observer (in fact the order of the observer is $n-p$) is used to reconstruct the states $\bar{x}_0[k]$ and $\bar{x}_1[k]$. The observed states will be represented by $\hat{\bar{x}}_0[k]$ and $\hat{\bar{x}}_1[k]$. The control law resulting from these considerations is given by:

$$u[k] = \left(\Phi - \tilde{A}_{44}\right)\sigma_y[k] - \tilde{A}_{43}\bar{y}_1[k] - \tilde{A}_{41}\hat{\bar{x}}_0[k] - \tilde{A}_{42}\hat{\bar{x}}_1[k] \tag{3.50}$$

The reduced order state error observer is defined as:

$$\begin{bmatrix} \hat{\bar{x}}_0[k+1] \\ \hat{\bar{x}}_1[k+1] \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \hat{\bar{x}}_0[k] \\ \hat{\bar{x}}_1[k] \end{bmatrix} + \begin{bmatrix} \tilde{A}_{13} & \tilde{A}_{14} \\ \tilde{A}_{23} & \tilde{A}_{24} \end{bmatrix} \begin{bmatrix} \bar{y}_1[k] \\ \sigma_y[k] \end{bmatrix} - L(\sigma_y[k+1] - \Phi\sigma_y[k]) \tag{3.51}$$

where $L \in \mathbb{R}^{(n-p)\times m}$ is a design matrix. The following lemma gives the conditions for the above observer to be stable.

Lemma 3.2.2 *Finding a stabilizing matrix L for observer (3.51) is equivalent to the observer design for the system (A_{obs}, C_{obs}) where $A_{obs} \in \mathbb{R}^{(n-p)\times(n-p)}$ and $C_{obs} \in \mathbb{R}^{(n-p)\times p}$ are given by:*

$$A_{obs} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \qquad C_{obs} = \begin{bmatrix} \tilde{A}_{41} & \tilde{A}_{42} \end{bmatrix} \tag{3.52}$$

Proof : *Applying control law (3.50) it can be found from equation (3.38) that:*

$$\sigma_y[k+1] - \Phi\sigma_y[k] = \begin{bmatrix} \tilde{A}_{41} & \tilde{A}_{42} \end{bmatrix} \begin{bmatrix} \bar{x}_0[k] - \hat{\bar{x}}_0[k] \\ \bar{x}_1[k] - \hat{\bar{x}}_1[k] \end{bmatrix} \tag{3.53}$$

Substituting the above equation in the observer equation (3.51) results in:

$$\begin{bmatrix} \hat{\bar{x}}_0[k+1] \\ \hat{\bar{x}}_1[k+1] \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \hat{\bar{x}}_0[k] \\ \hat{\bar{x}}_1[k] \end{bmatrix} + \begin{bmatrix} \tilde{A}_{13} & \tilde{A}_{14} \\ \tilde{A}_{23} & \tilde{A}_{24} \end{bmatrix} \begin{bmatrix} \bar{y}_1[k] \\ \sigma_y[k] \end{bmatrix} - L \begin{bmatrix} \tilde{A}_{41} & \tilde{A}_{42} \end{bmatrix} \begin{bmatrix} \bar{x}_0[k] - \hat{\bar{x}}_0[k] \\ \bar{x}_1[k] - \hat{\bar{x}}_1[k] \end{bmatrix} \tag{3.54}$$

From equation (3.38) it is known that:

$$\begin{bmatrix} \bar{x}_0[k+1] \\ \bar{x}_1[k+1] \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{x}_0[k] \\ \bar{x}_1[k] \end{bmatrix} + \begin{bmatrix} \tilde{A}_{13} & \tilde{A}_{14} \\ \tilde{A}_{23} & \tilde{A}_{24} \end{bmatrix} \begin{bmatrix} \bar{y}_1[k] \\ \sigma_y[k] \end{bmatrix} \tag{3.55}$$

Subtracting the previous two equations from each other, with $e_{obs}[k] = \left[\hat{\bar{x}}_0[k] \quad \hat{\bar{x}}_1[k]\right]^T - \left[\bar{x}_0[k] \quad \bar{x}_1[k]\right]^T$, leads to:

$$e_{obs}[k+1] = \left(\underbrace{\begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}}_{= A_{obs}} - L \underbrace{\begin{bmatrix} \tilde{A}_{41} & \tilde{A}_{42} \end{bmatrix}}_{= C_{obs}} \right) e_{obs}[k] \tag{3.56}$$

From the above it can be concluded that the observer error converges to zero if the eigenvalues of $A_{obs} - LC_{obs}$ are within the unit circle. Constructing a stable observer is possible for all detectable pairs (A_{obs}, B_{obs}). □

To study the closed-loop stability of the system under control the system state is augmented with the observer state leading to the total state $x_{tot} \in \mathbb{R}^{(2n-p)}$:

$$x_{tot}[k] = \begin{bmatrix} \bar{x}_0[k] & \bar{x}_1[k] & \bar{y}_1[k] & \sigma_y[k] & \hat{\bar{x}}_o[k] & \hat{\bar{x}}_1[k] \end{bmatrix}^T \tag{3.57}$$

Then with equations (3.38), (3.50), and (3.51), the closed-loop dynamics are represented by $x_{tot}[k+1] = A_{cl}x_{tot}[k] + f_{tot}(x,k)$ with the closed-loop system matrix:

$$A_{cl} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} & \tilde{A}_{14} & 0 & 0 \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} & 0 & 0 \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} & \tilde{A}_{34} & 0 & 0 \\ \tilde{A}_{41} & \tilde{A}_{42} & 0 & \Phi & -\tilde{A}_{41} & -\tilde{A}_{42} \\ -L_1\tilde{A}_{41} & -L_1\tilde{A}_{42} & \tilde{A}_{13} & \tilde{A}_{14} & \left(\tilde{A}_{11} + L_1\tilde{A}_{41}\right) & \left(\tilde{A}_{12} + L_1\tilde{A}_{42}\right) \\ -L_2\tilde{A}_{41} & -L_2\tilde{A}_{42} & \tilde{A}_{23} & \tilde{A}_{24} & L_2\tilde{A}_{41} & \left(\tilde{A}_{22} + L_2\tilde{A}_{42}\right) \end{bmatrix} \tag{3.58}$$

and closed-loop disturbance vector:

$$f_{tot}(x,k) = \begin{bmatrix} f_{\bar{x}_0}(x,k) \\ f_{\bar{x}_1}(x,k) \\ f_{\bar{y}_1}(x,k) \\ f_m(x,k) \\ L_1 f_m(x,k) \\ L_2 f_m(x,k) \end{bmatrix} \tag{3.59}$$

where L has been partitioned according to $[L_1 \quad L_2]^T = L$ with $L_1 \in \mathbb{R}^{r\times m}$ and $L_2 \in \mathbb{R}^{(n-p-r)\times m}$. Again, just as for the direct linear controller and the direct linear controller with disturbance estimation, the dynamics of the closed-loop system do not necessarily correspond to the designed dynamics (3.42). Consequently it is advisable to check the locations of the closed-loop poles. Also Remark 3.2.1 applies to the above control strategy. It is possible to design the parameters S, Φ, and L by making them explicit in the closed-loop matrix A_{cl}. However, since the complexity is even further increased (in relation to both previous methods) the complexity of finding the parameters has increased as well.

3.3 Practical Considerations

It was shown in Section 3.1 that the parameter Φ plays an important role in the magnitude of the quasi sliding mode band. For the smallest possible quasi sliding mode band, Φ should be set to zero. In that case, the reaching law reduces to $\sigma[k+1] = 0$ and consequently the control input aims at reaching the sliding surface in one time step. Especially for a large initial distance from the switching surface this leads to excessively large control inputs. Worthwhile to mention is that the above problem does not arise very often for a CSMC. For a CSMC the derivative of the switching function is prescribed by the reaching law $\dot{\sigma}(t) = -\eta \operatorname{sign}(\sigma(t))$ whereas for a DSMC the magnitude of the switching function is prescribed by the

reaching law ($\sigma[k+1] = 0$ in this case). Therefore, for a DSMC far away from the switching surface, the distance to travel within one sampling period may become excessively large leading to an excessively large control input. As a solution to this problem, Section 3.3.1 proposes a varying reaching law which is less aggressive further away from the switching surface.
Another important issue is the choice, if any, of the switching gain. While it was shown in Section 3.1 that a DSMC without a discontinuous control term leads to the smallest quasi sliding mode band, some publications still claim a good performance of a DSMC due to the discontinuous control part (see for example the practical study (Souverijns and Vaes 2001)). Whether it is truly the case that the discontinuous control part increases the performance for a DSMC remains an interesting area for research. It could be explained by the fact that for very fast sampling the closed-loop system behaves more or less as a continuous-time system. Another explanation could be that the reaching law $\sigma[k+1] = \Phi\sigma[k]$ can be made less aggressive (i.e. the eigenvalues of the matrix Φ closer to the unit circle) while the discontinuous control term still gives an acceptable reaching time (i.e. the time to reach the quasi sliding mode band). Nevertheless, in the case that a discontinuous control term is used, it might be wise to choose the switching gain small. Large switching gains lead to excessive wear in gearboxes, strong currents and associated heat consumption in electrical actuators and could excite unmodeled fast dynamics in the system. In analogy with the CSMC, solutions can be found in softening the switching function close to th switching surface or defining an adaptive switching gain. We refer to Section 2.3.2 for the softened switching gains because they are identical for CSMC and DSMC. Section 3.3.2 explores the field of the adaptive switching gains for a DSMC.

3.3.1 Varying Reaching Law

A typical approach to circumvent the problem of excessively large control inputs is to bound the control input, see for example (Bartolini et al. 1995). If the control input is limited to $|u_i[k]| \leq \hat{u}_i[k] \quad \forall \quad i = 1 \ldots m$, then the applied control signal $u_a[k]$ can be constructed from:

$$u_{a,i}[k] = \begin{cases} u_i[k] & \text{if } |u_i[k]| \leq \hat{u}_i \\ \hat{u}_i \text{ sign}(u_i[k]) & \text{if } |u_i[k]| > \hat{u}_i \end{cases} \tag{3.60}$$

While the above control signal is guaranteed to stay within the admissible control bounds $\hat{u}$, it still is a rather aggressive control signal which could hit the control bounds far too often. Furthermore, stability is no longer guaranteed since far away from the switching surface the reaching law is, in general, no longer satisfied. Another approach is to change the parameter Φ as a function of for example $\sigma[k]$, i.e.:

$$\sigma[k+1] = \Phi(\sigma)\sigma[k] \tag{3.61}$$

As long as $\|\Phi(\sigma)\| < 1 \quad \forall \quad \sigma$ stability of the closed-loop is guaranteed. A two step method was introduced in (Golo and Milosavljević 2000) where the function $\Phi(\sigma)$

was chosen as:

$$\Phi(\sigma) = \begin{cases} \Phi_1 & \text{if } \|\sigma[k]\| \leq \Omega \\ \Phi_2 & \text{if } \|\sigma[k]\| > \Omega \end{cases} \tag{3.62}$$

where they chose Φ_1 to have all eigenvalues at zero and Φ_2 is chosen such that it has eigenvalues closer to, while still within, the unit circle. The advantage of this approach is that further away from the switching surface, the controller is less aggressive thereby reducing the chance of hitting the boundaries of the control signal.
The disadvantage of $\Phi(\sigma)$ as defined in (3.62) is that a step in the input signal at $\|\sigma[k]\| = \Omega$ will occur. Therefore, we propose a gradual change of the eigenvalues of $\Phi(\sigma)$. In the following Theorem an alternative is proposed.

Theorem 3.3.1 *The system (3.4),(3.5) under control of the SDSMC (3.18) with the constant Φ replaced by the diagonal matrix $\Phi(\sigma_x)$ defined by:*

$$\Phi_{i,i}(\sigma_{x,i}) = \begin{cases} 0 & \text{if } |\sigma_{x,i}[k]| < \Omega_{1,i} \\ \alpha_i \frac{|\sigma_{x,i}[k]| - \Omega_{1,i}}{\Omega_{2,i} - \Omega_{1,i}} & \text{if } \Omega_{1,i} \leq |\sigma_{x,i}[k]| \leq \Omega_{2,i} \\ \alpha_i & \text{if } |\sigma_{x,i}[k]| > \Omega_{2,i} \end{cases} \tag{3.63}$$

where:

$$0 < \alpha_i < 1 \qquad \Omega_{1,i} > \hat{f}_{m,i} + \delta_\Omega \qquad \Omega_{2,i} > \frac{\hat{f}_{m,i} + \delta_\Omega}{1 - \alpha_i} \qquad \Omega_{2,i} > \Omega_{1,i} \tag{3.64}$$

will converge to the quasi sliding mode band $\Delta = \|\hat{f}_m\|$, where $\hat{f}_m$ is the vector with the entries $\hat{f}_{m,i} = \max(f_{m,i})$ and δ_Ω being a small positive constant.
Proof : *Because of the diagonal structure of $\Phi(\sigma_x)$, the m entries of $\sigma_x[k]$ can be considered independently. Furthermore, if every entry of the switching function is driven into the region $|\sigma_{x,i}| < \hat{f}_{m,i}$, then the quasi sliding mode band is given by $\Delta = \|\hat{f}_m\|$. Therefore, the Theorem is proved if for each entry $\sigma_{x,i}$ the following condition is fulfilled: $|\sigma_{x,i}[k]| - |\sigma_{x,i}[k+1]| > \epsilon \ \forall \ |\sigma_{x,i}[k] > \hat{f}_{m,i}$ for some arbitrarily small positive constant ϵ.*
For $\sigma_{x,i}[k] > 0$, three regions can be identified which are treated separately:

- $\sigma_{x,i}[k] > \Omega_{2,i}$
  From equations (3.5), (3.18), and (3.63), it can be found that:

$$\sigma_{x,i}[k+1] = \alpha_i \sigma_{x,i}[k] + f_{m,i}(x,k) \tag{3.65}$$

  Substituting equation (3.65) into the condition $\sigma_{x,i}[k] - \sigma_{x,i}[k+1] > \epsilon$ and taking the worst case disturbance $\hat{f}_{m,i}$, we need to establish:

$$\sigma_{x,i}[k] - \alpha_i \sigma_{x,i}[k] - \hat{f}_{m,i} > \epsilon \tag{3.66}$$

  Using the fact that $\sigma_{x,i}[k] > \Omega_{2,i}$ and $\Omega_{2,i} > \frac{\hat{f}_{m,i} + \delta_\Omega}{1 - \alpha_i}$, condition (3.66) can be satisfied by choosing $\epsilon < \frac{\hat{f}_{m,i} + \delta_\Omega}{1 - \alpha_i}$.

- $\sigma_{x,i}[k] < \Omega_{1,i}$
  From equations (3.5), (3.18), and (3.63), it can be found that:

$$\sigma_{x,i}[k+1] = f_{m,i}(x,k) \tag{3.67}$$

  From which it directly follows that $\sigma_{x,i}[k]$ remains in the band $\sigma_{x,i}[k] < \hat{f}_{m,i}$.

- $\Omega_{1,i} \leq \sigma_{x,i}[k] \leq \Omega_{2,i}$
  From equations (3.5), (3.18), and (3.63), it can be found that:

$$\sigma_{x,i}[k+1] = \alpha_i \frac{\sigma_{x,i}[k] - \Omega_{1,i}}{\Omega_{2,i} - \Omega_{1,i}} \sigma_{x,i}[k] + f_{m,i}(x,k) \tag{3.68}$$

  Substituting the above equation into the condition $\sigma_{x,i}[k+1] < \sigma_{x,i}[k]$ and taking the worst case disturbance $\hat{f}_{m,i}$, we need to establish:

$$\alpha_i \frac{\sigma_{x,i}[k] - \Omega_{1,i}}{\Omega_{2,i} - \Omega_{1,i}} \sigma_{x,i}[k] + \hat{f}_{m,i} < \sigma_{x,i}[k] \tag{3.69}$$

  Condition (3.69) can be written as:

$$a\sigma_{x,i}^2[k] + b\sigma_{x,i}[k] + c < 0 \tag{3.70}$$

  where $a = \alpha_i$, $b = (1-\alpha_i)\Omega_{1,i} - \Omega_{2,i}$, and $c = (\Omega_{2,i} - \Omega_{1,i})(\hat{f}_{m,i} + \epsilon)$. It can easily be shown that $a > 0$, $b < 0$, and $c > 0$. Therefore, since the left hand side of condition (3.70) has no maximum for $\Omega_{1,i} < \sigma_{x,i}[k] < \Omega_{2,i}$, we only need to show that condition (3.70) is satisfied for $\sigma_{x,i}[k] = \Omega_{1,i}$ and $\sigma_{x,i}[k] = \Omega_{2,i}$ which automatically follows by successively substituting these values into (3.70) and choosing $\epsilon < \frac{\hat{f}_{m,i} + \delta_\Omega}{1-\alpha_i}$.

For $\sigma_{x,i}[k] < 0$ the proof goes along the same line. □

The above procedure can also be employed for a SDSMC based on the linear reaching law with disturbance estimation (see Section 3.1.5) in which case the results of Theorem 3.3.1 can be straightforwardly applied by replacing the bound $\hat{f}_m$ on the disturbance by the bound $\hat{\delta}_f$ on the rate of change of the disturbance. The procedure is also applicable to a ODSMC.

3.3.2 Adaptive Switching Gain

In Section 2.3.2 two adaptive switching gains were introduced for a CSMC. This section aims at developing an adaptive switching gain for a DSMC. First the two continuous time methods are discretised and their applicability for a DSMC is considered. Finally, a new adaptive switching gain is defined, based on the definition of sliding mode according to the Gao reaching law (see Section 3.1.3).
The Method I adaptation law defined for a CSMC (Section 2.3.2) can be directly translated to the discrete-time domain by discretising the integral function in Definition 2.3.6 by a summation, as can be seen in the following definition.

Definition 3.3.2 *The discrete-time Method I adaptation law is given by:*

$$\tilde{K}_s[k] = \tilde{K}_s[k-1] + \gamma\|\sigma[k]\| \tag{3.71}$$

where $\gamma \in \mathbb{R}$ is a small positive constant.

However, as was already pointed out, 'true' sliding motion is no longer possible in discrete-time. For this reason, the term $\|\sigma[k]\|$ will never converge to zero and consequently the adaptive switching gain will grow unbounded.
The Method II adaptation law introduced in Section 2.3.2 can be discretised similarly to the Method I adaptation law as can be seen in the following definition.

Definition 3.3.3 *The discrete-time Method II adaptation law is given by:*

$$\tilde{K}_s[k] = \tilde{K}_s[k-1] + \gamma\left(\|\sigma[k]\| - \psi\right) \tag{3.72}$$

where $\gamma \in \mathbb{R}$ and $\psi \in \mathbb{R}$ are small positive constants.

The above adaptation law still increases the gain $\tilde{K}_s$ until the system remains within the region $\|\sigma[k]\| < \psi$. However, since in discrete-time a "true" sliding mode is no longer achievable, the boundary region $\|\sigma[k]\| < \psi$ cannot be chosen arbitrarily small. If the boundary ψ is chosen smaller then achievable there will not exist a switching gain which is able to keep the system within the selected boundary region, and consequently the adaptive gain will grow unbounded. Hence, ψ has to be chosen carefully. Example 3.3.12 demonstrates this.
To overcome the problems for the discretised adaptation laws (Method I and II) a new adaptation method is introduced, based on the Gao reaching law (see Section 3.1.3). The three rules for discrete-time sliding mode according to (Gao et al. 1995). can be used as a guideline for an adaptive switching gain. Exploiting rule R_{II} the Method III adaptation rule introduced in the following definition can be constructed.

Definition 3.3.4 *The discrete-time Method III adaptation law is given by:*

$$\tilde{K}_{s,i}[k] = \left|\tilde{K}_{s,i}[k-1] + \gamma_i sign(\sigma_i[k])sign(\sigma_i[k-1])\right| \tag{3.73}$$

where $\gamma_i \in \mathbb{R}$ is some small positive constant which determines the speed of adaptation.

The Method III adaptation law can easily be explained in words. According to rule R_{II}, the switching surface should be crossed in every successive time step. Furthermore, it is known from Section 3.1.3 that in order to fulfill rule R_{II} the switching gain should be larger than some lower bound. Therefore, the above adaptation law increases the switching gain while the switching surface has not been crossed and decreases the gain if the switching surface has been crossed. Hence, the adaptation Method III aims at crossing the switching surface while minimizing the switching gain.

To study the Method III adaptation law closely we consider the system (3.4),(3.5) in closed-loop with controller (3.13). Substituting equation (3.13) into equation (3.5) results in:

$$\sigma_x[k+1] = \Phi\sigma_x[k] + f_m(x,k) - \begin{bmatrix} \tilde{K}_{s,1}[k]\ \mathrm{sign}(\sigma_{x,1}[k]) \\ \tilde{K}_{s,2}[k]\ \mathrm{sign}(\sigma_{x,2}[k]) \\ \vdots \\ \tilde{K}_{s,m}[k]\ \mathrm{sign}(\sigma_{x,m}[k]) \end{bmatrix} \tag{3.74}$$

Since, by design choice, the matrix Φ is diagonal, each entry of the switching function can be considered independently as:

$$\sigma_{x,i}[k+1] = \Phi_{i,i}\sigma_{x,i}[k] + f_{m,i}(x,k) - \tilde{K}_{s,i}[k]\ \mathrm{sign}(\sigma_{x,i}[k]) \tag{3.75}$$

The above equation will be used to determine the properties of the Method III adaptation law.
The first Theorem states that, regardless of the initial conditions (though bounded), the switching surface is crossed in finite time.

Theorem 3.3.5 *For any bounded $\sigma_{x,i}[0]$ and $\tilde{K}_{s,i}[0] \geq 0$ there exists a finite time p such that $sign(\sigma_{x,i}[p]) = -sign(\sigma_{x,i}[0])$.*
Proof : *If $\sigma_{x,i}[0] < 0$. Then if $\sigma_{x,i}[j] < 0, \quad \forall \quad 0 < j < p$, we can write for $\sigma_{x,i}[p]$:*

$$\sigma_{x,i}[p] = \Phi_{i,i}^p\sigma_{x,i}[0] + \tilde{K}_{s,i}[0]\sum_{j=0}^{p-1}\Phi_{i,i}^j + \sum_{j=1}^{p-1}(p-j)\Phi_{i,i}^{j-1}\gamma_i - \sum_{j=0}^{p-1}\Phi_{i,i}^{p-1-j}f_{m,i}(x,p) \tag{3.76}$$

Using the fact that $|f_{m,i}(x,p)| < \hat{f}_{m,i}$ and since $|\Phi_{i,i}| < 1$ it is true that $\sum_{j=0}^{l}\Phi_{i,i}^j < \frac{1}{1-\Phi_{i,i}}$. Using these facts the following expression can be obtained:

$$\sigma_{x,i}[p] > \Phi_{i,i}^p\sigma_{x,i}[0] + \tilde{K}_{s,i}[0]\sum_{j=0}^{p-1}\Phi_{i,i}^j + \sum_{j=1}^{p-1}(p-j)\Phi_{i,i}^{j-1}\gamma_i - \frac{1}{1-\Phi_{i,i}}\hat{f}_{m,i} \tag{3.77}$$

Looking at the righthand terms, we see that the negative terms $\Phi_{i,i}^p\sigma_{x,i}[0]$ and $-\frac{1}{1-\Phi_{i,i}}\hat{f}_{m,i}$ are bounded. The positive term $\tilde{K}_{s,i}[0]\sum_{j=0}^{p-1}\Phi_{i,i}^j$ is bounded as well, but the positive term $\sum_{j=1}^{p-1}(p-j)\Phi_{i,i}^{j-1}\gamma_i$ is growing with p. Therefore, there exists some time instant p for which the righthand side, and consequently $\sigma_{x,i}[p]$, is positive.
For the case where $\sigma_{x,i}[0] > 0$ the proof follows along the same lines. □

The above theorem states that the sliding surface is crossed in finite time. As the next Theorem states, this also implies that the switching gain has become larger than the disturbance.

Theorem 3.3.6 *If* $sign(\sigma_{x,i}[p+1] = -sign(\sigma_{x,i}[p])$ *and* $sign(f_{m,i}(x,p)) = sign(\sigma_{x,i}[p])$ *then* $\tilde{K}_{s,i}[p] > f_{m,i}(x,p)$.
Proof : *Rewriting equation (3.75) results in:*

$$\tilde{K}_{s,i}[p] = \frac{\Phi_{i,i}\sigma_{x,i}[p]}{sign(\sigma_{x,i}[p])} - \frac{\sigma_{x,i}[p+1]}{sign(\sigma_{x,i}[p])} + \frac{f_{m,i}(x,p)}{sign(\sigma_{x,i}[p])} \tag{3.78}$$

Since $sign(\sigma_{x,i}[p+1]) = -sign(\sigma_{x,i}[p])$ *and* $0 \leq \Phi_{i,i} < 1$ *it follows that:*

$$\tilde{K}_{s,i}[p] = |\Phi_{i,i}\sigma_{x,i}[p]| + |\sigma_{x,i}[p+1]| + \frac{f_{m,i}(x,p)}{sign(\sigma_{x,i}[p])} \tag{3.79}$$

By assumption $sign(f_{m,i}(x,p)) = sign(\sigma_{x,i}[p])$ *hence* $\frac{f_{m,i}(x,p)}{sign(\sigma_{x,i}[p])} = |f_{m,i}(x,p)|$. *Furthermore,* $|\sigma_{x,i}[p]| > 0$ *and* $|\sigma_{x,i}[p+1]| > 0$ *leading to* $\tilde{K}_{s,i}[p] > |f_{m,i}(x,p)|$. □

Theorem 3.3.5 states that the system will cross the switching surface in finite time, starting from any initial condition. Consequently, the system will cross the switching surface over and over again. Then Theorem 3.3.6 states that under the condition that at the moment of crossing the switching surface the sign of the disturbance is the same as the sign of the switching function, the switching gain is larger than the absolute value of the disturbance.Therefore, it may be concluded, especially for slowly varying disturbances, that the switching gain will pass some lower bound. Before it is shown that the switching gain does not grow unbounded, the notion of a p-cycle is introduced in the next Definition:

Definition 3.3.7 *With a* p-cycle *it is meant that* $sign(\sigma_{x,i}[k]) = sign(\sigma_{x,i}[k+p])$, *while* $sign(\sigma_{x,i}[k]) = -sign(\sigma_{x,i}[k+i])$ $\forall$ $i = \{1...p-1\}$. □

The value of the switching function after a p-cycle can easily be determined, which is described in the next Lemma.

Lemma 3.3.8 *Given the adaptation law (3.73), the value of* $\tilde{K}_{s,i}[k+p]$ *after a p-cycle will be:*

$$\tilde{K}_{s,i}[k+p] = \tilde{K}_{s,i}[k] + (p-4)\gamma_i \tag{3.80}$$

Proof : *Within every p-cycle,* $\sigma_{x,i}[k]$ *changes sign only twice. All other signs will be equal. This means that* γ_i *is subtracted twice from* $\tilde{K}_{s,i}[k]$ *and added* $p-2$ *times to* $\tilde{K}_{s,i}[k]$, *which adds effectively* $(p-4)\gamma_i$ *to* $\tilde{K}_{s,i}[k]$. □

Clearly, the above Lemma states that the switching gain over one p-cycle is:

- decreasing for $p < 4$
- constant for $p = 4$
- increasing for $p > 4$

It can be found in Section 3.1.3 that if the switching gain is larger then some value, specified by condition (3.14), the closed-loop will go into a 2-cycle. For 2-cycles, the switching gain is decreasing and therefore the switching gain is bounded from above.
In the special case of a constant disturbance, it can be expected that the system will settle down in some steady-state behaviour. From Lemma 3.3.8 it can be concluded that the switching gain remains constant in an average sense in a 4-cycle. To come to an upper and lower bound for the switching gain in a 4-cycle first the following Lemma is given.

Lemma 3.3.9 *Consider the system (3.75) with the constant disturbance* $f_{m,i} = -\hat{f}_{m,i}\ \forall\ k$. *If* $\sigma_{x,i}[k-1] < 0$, $\sigma_{x,i}[k] > 0$ *and* $\gamma_i \ll \hat{f}_{m,i}$, *then* $\sigma_{x,i}[k+1] < 0$.
Proof : *By assumption* $\sigma_{x,i}[k-1] < 0$, *leading with equation (3.75) to:*

$$\sigma_{x,i}[k] = \Phi_{i,i}\sigma_{x,i}[k-1] + \tilde{K}_{s,i}[k-1] - \hat{f}_{m,i} \tag{3.81}$$

Because $\sigma_{x,i}[k-1]$ *and* $\sigma_{x,i}[k]$ *have opposite signs it follows that* $\tilde{K}_{s,i}[k] = \tilde{K}_{s,i}[k-1] - \gamma$, *therefore* $\sigma_{x,i}[k+1]$ *can be determined to be:*

$$\sigma_{x,i}[k+1] = \Phi_{i,i}^2\sigma_{x,i}[k-1] - (1-\Phi_{i,i})\tilde{K}_{s,i}[k-1] - (1+\Phi_{i,i})\hat{f}_{m,i} + \gamma \tag{3.82}$$

All right hand terms in equation (3.82) are negative, except the term γ. *By assumption* $\gamma_i \ll \hat{f}_{m,i}$ *from which, together with equation (3.82), immediately follows that* $\sigma_{x,i}[k+1] < 0$. □

The following Lemma relates the value of the switching gain to length of the p-cycle.

Lemma 3.3.10 *Consider the system (3.75) with the constant disturbance* $f_{m,i} = -\hat{f}_{m,i}\ \forall\ k$ *and* $\gamma_i \ll \hat{f}_{m,i}$. *If* $\sigma_{x,i}[k-1] < 0$ *and* $\sigma_{x,i}[k] > 0$ *then* $\sigma_{x,i}[k+p^*] > 0$ *where the smallest* p^* *can be found from:*

- $p^* = 2$ *if:*

$$\tilde{K}_{s,i}[k] > \frac{-\Phi_{i,i}^2\sigma_{x,i}[k] + (1+\Phi_{i,i})\hat{f}_{m,i} + \gamma_i}{1-\Phi_{i,i}} \tag{3.83}$$

- p^* *is the smallest p satisfying:*

$$\tilde{K}_{s,i}[k] > \frac{-\Phi_{i,i}^p\sigma_{x,i}[k] + \sum_{j=0}^{p-1}\Phi_{i,i}^j\hat{f}_{m,i} + (\Phi_{i,i}^{p-2} - \sum_{j=1}^{p-2}\Phi_{i,i}^{j-1}(p-j-1)\gamma_i}{\sum_{j=0}^{p-2}\Phi_{i,i}^j - \Phi_{i,i}^{p-1}} \tag{3.84}$$

Proof : *According to Lemma 3.3.9* $\sigma_{x,i}[k+1]$ *is always negative, so* $p \geq 2$. *Because* $\sigma_{x,i}[k]$ *and* $\sigma_{x,i}[k+1]$ *have opposite signs we can find* $\tilde{K}_{s,i}[k+1] = \tilde{K}_{s,i}[k] - \gamma_i$. *For* $\sigma_{x,i}[k+2]$ *we can write:*

$$\sigma_{x,i}[k+2] = \Phi_{i,i}^2\sigma_{x,i}[k] - (1+\Phi_{i,i})\hat{f}_{m,i} + (1-\Phi_{i,i})\tilde{K}_{s,i}[k] - \gamma_i \tag{3.85}$$

which leads to the condition for $p^* = 2$ *by the assumption that* $\sigma_{x,i}[k+2] > 0$ *and consequently rewriting for* $\tilde{K}_{s,i}$. *The subsequent values for* $\sigma_{x,i}[k+p]$ *(while* $\sigma_{x,i}[k+j] < 0, \quad \forall \quad j = 1 \ldots p-1$ *and* $p > 2$*) can be found from:*

$$\begin{aligned} \sigma_{x,i}[k+p] &= \Phi_{i,i}^{p}\sigma_{x,i}[k] - \textstyle\sum_{j=0}^{p-1}\Phi_{i,i}^{j}\hat{f}_{m,i} - \left(\Phi_{i,i}^{p-1} - \textstyle\sum_{j=0}^{p-2}\Phi_{i,i}^{j}\right)\tilde{K}_{s,i}[k] \\ &\quad - \left(\Phi_{i,i}^{p-2} - \textstyle\sum_{j=1}^{p-2}\Phi_{i,i}^{j-1}(p-j-1)\right)\gamma_i \end{aligned} \tag{3.86}$$

which leads to condition for $p^* > 2$. □

From Lemma 3.3.10 it can be concluded that for smaller switching gains $\tilde{K}_{s,i}[k]$ the system will be in a longer p-cycle than for larger switching gains. The next Theorem gives the upper and lower bound of the switching gain in a 4-cycle for constant disturbances.

Theorem 3.3.11 *Consider the system (3.75) with a constant disturbance* $-\hat{f}_{m,i}$ *and* $0.58 < \Phi_{i,i} < 1$. *In steady-state the adaptive gain* $\tilde{K}_{s,i}[k]$ *(where* k *is the time instant where* $\sigma_{x,i}[k] > 0$*) will be in the region:*

$$\tilde{K}_{l,i} < \tilde{K}_{s,i}[k] < \tilde{K}_{u,i} \tag{3.87}$$

where:

$$\begin{aligned} \tilde{K}_{l,i} &= \tfrac{1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}{1+\Phi_{i,i}+\Phi_{i,i}^2-\Phi_{i,i}^3}\tilde{f}_{m,i} - \tfrac{1-\Phi_{i,i}^2}{1+\Phi_{i,i}+\Phi_{i,i}^2-\Phi_{i,i}^3}\gamma_i \\ \tilde{K}_{u,i} &= \tfrac{1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}{1+\Phi_{i,i}-\Phi_{i,i}^2+\Phi_{i,i}^3}\tilde{f}_{m,i} + \tfrac{\Phi_{i,i}-\Phi_{i,i}^3}{1+\Phi_{i,i}-\Phi_{i,i}^2+\Phi_{i,i}^3}\gamma_i \end{aligned} \tag{3.88}$$

Proof : *In steady-state the system will be in a 4-cycle and, assuming* $\sigma_{x,i}[k] > 0$*:*

$$\begin{aligned} \sigma_{x,i}[k+1] &= \Phi_{i,i}\sigma_{x,i}[k] - \tilde{K}_{s,i}[k] - \hat{f}_{m,i} \\ \sigma_{x,i}[k+2] &= \Phi_{i,i}^2\sigma_{x,i}[k] + (1-\Phi_{i,i})\tilde{K}_{s,i}[k] - (1+\Phi_{i,i})\hat{f}_{m,i} - \gamma_i \\ \sigma_{x,i}[k+3] &= \Phi_{i,i}^3\sigma_{x,i}[k] + (1+\Phi_{i,i}-\Phi_{i,i}^2)\tilde{K}_{s,i}[k] \\ &\quad -(1+\Phi_{i,i}+\Phi_{i,i}^2)\hat{f}_{m,i} - \Phi_{i,i}\gamma_i \\ \sigma_{x,i}[k+4] &= \Phi_{i,i}^4\sigma_{x,i}[k] + (1+\Phi_{i,i}+\Phi_{i,i}^2-\Phi_{i,i}^3)\tilde{K}_{s,i}[k] \\ &\quad -(1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3)\hat{f}_{m,i} + (1-\Phi_{i,i}^2)\gamma_i \end{aligned} \tag{3.89}$$

As can be found in Lemma 3.3.9, $\sigma_{x,i}[k+1] < 0$. *Therefore, a 4-cycle also implies* $\sigma_{x,i}[k+2] < 0$, $\sigma_{x,i}[k+3] < 0$, *and* $\sigma_{x,i}[k+4] > 0$, *leading to the conditions:*

$$\begin{aligned} \tilde{K}_{s,i}[k] &< \tfrac{1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}{-1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}\hat{f}_{m,i} - \tfrac{\Phi_{i,i}-\Phi_{i,i}^3}{-1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}\gamma_i \\ \tilde{K}_{s,i}[k] &< \tfrac{1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}{1-\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}\hat{f}_{m,i} + \tfrac{1-\Phi_{i,i}^2}{1-\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}\gamma_i \\ \tilde{K}_{s,i}[k] &< \tfrac{1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}{1+\Phi_{i,i}-\Phi_{i,i}^2+\Phi_{i,i}^3}\hat{f}_{m,i} + \tfrac{\Phi_{i,i}-\Phi_{i,i}^3}{1+\Phi_{i,i}-\Phi_{i,i}^2+\Phi_{i,i}^3}\gamma_i \\ \tilde{K}_{s,i}[k] &> \tfrac{1+\Phi_{i,i}+\Phi_{i,i}^2+\Phi_{i,i}^3}{1+\Phi_{i,i}+\Phi_{i,i}^2-\Phi_{i,i}^3}\hat{f}_{m,i} - \tfrac{1-\Phi_{i,i}^2}{1+\Phi_{i,i}+\Phi_{i,i}^2-\Phi_{i,i}^3}\gamma_i \end{aligned} \tag{3.90}$$

Taking the lowest upper bound and the lower bound completes the proof. □

In Lemma 3.3.10 it is shown for the case of constant disturbances that shorter p-cycles are achieved by larger switching gains, and conversely, longer p-cycles for smaller gains. By Lemma 3.3.8, the switching will be increased for p-cycles with $p > 4$ and decreased for $p < 2$ decreased. Therefore, assuming that γ is chosen sufficiently small (ideally $\gamma << \hat{f}_{m,i}$), the switching gain will always be driven into the region given by Theorem 3.3.11.
In the following, two possible extensions to the Method III adaptation method are proposed.

- For constant disturbances it can be found that the switching gain will settle down in the region given by Theorem 3.3.11. Within this region, no further adaptation takes place. However it is desirable to converge to the smallest switching gain in this region. This can be accomplished by altering the Method III adaptation procedure of Definition 3.73 to:

$$\tilde{K}_{s,i}[k] = \left| \tilde{K}_{s,i}[k-1] + \gamma_i \text{sign}(\sigma_i[k]) \text{sign}(\sigma_i[k-1]) - \beta_i \right| \tag{3.91}$$

  where $0 < \beta_i << \gamma_i$. By the fact that β_i is much smaller than γ_i the adaptation procedure still works approximately the same. However, the adaptive gain will decrease slowly in the band given in Theorem 3.3.11 thereby decreasing towards the the smallest value in the given band.

- It was already mentioned in Section 3.1.6 that the discontinuous control results in a severe chattering behavior of the input and the switching function. This effect can be reduced considerably by changing the discontinuous control term $u_{d,i}[k] = -\tilde{K}_s \text{ sign}(\sigma[k])$ to:

$$u_{i,d}[k] = \begin{cases} -\tilde{K}^+_{s,i}[k] & \text{if} \quad \sigma_i[k] \geq 0 \\ \tilde{K}^-_{s,i}[k] & \text{if} \quad \sigma_i[k] < 0 \end{cases} \tag{3.92}$$

  and letting the switching gains $\tilde{K}^+_s$ and $\tilde{K}^-_s$ be either positive or negative, whatever suits the circumstances best. Studying the result one could result that a much smaller switching term superpositioned on a non-switching term which approximately has the size of the disturbance. The Method III adaptation law can again be used to determine the switching gains, which is accomplished by:

$$\begin{array}{lrcl} \text{if } \sigma_i[k-1] \geq 0 : & \tilde{K}^+_{s,i}[k] & = & \tilde{K}^+_{s,i}[k-1] + \gamma_i \text{sign}(\sigma_i[k]) \text{sign}(\sigma_i[k-1]) \\ & \tilde{K}^-_{s,i}[k] & = & \tilde{K}^-_{s,i}[k-1] \\ \text{if } \sigma_i[k-1] < 0 : & \tilde{K}^-_{s,i}[k] & = & \tilde{K}^-_{s,i}[k-1] + \gamma_i \text{sign}(\sigma_i[k]) \text{sign}(\sigma_i[k-1]) \\ & \tilde{K}^+_{s,i}[k] & = & \tilde{K}^+_{s,i}[k-1] \end{array} \tag{3.93}$$

We conclude this section with two simulation examples to illustrate and compare the developed adaptation laws.

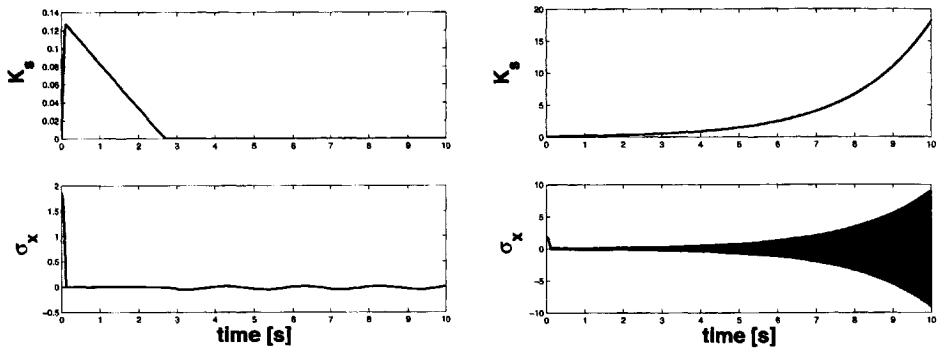


Fig. 3.2: Simulation results for the discretised Method II adaptation law. Figure (a) presents the switching gain (top) and the switching function for $\psi = 0.1$ (bottom), Figure (b) presents the same for $\psi = 0.001$.

Example 3.3.12 *As an example for the adaptive switching gains the pendulum example of Chapter 2 is used. The continuous-time model is linearised around the origin and discretised with a sampling period of* $T_s = 5e^{-3}$ *s. The system can then be described by the state space model* $x[k+1] = Ax[k] + Buk$, *where* A_d *and* B_d *are given by:*

$$A_d = \begin{bmatrix} 1.0 & 5.0\ e^{-3} \\ -1.2\ e^{-3} & 1.0 \end{bmatrix} \quad B_d = \begin{bmatrix} 0.0 \\ 5.0\ e^{-3} \end{bmatrix}$$

The unmodeled nonlinearity can now be considered as a matched disturbance (the term $f_m[k, x, u]$ *in Equation (3.5)). The switching function for the SDSMC is obtained by LQR design, resulting in:*

$$\sigma_x[k] = Se[k] = [-6.23 \quad -0.98]e[k]$$

where $e[k] = x_d[k] - x[k]$, $x_d[k]$ *being the desired state.*
The control law is taken as equation (3.13) and the switching gain K_s *is now adaptive (i.e. replaced by* $\tilde{K}_s[k]$*). The system has to track the following signal:*

$$x_d = \begin{bmatrix} \sin(\omega t) \\ \omega \cos(\omega t) \end{bmatrix}$$

so the angle and the angular velocity of the pendulum should be perfect sine- and cosine functions of the time ($\omega = \frac{2\pi}{20}$). *The controller is tested on a (simulated) system with an error in the parameters given by* $a_1^s = \frac{1}{2}a_1^m$ *and* $a_2^s = 2a_2^m$ *(where the superscript s stands for the (simulated) system and the superscript m stands for the model which is used for the controller design).*
The simulation results for the Method II adaptation law are given for the parameters $\psi = 0.1$ *(Figure 3.2.a) and* $\psi = 0.001$ *(Figure 3.2.b). The results for* $\psi = 0.1$ *are satisfactory, however, for* $\psi = 0.001$ *the results are unstable. The chosen region is*

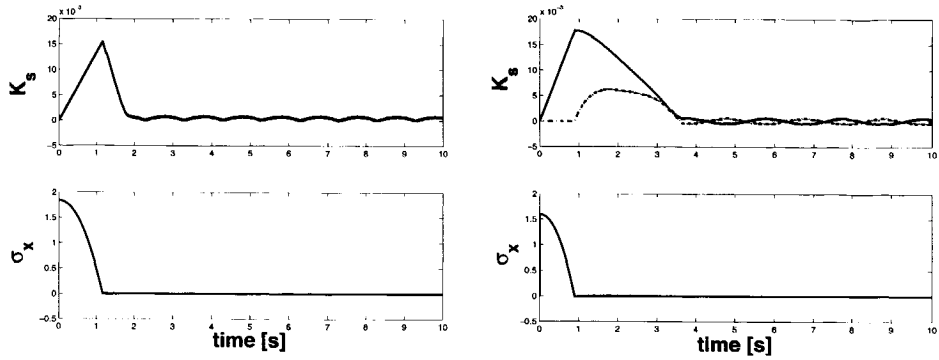


Fig. 3.3: Simulation results for the Method III adaptation law. Figure (a) presents the switching gain (top) and the switching function for the standard Method III defined by equation (3.73) (bottom), Figure (b) presents the same for the Method III procedure as defined in equations (3.92) and (3.93).

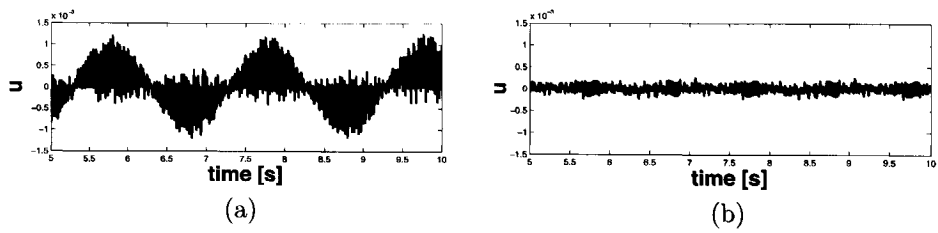


Fig. 3.4: Simulation results for the Method III adaptation law. Figure (a) presents the switching function for the normal Method III for $t > 5$, Figure (b) presents the same for the Method III procedure as defined in equations (3.92) and (3.93).

smaller then the smallest possible quasi sliding mode band, in which case there does not exist a switching gain which satisfies the objective of the adaptation law II. Figure 3.3 presents the results for the Method III adaptation law. Both the normal Method III procedure (Figure 3.3.a) and the extended Method III procedure given by equations (3.92) and (3.93) (Figure 3.3.b) work well. The difference in performance can be seen by zooming in on the switching function as presented in Figure 3.4. The extended Method III adaptation procedure remains closer to the switching surface.

Example 3.3.13 *To illustrate the Method III adaptation law for the multiple input case we have chosen the hover control of a Bell 205 helicopter. The simulation model can be obtained from for example (Pieper et al. 1996, Trentini and Pieper 2001). The linear model is represented by the differential state equation:*

$$\dot{x}(t) = Ax(t) + Bu(t)$$

with the matrices:

$$A = \begin{bmatrix} 0 & 0.03 & 0.18 & -0.01 & -0.42 & 0.08 & -9.81 & 0 \\ -0.10 & -0.39 & 0.09 & -0.10 & -0.72 & 0.68 & 0 & 0 \\ 0.01 & -0.01 & -0.19 & 0 & 0.23 & 0.04 & 0 & 0 \\ 0.02 & 0 & -0.41 & -0.05 & -0.27 & 0.27 & 0 & 9.81 \\ 0.03 & -0.02 & -0.88 & -0.04 & -0.57 & 0.14 & 0 & 0 \\ -0.01 & -0.02 & -0.06 & 0.07 & -0.32 & -0.71 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.08 & 0.13 & 0 & 0 \\ -1.17 & 0.04 & 0 & 0.01 \\ 0 & -0.07 & 0 & 0.01 \\ -0.04 & 0 & 0.11 & 0.19 \\ -0.04 & 0 & 0.22 & 0.17 \\ 0.17 & 0 & 0.03 & -0.47 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The states and input variables are given by:

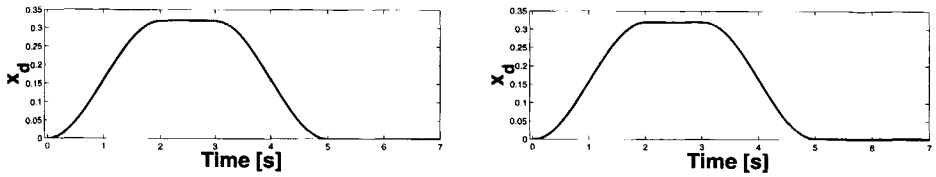
$$x(t) = \begin{bmatrix} \textit{forward velocity} \\ \textit{vertical velocity} \\ \textit{pitch rate} \\ \textit{lateral velocity} \\ \textit{roll rate} \\ \textit{yaw rate} \\ \textit{pitch attitude} \\ \textit{roll attitude} \end{bmatrix} \qquad u(t) = \begin{bmatrix} \textit{collective} \\ \textit{longitudinal cyclic} \\ \textit{lateral cyclic} \\ \textit{tail rotor collective} \end{bmatrix}$$

The model is discretised (zero order hold) with sampling frequency $T_s = 5ms$. The parameters of the sliding mode controller are given by Φ being a $m \times m$ zero matrix, and:

$$S = \begin{bmatrix} 4.28e^{-2} & -0.17 & -1.67e^{-2} & 2.80e^{-3} & -1.30e^{-3} & -2.90e^{-3} & 2.00e^{-4} & 5.00e^{-4} \\ 1.18 & 7.88e^{-2} & -0.63 & 6.40e^{-3} & -0.10e^{-3} & -1.01e^{-2} & 1.32e^{-2} & 0.10e^{-2} \\ -8.40e^{-2} & 2.00e^{-2} & -2.20e^{-2} & 0.30 & 0.71 & 0.38 & -0.30e^{-2} & 1.83e^{-2} \\ 1.96e^{-2} & -5.84e^{-2} & 0.31e^{-2} & 8.89e^{-2} & 0.81e^{-2} & -0.39 & -0.12e^{-2} & 0.20e^{-3} \end{bmatrix}$$

where S has been obtained by LQR design. The nonzero components of the desired state, the vertical and lateral velocity (i.e. 2^{nd} and 4^{th} components of the state vector), are depicted in Figure 3.5. In simulation, the system matrix A is perturbed by the matrix δ_A (i.e. $A_s = A + \delta_A$) given by:

$$\delta_A = 1e^{-4} * \begin{bmatrix} 0.46 & 0.18 & 0.81 & 0.02 & 0.20 & 0.71 & 0.13 & 0.82 \\ 0.02 & 0.40 & 0.01 & 0.75 & 0.67 & 0.43 & 0.70 & 0.68 \\ 0.82 & 0.93 & 0.14 & 0.44 & 0.83 & 0.31 & 0.34 & 0.84 \\ 0.44 & 0.92 & 0.21 & 0.93 & 0.02 & 0.19 & 0.84 & 0.71 \\ 0.62 & 0.41 & 0.20 & 0.47 & 0.68 & 0.20 & 0.82 & 0.36 \\ 0.79 & 0.89 & 0.60 & 0.42 & 0.37 & 0.68 & 0.55 & 0.31 \\ 0.92 & 0.06 & 0.27 & 0.85 & 0.83 & 0.30 & 0.45 & 0.38 \\ 0.74 & 0.35 & 0.20 & 0.52 & 0.50 & 0.54 & 0.86 & 0.56 \end{bmatrix}$$



(a) (b)

Fig. 3.5: Desired trajectories for the vertical velocity (a) and (b) lateral velocity. All other desired state components are zero.

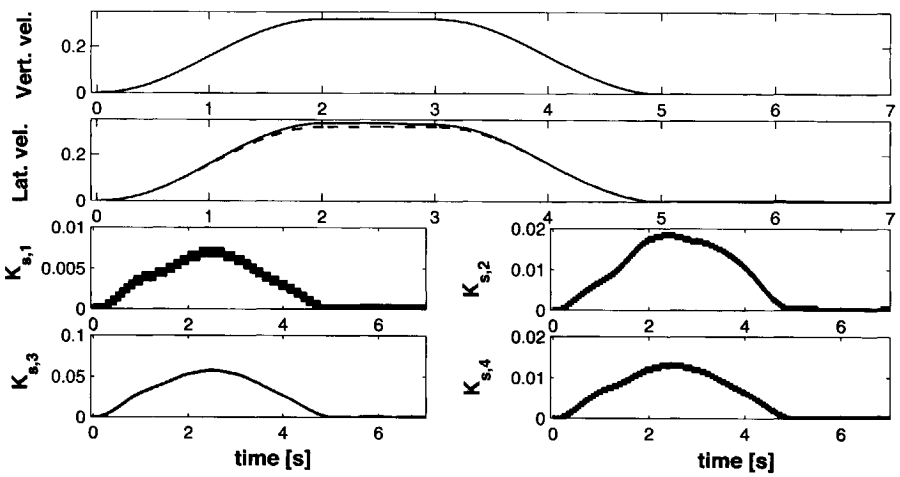


Fig. 3.6: Simulation results for the Method III adaptation law. the two top figures represent the desired and achieved vertical and lateral velocity respectively. The four bottom figures represent the switching gain components.

The control law is taken as equation (3.13) where in this case the switching gains are adaptive, given by equation (3.73). Figure 3.6 presents the simulation results for the Method III adaptation law with $\gamma = 5e^{-5}$. *As can be seen in Figure 3.6, the switching gains remain bounded and the desired state is tracked with high accuracy.*

Figure 3.7 presents the simulation results for the extended Method III given by equations (3.92) and (3.93). In this case, the switching gains are again stable. The main difference between the Method III adaptation law (given by equation (3.73)) and the extended Method III adaptation law (given by equations (3.92) and (3.93)) can be seen in the input signals presented in Figure 3.8. The advantage of the extended Method III is obvious, the high frequency component of the input signal for the regular Method III has largely been suppressed.

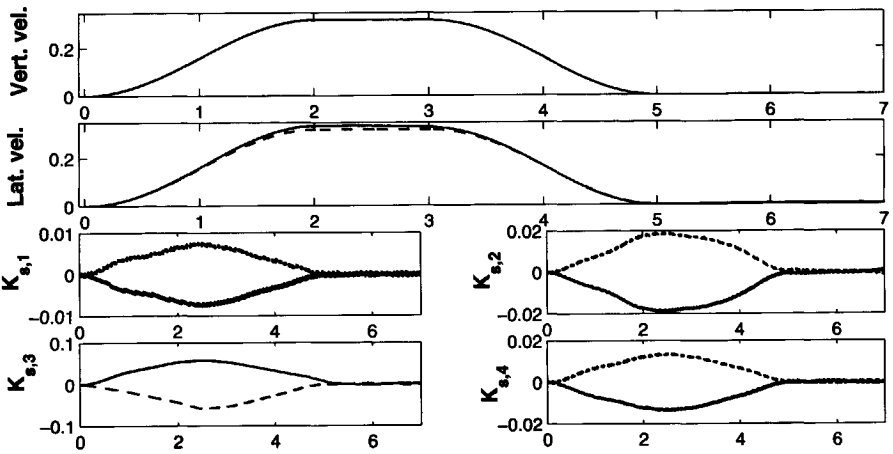


Fig. 3.7: Simulation results for the extended Method III adaptation law. the two top figures represent the desired and achieved vertical and lateral velocity respectively. The four bottom figures represent the switching gain components.

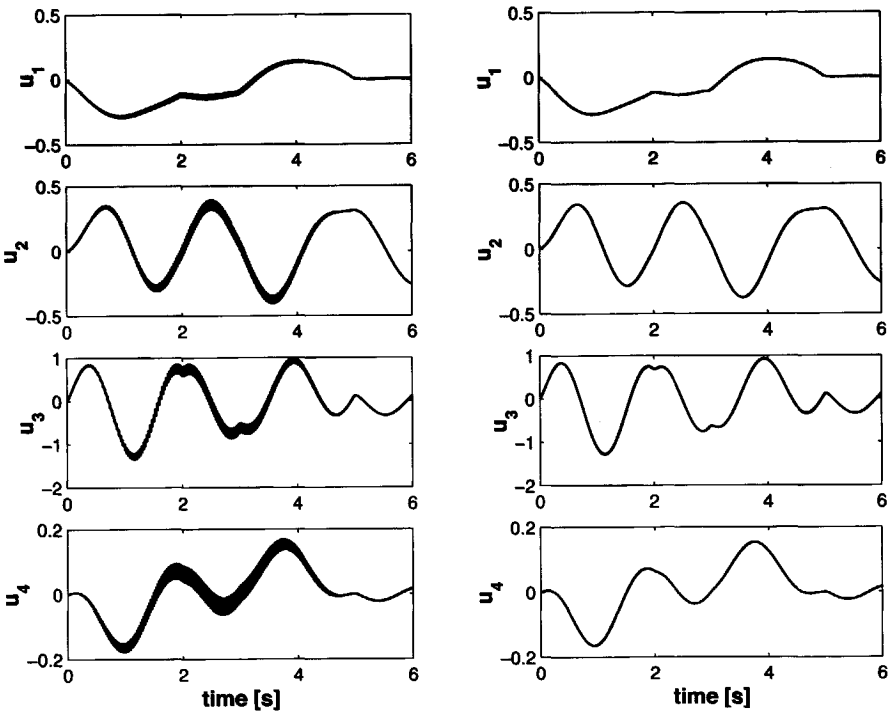


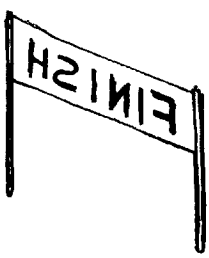
Fig. 3.8: Comparison of the input signals for the Model III adaptation method (left plots) and the extended Model III adaptation method (right plots).

3.4 Summary

This chapter introduces a design procedure for a DSMC. It is divided into the same design steps as for the CSMC. In fact, the first design step (computation of the sliding surface) is similar to the continuous-time case. The difference between the discrete time case and the continuous-time case is the region where the poles should be located (within the unit disc instead of the left half plane). However, the procedure to accomplish this is similar. Therefore, this chapter is focused on finding a controller which ensures that the closed-loop system is driven into a sliding mode. In Section 3.1 an overview is given of existing reaching laws for a SDSMC together with their control laws. It turns out that perfect a sliding mode can no longer be attained. Instead, the closed-loop system is driven into a quasi sliding mode. It is shown that the linear reaching law leads to the smallest quasi sliding mode band, especially if it is extended with a disturbance estimator. Then Section 3.2 introduces three possible controller setups for an ODSMC. The first, known as the direct linear controller, is based on the linear reaching law, which is equivalent to the OCSMCs known from literature. Then the direct linear controller is extended with a disturbance estimator and a reduced order state estimator respectively. Finally, Section 3.3 describes two practical considerations. The first, Section 3.3.1, is the idea of introducing a variable reaching law. The second practical consideration is the use of a adaptive switching gain. The Method I and II adaptation laws known from for the continuous-time case are discretised such that they are applicable to a DSMC. Then a third adaptation method is derived from the Gao reaching law.
The linear reaching law with disturbance estimation has been published in (Monsees and Scherpen 2001*a*, Monsees and Scherpen 2001*b*, Monsees and Scherpen 2001*c*). The design procedure for an ODSMC has been published in (Monsees and Scherpen 2001*a*, Monsees and Scherpen 2001*b*, Monsees and Scherpen 2001*d*). The idea of the variable reaching law has not been published. Finally, the results for the adaptive switching gain have been published in (Monsees and Scherpen 2002*a*, Monsees and Scherpen 2000).

4

Feedforward Control



Precision output tracking of sampled data systems still forms an important control problem. To obtain both good tracking and robustness, the control problem is typically split into two parts. The first part is the computation of the feedforward controller which ideally results in perfect tracking for the nominal system. The second part of the control problem is the design of a feedback controller which is added to the system to increase robustness against parameter variations and disturbances. This chapter is devoted to the design of a feedforward controller.
The target tracking problem for the following system is considered:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] + Du[k] \end{aligned} \tag{4.1}$$

with $x \in \mathbb{R}^n$, $y \in \mathbb{R}^p$, $u \in \mathbb{R}^m$, and the matrices (A, B, C, D) of appropriate size. The system is assumed to be controllable, observable, and stable. The latter condition can always be met by adding a feedback loop which stabilizes the system. Furthermore, it is assumed that the system is square, i.e. $p = m$. In the case that the system has more outputs than inputs (i.e. $p > m$), the system can be made square by defining a new output $y'[k] = Py[k]$, $P \in \mathbb{R}^{m \times p}$ and $\text{Rank}\{P\} = m$, for which the input signal is computed.
The problem of target tracking can now be defined as generating a bounded input $u[k]$, based on the obtained model from the system, which results in an output $y[k]$ which is equal to the desired output $y_d[k]$. This particular generated input will be called the desired input $u_d[k]$. This chapter studies the design of an input signal which gives perfect tracking for the model. The solution to this design problem seems to be trivial. Let us suppose for the moment that the system (4.1) has one

input and one output. Replacing the state space representation (4.1) by the transfer function description results in:

$$Y(z) = G_p(z)U(z) \tag{4.2}$$

where $G_p(z)$ can be found from:

$$G_p(z) = C\left(zI - A\right)^{-1}B + D \tag{4.3}$$

From equation (4.2) the desired input signal can easily be determined to be:

$$U(z) = G_p^{-1}(z)Y_d(z) \tag{4.4}$$

Of course, it might very well be the case that the transfer function $G_p^{-1}(z)$ is not proper. This can be circumvented by adding r (r being the relative degree of $G_p(z)$) poles at $z = 0$, leading to:

$$U(z) = G_p^{-1}(z)z^{-r}Y_d(z) \tag{4.5}$$

The effect of the added poles at $z = 0$ is a pure delay of r samples. In other words, the desired output should be known at least r samples in advance in order to obtain perfect tracking.

Unfortunately, the above procedure only leads to a stable input signal for minimum phase systems. The zeros of $G_p^{-1}(z)$ are the poles of $G_p(z)$ and conversely the poles of $G_p^{-1}(z)$ are the zeros of $G_p(z)$. Therefore, nonminimum phase systems (i.e. zeros outside the unit circle) have an unstable inverse. Choosing another output could solve the problem in some situations. In for example the flatness approach a new output is chosen such that it has no zeros at all. It is possible to define a trajectory for the flat output to transfer the original nonminimum phase output from one set-point to another, see for example (Fliess et al. 1998). Also in e.g. (Scherpen et al. 2000) a new output is selected without any unstable zeros for the case of an electrical network. Steering this new output to a set-point indirectly stabilizes the original output to the given desired value. While these methods have great potential for the stabilization of the system in some steady state, they are not applicable to an arbitrary desired output signal. While the flatness approach can be extended to periodical signals (Sira-Ramírez 1999), translating the trajectory of an arbitrary desired output trajectory (as opposed to a, possibly step-wise, constant or periodic desired output trajectory) to a desired trajectory for the flat output still leads to inversion of the unstable zeros. Therefore, the flatness approach does not solve the tracking problem of an arbitrary desired output trajectory.

A typical solution to the problem of finding a feedforward control signal is to formulate the tracking problem as an optimization problem. This is what happens in optimal control where the feedforward control problem is solved by minimizing a cost criterion like:

$$C(y,u) = \frac{1}{2}\sum_{k=0}^{\infty}\left\{(y[k] - y_d[k])^T Q_1 (y[k] - y_d[k]) + u^T[k]Q_2 u[k]\right\} \tag{4.6}$$

subject to the dynamics (4.1). The above cost-function can only be optimized if the desired output is known fully in advance. Even then, the procedure does not automatically lead to perfect tracking. The optimization procedure balances the costs of a tracking error and the costs of the applied control input. To achieve perfect tracking it might be tempting to set the matrix Q_2 to zero, thereby giving full attention to the tracking error. However, taking the input signal into account in the cost function ensures stability of the control signal, which is especially crucial in the case of nonminimum phase systems. A related method can be found in (Ghosh and Paden 2001) where the desired input signal is found via an iterative optimization procedure. Again the desired output signal should be known in advance, however the tracking error can be made arbitrary small.

Another solution is to define a Finite Impulse Response (FIR) filter to compensate for the nonminimum phase zeros. This approach can be found in for example (Gross and Tomizuka 1994) where they use a Taylor expansion of the unstable pole of the inverted system. Similarly, (Holmberg et al. 1995) directly define a FIR filter for which they determine the parameters such that the maximal undershoot is minimized. Just like the optimization approach this solution does not lead to perfect tracking. However, since the feedforward controller is constructed as a FIR filter, the preview time (i.e. the time that the desired output signal should be known in advance) is bounded. A disadvantage of both FIR filter approaches is that they only treat the case of one or two unstable zeros.

Another way to solve the problem is to split the inverse of the system into a series connection of a stable and an unstable part. The unstable part is approximated by a bounded preview filter which is designed such that the phase-shift is equal to the phase-shift of the unstable part and its gain is as close as possible to gain of the unstable part (Xia and Menq 1995). A similar procedure, called Steering Along the Zeros Control (SAZC), was introduced in (Marro and Hamano 1996) and (Marro and Marconi 1997). This method splits the inverse of the system into a stable and an unstable part by partial fraction decomposition. The stable inversion part is solved in the conventional manner whereas the unstable part is solved by translating the problem of finding the inverse of the unstable part to an initialization procedure. An input signal is generated which drives the system towards the required initial condition, without affecting the output. Just as the optimization approach, SAZC assumes to have full knowledge of the desired output, either in the past or the future. However, unlike the optimization solution, SAZC leads to perfect tracking. In the case where no full knowledge of the desired output signal is available, SAZC can still be employed leading to a decrease of tracking performance (Marconi et al. 2001). The SAZC method is described in detail in Section 4.1. Whereas zeros on the unit circle are excluded in the publications (Marro and Hamano 1996), (Marro and Marconi 1997), and (Marconi et al. 2001), we show that, under an additional condition, systems with zeros on the unit circle can be treated as well.

It is also possible to assume that there exists a dynamic model which describes the unknown input signal which leads to perfect tracking. This assumption is used in the Stable Dynamic Inversion (SDI) technique (George et al. 1999*a*, George et al. 1999*b*), where the unknown input signal is modeled as an integrator with random noise

as input. The system description is then augmented with the input model. For the augmented system a Kalman filter is constructed which recovers the desired augmented state from the desired output signal. The method does, in general, not lead to perfect tracking but instead leads to approximate tracking. However, it can be applied straightforwardly to MIMO systems, whereas all previous solutions are limited to SISO systems. Furthermore, in the case where the desired output signal is obtained from measurements the SDI procedure is capable of reducing the sensitivity to measurement noise. The SDI procedure is described in more detail in Section 4.2.

Another state-space approach can be found in (Devasia et al. 1996), called the Method of Dichotomies (MD). The MD brings the system into the Brunovsky canonical form and separates the stable- from the unstable zero dynamics. By assuming that the initial state and the final state are known, the solution for the zero dynamics can be found by solving a two-point boundary value problem. The MD again assumes full knowledge of the desired output signal and leads to perfect tracking, just like SAZC. However, the MD can be straightforwardly applied to MIMO systems, this in contrast to SAZC. The preview time can be limited in a similar manner as done for SAZC, again leading to a decrease of tracking performance. The performance of the Bounded Preview time Method of Dichotomies (BPMD) can be increased by the inclusion of a DSMC within the feedforward controller (leading to the Bounded Preview time Method of Dichotomies with Sliding Mode control (BPMDSM)). The state generated by the BPMD is used as the desired state for the SDSMC. If the SDSMC would achieve perfect tracking of this state then perfect tracking would be achieved because bounding the preview time only affects the zero-dynamics. Unfortunately, the generated state is not trackable leading to approximate tracking. However, as shown in a simulation example, the tracking performance is improved considerably. The MD is presented in Section 4.3. In Section 4.4 a simple simulation example is used to compare the performance of the BPMD, the SDI, and the BPMDSM. Finally Section 4.5 presents a summary.

4.1 Steering Along the Zeros Control

To study the Steering Along the Zeros Control technique (SAZC), the transfer function representation (4.2) is considered. It is assumed that the system is SISO and that the transfer function $G_p(z)$ can be written as:

$$G_p(z) = z^{-r}\frac{N(z)}{D(z)} \tag{4.7}$$

with $N(z) = p_1 + p_2 z + \ldots + p_{n_p} z^{n_p-1}$, $D(z) = p_1 + p_2 z + \ldots + p_{n_q} z^{n_q-1}$, and $n_q = n_p$. The constant r is known as the relative degree of the system. Therefore, by defining the signal $r[k] = y[k+r]$ (and consequently $R(z) = z^r Y(z)$) the following expression is obtained:

$$U(z) = \frac{D(z)}{N(z)}R(z) \tag{4.8}$$

For minimum phase systems, the above system is stable and the signal $u[k]$ leads to perfect tracking of $r[k]$. For nonminimum phase systems, the above representation is decomposed by partial fraction decomposition into:

$$U(z) = \underbrace{\frac{A(z)}{N_s(z)}R(z)}_{=U_s(z)} + \underbrace{\frac{B(z)}{N_u(z)}R(z)}_{=U_u(z)} \tag{4.9}$$

where the polynomial $N_s(z)$ contains all stable roots of $N(z)$ and $N_u(z)$ contains all unstable roots.
In order to find a control signal which leads to perfect tracking of the signal $R(z)$ it is sufficient to find the control input which leads to perfect tracking of the unit impulse function given by $R(z) = 1$. The control input to track an arbitrary signal $R(z)$ can later be found by convoluting the generated input signal in the time domain with the desired output signal (or by multiplication in the frequency domain).
Applying the unstable control term $U_u(z)$ for the unit impulse function to the system (4.7) results in:

$$Y_u(z) = \frac{B(z)N_s(z)}{D(z)} \tag{4.10}$$

Noticing that the denominator of the above expression is identical to the denominator of the free system, it can be argued that the nominator $B(z)N_s(z)$ represents an initial situation of the free response for the system $G_p(z)$. Therefore, if the system is steered to that initial situation then it is possible to track the given unit impulse signal. The control input which brings the system in the correct initial situation can be determined by the use of the following theorem:

Theorem 4.1.1 *(Marconi et al. 2001) Consider a discrete-time causal SISO system $G_p = N(z)/D(z)$ without any poles at zero. Let $u_u[k]$, $k \in [-\infty, -1]$, be an input signal that, starting from the zero state at $k = -\infty$, carries G_p at $k = 0$ to the state corresponding to equation (4.10), and let $y_u[k]$, $k \in [-\infty, -1]$, be the corresponding output signal. Let $U_u(z)$ and $Y_u(z)$ denote the reverse-time $\mathcal{Z}$-transform of the signals $u_u[-h]$, $y_u[-h]$, $h \in [1, \infty]$, defined by:*

$$U_u(z) = \sum_{k=-\infty}^{-1} u_u[k]z^k \qquad Y_u(z) = \sum_{k=-\infty}^{-1} y_u[k]z^k \tag{4.11}$$

Then, the following equality holds:

$$Y_u(z) = \frac{N(z^{-1})}{D(z^{-1})}U_u(z) - \frac{B(z^{-1})N_s(z^{-1})}{D(z^{-1})} \tag{4.12}$$

where the polynomials $\{N(z^{-1}), D(z^{-1}), B(z^{-1}), N_s(z^{-1})\}$ can be found from the polynomials $\{N(z), D(z), B(z), N_s(z)\}$ by simple replacement of z by z^{-1}. □

For the unit impulse function it is known that $y_u[-h] = 0$, $h \in [1, \infty]$ and consequently $Y_u(z) = 0$. Therefore, equation (4.12) can be rewritten as:

$$U_r(z) = \frac{B(z^{-1})}{N_u(z^{-1})} \tag{4.13}$$

Important to notice is that the time domain representation $u_r[k]$ of the above input signal is only nonzero for $k < 0$ whereas the target signal $r[k]$ is only nonzero for $k = 0$. Therefore, $u_r[k]$ represents the preactuation which is required to drive the system to the required initial situation.
For an arbitrary reference signal the preactuation signal can easily be obtained from the following convolution:

$$u_u[k] = \sum_{l=0}^{\infty} r[k]u_r[k-l] \tag{4.14}$$

with $u_r[k]$, $k \in [-\infty, -1]$ as defined in equation (4.13) and $u_r[k] = 0$, $k \in [0, \infty]$. The total input signal is given by:

$$u[k] = u_u[k] + u_s[k] \tag{4.15}$$

where $u_s[k]$ can be found as the inverse $\mathcal{Z}$-transform of $U_s(z) = \frac{A(z)}{N_s(z)}R(z)$.
The generated feedforward signal is anti-causal. For a minimum phase system, the preactuation time is given by the relative degree of the system. However, as can be seen in equation (4.14), the preactuation time for nonminimum phase systems is infinite. Fortunately this is, while theoretically true, not entirely the case. The contribution of future desired output values will decay as they are further away. Therefore, the preview time can be limited to T_{pre} by (Marconi et al. 2001):

$$u_u[k] = \sum_{l=k}^{k+T_{pre}} r[k]u_r[k-l] \tag{4.16}$$

In (Marconi et al. 2001) it is shown that the tracking error due to the limited preview time is bounded and decreasing while increasing the preview time. The minimum preview time depends on the required performance and on the distance of the nonminimum phase zeros from the unit circle. The further away the zeros from the unit circle, the shorter the preview time can be while achieving some desired performance.
Another issue is the presence of zeros on the unit circle. In (Marro and Hamano 1996, Marro 1996, Marconi et al. 2001) it is assumed that the system does not have any zeros on the unit circle. However, even in the case of zeros on the unit circle it is still possible to use the above method. Again the system $G_p(z) = N(z)/D(z)$ is considered where it is now assumed that $G_p(z)$ contains zeros on the unit circle. By partial fraction decomposition, the required input signal can be written as:

$$U(z) = \underbrace{\frac{A(z)}{N_s(z)}R(z)}_{=U_s(z)} + \underbrace{\frac{B(z)}{N_u(z)}R(z)}_{=U_u(z)} + \underbrace{\frac{C(z)}{N_m(z)}R(z)}_{=U_m(z)} \tag{4.17}$$

where $N_u(z)$ contains all strictly unstable zeros of $G_p(z)$, $N_s(z)$ contains all strictly stable zeros of $G_p(z)$, and $N_m(z)$ contains all zeros of $G_p(z)$ located on the unit circle. The control signal is constructed as:

$$u[k] = u_u[k] + u_s[k] + u_m[k] \tag{4.18}$$

The anti-causal part $u_u[k]$ can be obtained in exactly the same way as described earlier by equation (4.14), and also $u_s[k]$ can easily be obtained from $U_s[k] = \frac{A(z)}{N_s(z)}R(z)$. In principle there is no problem in finding $u_m[k]$ either, taking the inverse $\mathcal{Z}$-transform of the following equation immediately results in the required input signal:

$$U_m(z) = \frac{C(z)}{N_m(z)}R(z) \tag{4.19}$$

The problem however, is that an arbitrary desired output trajectory $r[k]$ which is bounded and has finite support does not automatically lead to an input signal with the same properties. For example the case that the system has exactly one zero at $z = 1$, equation (4.19) becomes:

$$U_m(z) = \frac{1}{1-z}C(z)R(z) \tag{4.20}$$

The above formula clearly demonstrates that in this special case the marginally stable control part integrates the product $C(z)R(z)$. As a direct result, the bounded and finite support desired output signal $R(z)$ only leads to an input signal with the same properties if the signal $C(z)R(z)$ has zero mean value. In general it follows from equation (4.19) that a bounded and finite support desired output signal only leads to an input signal with the same properties if the signal $R(z)$ stabilizes the marginally stable system:

$$G_m(z) = \frac{C(z)}{N_m(z)} \tag{4.21}$$

4.2 Stable Dynamic Inversion

The Stable Dynamic Inversion (SDI) method has been developed to reduce the sensitivity to measurement noise in the feedforward path (George et al. 1999*a*). The sensitivity to measurement noise only appears in tracking problems where the target signal has been measured and needs to be replicated. An example of these problems can be found in for example the automobile industry where measured road profiles have to be replicated on a test-bench. In these cases, knowledge on the measurement noise can be incorporated in the design procedure and the measurement noise will then be rejected to a large extend. As a direct consequence, the SDI procedure does not aim at, nor does result in, perfect tracking. In fact, the SDI procedure does not invert the system. Instead it assumes a model for the input and augments the system model with this model. For this augmented model a Kalman filter is generated which is capable of incorporating any available noise knowledge. The following system is considered:

$$x[k+1] = Ax[k] + Bu[k] + Gw[k] \tag{4.22}$$

$$y[k] = Cx[k] + Du[k] + v[k] \tag{4.23}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, $p = m$, and the matrices (A, B, C, D, G) are of appropriate size. The variables w and v represent the white noise which are

assumed to be uncorrelated with each other. Furthermore, it is assumed that the system (A, B, C, D) is both controllable and observable and the system is stable. The key-assumption is that the input signal can be modeled as:

$$u[k+1] = u[k] + \zeta[k] \tag{4.24}$$

where $\zeta[k]$ is white noise with covariance Q_ζ, and uncorrelated with $w[k]$ and $v[k]$. Augmenting the model (4.22) (4.23) with the input model (4.24) leads to the augmented system:

$$\begin{bmatrix} x[k+1] \\ u[k+1] \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} + \begin{bmatrix} G & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} w[k] \\ v[k] \\ \zeta[k] \end{bmatrix} \tag{4.25}$$

$$y[k] = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} + \begin{bmatrix} 0 & I & 0 \end{bmatrix} \begin{bmatrix} w[k] \\ v[k] \\ \zeta[k] \end{bmatrix} \tag{4.26}$$

By defining the augmented state $x_a[k] = [x[k] \;\; u[k]]^T$ and the augmented noise vector $n_a[k] = [w[k] \;\; v[k] \;\; \zeta[k]]$ the above system reduces to:

$$x_a[k+1] = A_a x_a[k] + G_a n_a[k] \tag{4.27}$$

$$y[k] = C_a x_a[k] + H_a n_a[k] \tag{4.28}$$

Lemma 4.2.1 *If the pair (C, A) is observable then the pair (C_a, A_a) is observable as well if the system (A, B, C, D) has no zeros at $z = 1$ (George et al. 1999a).*

The fact that the augmented system is no longer observable if the system has a zero at $z = 1$ (see Lemma 4.2.1) is caused by the cancellation of the pole at $z = 1$ of the input model. This problem can be circumvented by factoring out the zero at $z = 1$. First the transfer function of the plant is defined as:

$$G_p(z) = C(zI - A)^{-1}B + D \tag{4.29}$$

Then the zeros at $z = 1$ can be factored out by defining:

$$G_p(z) = P(z)G'_p(z) \tag{4.30}$$

where $P(z)$ contains the zero at $z = 1$. The problem can now be solved by defining a new desired output signal as $Y'_d(z) = P^{-1}(z)Y_d(z)$ and applying the SDI procedure for this new desired output to the system $G'_p(z)$.

In principle, the SDI procedure could be implemented such that it only requires a finite preview time of the desired output trajectory. However, in the software implementation of SDI a Kalman smoother is used which requires a large preview of the desired output. Therefore, altering the SDI procedure to the finite preview case is, though not impossible, not straightforward.

A method related to the SDI procedure is the optimal filtering method proposed by (Hou and Patton 1998). The underlying idea of this method is to reconstruct

the state of the system without knowing the inputs. However, unlike the optimal filtering procedure, SDI is applicable to nonminimum phase systems. Furthermore, the noise handling capabilities of the optimal filtering procedure are more restricted then for the SDI procedure.

4.3 Method of Dichotomies

The Method of Dichotomies (MD) was introduced in (Devasia et al. 1996) for nonlinear continuous-time systems, and extended to the class of linear discrete-time systems in for example (George et al. 1999*a*). In both (Devasia et al. 1996) and (George et al. 1999*a*), marginally stable zero dynamics (i.e. zeros on the unit circle for linear discrete time systems) were not permitted. In Section 4.3.1 the MD is introduced for linear discrete-time systems where it also shown that with an additional constraint on the desired output trajectory the method is applicable, without any modification, to systems with zeros on the unit circle. Section 4.3.2 considers the case where only a finite preview time of the desired output trajectory is available. Finally, Section 4.3.3 proposes an extended feedforward controller structure which improves the performance for extremely short preview times considerably.

4.3.1 Standard Method of Dichotomies

Consider system (4.1) where marginally stable zero dynamics are permitted. Assuming that the square system (4.1) has a well-defined vector relative degree $r = [r_1, r_2, \ldots, r_p]$, then by shifting each entry $y_i[k]$ $(i = 1 \ldots m)$ of the output vector $y[k]$ forward in time by its relative degree r_i (for simplicity written as $y[k+r]$ with $r \in \mathbb{R}^m$ the vector with the relative degree of each output as components) the following equation can be obtained obtained (Devasia 1997):

$$y[k+r] = \tilde{C}x[k] + \tilde{D}u[k] \tag{4.31}$$

The matrices $\tilde{C}$ and $\tilde{D}$ are defined by:

$$\tilde{C} = \begin{bmatrix} c_1 A^{r_1} \\ c_2 A^{r_2} \\ \vdots \\ c_p A^{r_m} \end{bmatrix} \qquad \tilde{D} = \begin{bmatrix} c_1 A^{r_1 - 1} B \\ c_2 A^{r_2 - 1} B \\ \vdots \\ c_p A^{r_m - 1} B \end{bmatrix} \tag{4.32}$$

where c_i represents the i^{th} row of C. Obviously the following input signal $u_d[k]$ leads to perfect tracking of the desired output $y_d[k]$ with a delay r:

$$u_d[k] = \tilde{D}^{-1}\left(y_d[k+r] - \tilde{C}x[k]\right) \tag{4.33}$$

With the above control law the closed-loop dynamics can be written as:

$$x[k+1] = \left(A - B\tilde{D}^{-1}\tilde{C}\right)x[k] + B\tilde{D}^{-1}y_d[k+r] \tag{4.34}$$

The poles of the closed-loop system correspond to the eigenvalues of the matrix $\tilde{A} = A - B\tilde{D}^{-1}\tilde{C}$, which also contains the zeros of the system. Hence, for non-minimum phase systems, the above direct inversion technique does not lead to a stable closed-loop system.
By the nonsingular transformation T_b:

$$\begin{bmatrix} \zeta[k] \\ \eta[k] \end{bmatrix} = T_b x[k] \tag{4.35}$$

the system (4.34) is brought into the Brunovsky canonical form, given by (Isidori 1995):

$$\begin{bmatrix} \zeta[k+1] \\ \eta[k+1] \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & 0 \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \zeta[k] \\ \eta[k] \end{bmatrix} + \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix} y_d[k+r] \tag{4.36}$$

$$y[k] = \begin{bmatrix} \tilde{C}_1 & 0 \end{bmatrix} \begin{bmatrix} \zeta[k] \\ \eta[k] \end{bmatrix} \tag{4.37}$$

with:

$$\zeta[k] = \begin{bmatrix} y_1[k] & y_1[k+1] & \dots & y_1[k+r_1-1] & \dots & y_m[k] & \dots & y_m[k+r_m-1] \end{bmatrix}^T \tag{4.38}$$

The poles of the closed-loop system (4.36) are given by the eigenvalues of the matrices $\tilde{A}_{11}$ and $\tilde{A}_{22}$. The matrix $\tilde{A}_{11}$ has $r_\Sigma = \sum_{i=1}^{m} r_i$ eigenvalues located at the origin. The remaining $n - r_\Sigma$ eigenvalues, corresponding to those of $\tilde{A}_{22}$, are located at the zeros of the output $y[k]$.
The variable $y_d[k]$ in equation (4.36) is completely specified and also $\zeta[k]$ can be substituted by $\zeta_d[k]$ ($\zeta_d[k]$ can be found by replacing all entries $y_i[k]$ in equation (4.38) by $y_{i,d}[k]$) and hence is completely specified. However, $\eta[k]$, representing the zero dynamics, still has to be found. The description for the zero dynamics can be extracted from equation (4.36):

$$\eta[k+1] = \tilde{A}_{22}\eta[k] + \tilde{B}_2 y_d[k+r] + \tilde{A}_{21}\zeta_d[k] \tag{4.39}$$

The transformation matrix T_η is defined such that:

$$\begin{bmatrix} \eta_s[k] \\ \eta_m[k] \\ \eta_u[k] \end{bmatrix} = T_\eta \eta[k] \qquad \begin{bmatrix} A_s & 0 & 0 \\ 0 & A_m & 0 \\ 0 & 0 & A_u \end{bmatrix} = T_\eta \tilde{A}_{22} T_\eta^{-1}, \tag{4.40}$$

where A_s contains all strictly stable eigenvalues of A_{22}, A_m contains all eigenvalues of A_{22} on the unit circle, and A_u contains all strictly unstable eigenvalues of A_{22}. Applying the transformation T_η to the zero dynamics given in equation (4.36) results in:

$$\eta_s[k+1] = A_s\eta_s[k] + \tilde{B}_{2,s} y_d[k+r] + \tilde{A}_{21,s}\zeta_d[k] \tag{4.41}$$

$$\eta_m[k+1] = A_m\eta_m[k] + \tilde{B}_{2,m} y_d[k+r] + \tilde{A}_{21,m}\zeta_d[k] \tag{4.42}$$

$$\eta_u[k+1] = A_u\eta_u[k] + \tilde{B}_{2,u} y_d[k+r] + \tilde{A}_{21,u}\zeta_d[k] \tag{4.43}$$

The solutions for equations (4.41) and (4.42) can be found by simple summation starting from some known initial $\eta_s[k_i]$ and $\eta_m[k_i]$. Since the eigenvalues of the matrix A_u are unstable this cannot be employed to obtain the solution for equation (4.43). However, if the final value $\eta_u[\infty]$ is known, the solution for equation (4.43) can be found by solving it backward in time. Summarizing, equations (4.41), (4.42) and (4.43) can be solved under the following conditions:

C_I The initial and final states $x[-\infty]$ and $x[\infty]$ (and consequently the zero dynamics) are known to be zero.

C_{II} The desired output trajectory $y_d[k]$ is bounded and has finite support.

C_{III} In the special case of any marginally stable zeros, the desired output trajectory $y_d[k]$ is such that the zero dynamics, starting from zero at $k = -\infty$, are driven back to zero at $k = \infty$.

Conditions C_{II} and C_{III} are additional conditions which guarantee that the boundary conditions given in condition C_I exist for the given desired output trajectory. Conversely, if condition C_I is fulfilled, then conditions C_{II} and $_{III}$ are automatically fulfilled. However, conditions C_{II} and $_{III}$ can be checked a priori in contrast of condition C_I. Keeping all conditions in mind, the solutions for $\eta_s[k]$, $\eta_m[k]$, and $\eta_u[k]$ are given by:

$$\eta_s[k] = \sum_{i=-\infty}^{k-1} A_s^{(k-i)} \left\{ \tilde{B}_{2,s} y_d[i+r] + \tilde{A}_{21,s} \zeta_d[i] \right\} \tag{4.44}$$

$$\eta_m[k] = \sum_{i=-\infty}^{k-1} A_m^{(k-i)} \left\{ \tilde{B}_{2,m} y_d[i+r] + \tilde{A}_{21,m} \zeta_d[i] \right\} \tag{4.45}$$

$$\eta_u[k] = -\sum_{i=k}^{\infty} A_u^{-(i+1-k)} \left\{ \tilde{B}_{2,u} y_d[i+r] + \tilde{A}_{21,u} \zeta_d[i] \right\} \tag{4.46}$$

Unfortunately, in order to achieve perfect tracking while preserving condition C_I, condition C_{III} is unavoidable. It can be seen from equation (4.45) that the marginally stable zero dynamics solely are driven by the desired output trajectory. Therefore, as demonstrated in the next example for one marginally stable zero at $z = 1$, the desired output trajectory should fulfill condition C_{III} in order to fulfill condition C_I.

Example 4.3.1 *Suppose that the system under consideration has one marginally stable zero at $z = 1$. Furthermore let us consider that the system started at the initial state $x[-\infty] = 0$, therefore $\eta_m[-\infty] = 0$. From equation (4.45) the final value $\eta_m[\infty]$ can be found to be:*

$$\eta_m[\infty] = \sum_{i=-\infty}^{\infty} \left\{ \tilde{B}_{2,m} y_d[i+r] + \tilde{A}_{21,m} \zeta_d[i] \right\} \tag{4.47}$$

which can be written with equation (4.38) as:

$$\eta_m[\infty] = \tilde{B}_2 \begin{bmatrix} \sum_{i=-\infty}^{\infty} y_{d,1}[i+r_1] \\ \sum_{i=-\infty}^{\infty} y_{d,2}[i+r_2] \\ \cdots \\ \sum_{i=-\infty}^{\infty} y_{d,m}[i+r_m] \end{bmatrix} + \tilde{A}_{21,m} \begin{bmatrix} \sum_{i=-\infty}^{\infty} y_{d,1}[i] \\ \sum_{i=-\infty}^{\infty} y_{d,1}[i+1] \\ \cdots \\ \sum_{i=-\infty}^{\infty} y_{d,1}[i+r_1-1] \\ \cdots \\ \cdots \\ \sum_{i=-\infty}^{\infty} y_{d,m}[i] \\ \cdots \\ \sum_{i=-\infty}^{\infty} z_m[i+r_m-1] \end{bmatrix} \tag{4.48}$$

(the notation $y_{d,j}$ is used to represent the j^{th} component of the desired output vector) Equation (4.48) clearly shows that $\eta_m[\infty]$ is formed by a linear combination of the average value of the desired output signal. To guarantee that $\eta_m[\infty]$ returns to zero it is sufficient that $\sum_{i=-\infty}^{\infty} y_{d,j}[i] = 0 \quad \forall \quad j = 1 \ldots m$. In other words, the desired output trajectory should be such that it stabilizes the marginally stable zero dynamics.

If the desired output trajectory does not satisfy condition C_{III} the generated input signal will still be bounded and result in perfect tracking. However, the generated input signal will no longer have finite support. Furthermore, in the case that the system has marginally stable zeros condition C_I can no longer be satisfied if condition C_{III} is not satisfied. This can easily be seen by constructing the transformation T_b' from transformations (4.35) and (4.40) as:

$$T_b' = \begin{bmatrix} I & 0 \\ 0 & T_\eta \end{bmatrix} T_b \tag{4.49}$$

The transformation matrix T_b' is invertible by the invertibility of T_η and T_b. Therefore, the desired state can be reconstructed from:

$$x_d[k] = (T_b')^{-1} \begin{bmatrix} \zeta_d[k] \\ \eta_s[k] \\ \eta_m[k] \\ \eta_u[k] \end{bmatrix} \tag{4.50}$$

where $(T_b')^{-1}$ denotes the inverse of T_b'. If we now assume that the system has marginally stable zeros, that conditions C_I and C_{II} are fulfilled, and that condition C_{III} is not fulfilled, then we find for $\lim_{k\to\infty} x_d[k]$:

$$\lim_{k\to\infty} x_d[k] = (T_b')^{-1} \begin{bmatrix} \lim_{k\to\infty} \zeta_d[k] \\ \lim_{k\to\infty} \eta_s[k] \\ \lim_{k\to\infty} \eta_m[k] \\ \lim_{k\to\infty} \eta_u[k] \end{bmatrix} \tag{4.51}$$

From condition C_{II} it follows that $\lim_{k\to\infty} \zeta_d[k] = 0$. Furthermore, from equations (4.44) and (4.46) it can be concluded that both $\lim_{k\to\infty} \eta_s[k] = 0$ and $\lim_{k\to\infty} \eta_u[k] =$

0. However, by assumption $\lim_{k\to\infty} \eta_m[k] \neq 0$ leading to:

$$\lim_{k\to\infty} x_d[k] =\neq 0 \tag{4.52}$$

While assuming that conditions C_I and C_{II} are fulfilled and condition C_{III} is not fulfilled, it is not possible to generate a desired state trajectory $x_d[k]$ fulfilling C_I. Hence, we found a contradiction. Unfortunately, in practice it is not uncommon to have a conflict with condition C_{III}. Examples can be found in for example the control of a helicopter (Devasia 1997). Possible solutions to this problem are:

1. Redefining the desired output trajectory in such a way that condition C_{III} is fulfilled.

2. Change the model of the system, thereby removing the zeros from the unit circle (Devasia 1997).

While both solutions are conceptually quite different, their result is similar: instead of achieving perfect tracking, approximate tracking is achieved.
Let us now return to the main topic: the generation of a feedforward control signal. Substituting the desired state given in equation (4.50) into equation (4.33) results in the desired input signal. For the model, the desired input signal leads to perfect tracking, under the assumption of conditions C_I, C_{II} and, in the case of marginally stable zero dynamics, C_{III}.

4.3.2 Method of Dichotomies with Bounded Preview Time

From the equations (4.44), (4.45), and (4.46), it can be seen that the developed feedforward controller is acausal. Computing the stable and marginally stable part requires the desired output $\max\{r_i\}_{i=1\ldots m}$ time steps in advance. However, the computation of the unstable part depends on the total future of the desired output trajectory.
In the case that the number of time steps that the desired output signal is known in advance is finite, while still larger then the largest entry of the relative degree vector, it is still possible to obtain an estimate of the desired state, leading to the Method of Dichotomies with Bounded Preview Time (MDBP). It can be seen that equations (4.44) and (4.45) are not affected by this in the sense that all required information to compute $\eta_s[k]$ and $\eta_m[k]$ are available at time k, however we cannot guarantee whether condition C_{III} will be satisfied in advance. Due to the limited preview time, not all required information for the unstable zero dynamics represented by equation (4.46) are available. Setting all unknown terms of $y_d[k+i] \;\forall\; i > T_{pre}$ to zero the following estimate of $\eta_u[k]$ can be found:

$$\tilde{\eta}_u[k] = -\sum_{i=k}^{T_{pre}} A_u^{-(i+1-k)} \left\{ \tilde{B}_{2,u} y_d[i+r] + \tilde{A}_{21,u} \zeta_d[i] \right\} \tag{4.53}$$

The total preview time is then given by $T_{tot} = T_{pre} + \max\{r_i\}_{i=1\ldots m}$. The error of

the above estimation can be found to be:

$$e_{pre}[k] = -\sum_{i=T_{pre}+1}^{\infty} A_u^{-(i+1-k)} \left\{ \tilde{B}_{2,u} y_d[i+r] + \tilde{A}_{21,u} \zeta_d[i] \right\} \tag{4.54}$$

By boundedness of the desired output signal $y_d[k]$ and the stable eigenvalues of the matrix A_u^{-1}, the above error signal is bounded as well. Furthermore, it is easy to see that the estimation error decreases while increasing the preview time T_{pre}. In the limit case $T_{pre} \to \infty$, the error e_{pre} becomes zero. The following can be found for the error state $e_x[k] = x_d[k] - \tilde{x}_d[k]$ and the tracking error $e_y[k] = y_d[k] - \tilde{y}_d[k]$:

$$\begin{aligned} e_x[k+1] &= Ae_x[k] + B\tilde{D}^{-1}\tilde{C}\left(T_b'\right)^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ e_{pre}[k] \\ 0 \\ 0 \\ 0 \\ e_{pre}[k] \end{bmatrix} \\ e_y[k] &= Ce_x[k] + D\tilde{D}^{-1}\tilde{C}\left(T_b'\right)^{-1} \end{aligned} \tag{4.55}$$

The above error system is driven by the error signal $e_{pre}[k]$. It was already concluded that $e_{pre}[k]$ is bounded and it was assumed that the system (A, B, C, D) is stable. Therefore, the tracking error $e_y[k]$ is bounded as well.

Remark 4.3.2 *The desired output trajectory was estimated to be zero after the preview horizon represented by T_{tot}. Of course, depending on the application, more sophisticated approximation schemes could be employed such as polynomial extrapolation.*

The estimated desired state $\tilde{x}_d[k]$ generated by BPMD is given by:

$$\tilde{x}_d[k] = \left(T_b'\right)^{-1} \begin{bmatrix} \zeta_d[k] \\ \eta_s[k] \\ \eta_m[k] \\ \tilde{\eta}_u[k] \end{bmatrix} \tag{4.56}$$

Interesting to mention is that the desired state (4.56) exhibits the property:

$$\tilde{y}_d[k] = C\tilde{x}_d[k] = y_d[k] \tag{4.57}$$

The tracking error, given by equation (4.55), is caused by the error in the computed input signal. Therefore, the next section explores a method which uses a tracking controller within the feedforward controller to track the estimated desired state trajectory better.

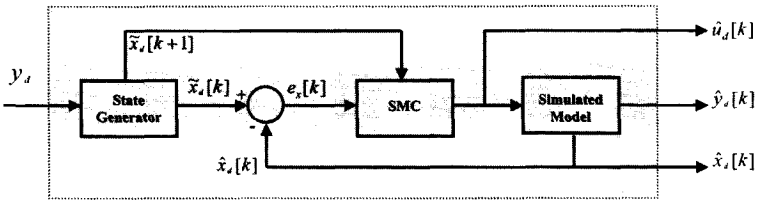


Fig. 4.1: BPMDSM schematic. The shaded box represents the boundaries of the feedforward controller.

4.3.3 Extended Method of Dichotomies with Bounded Preview Time

As can be seen in the simulation example of Section 4.4, the BPMD works very well for a relatively large preview time. For extremely short preview times the performance decreases considerably. Therefore, as motivated in the previous section, a feedback-loop is included within the feedforward controller. Although this looks as if it contradicts the feedforward design, as figure Figure 4.1 shows, this is exactly what happens in the proposed setup. The finite preview time state generator generates an estimated desired state according to the procedure of Section 4.3.2. This estimated desired state is denoted by $\tilde{x}_d[k]$. The simulated model block in Figure 4.1 is described by the equation:

$$\hat{x}_d[k+1] = A\hat{x}_d[k] + B\hat{u}_d[k] \tag{4.58}$$

The generated desired state $\tilde{x}_d[k]$ is now used as the target state for the simulated model. To accomplish this, a SDSMC is constructed similar to the SDSMC in Section 3.1 leading to the Bounded Preview Method of Dichotmies with Sliding Mode control (BPMDSM). The difference is the definition of the switching function. Instead of defining the switching function on the system state, the SDSMC is now defined on the following error state:

$$e_{\hat{x}}[k] = \tilde{x}_d[k] - \hat{x}_d[k] \tag{4.59}$$

resulting in:

$$\sigma_{e_{\hat{x}}}[k] = Se_{\hat{x}}[k] = S\left(\tilde{x}_d[k] - \hat{x}_d[k]\right) \tag{4.60}$$

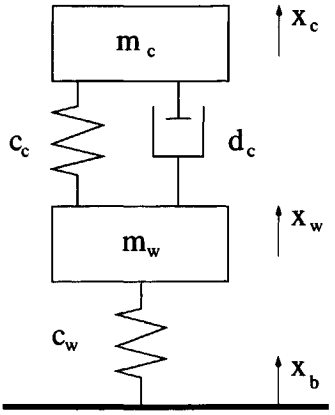
For the determination of the matrix S exactly the same procedure can be used as in Section 3.2.

Then, together with the reaching law $\sigma_e[k+1] = 0$ (which is the linear reaching law introduced in Section 3.1.4 with $\Phi = 0$) and equations (4.58),(4.60) the following control law is obtained:

$$\hat{u}_d[k] = (SB)^{-1}\left(S\tilde{x}_d[k+1] - SA\hat{x}_d[k]\right) \tag{4.61}$$

Despite the fact that we have perfect knowledge of the error state $e_x[k]$ (it is computed from $\tilde{x}_d[k]$ and $\hat{x}_d[k]$ which are both available), it is not possible to achieve perfect tracking. The reason for this is that the generated state $\tilde{x}_d[k]$ is not trackable which originates in the fact that $\tilde{x}_d[k]$ is an estimate of the desired state $x_d[k]$,

(a)



(b)

Fig. 4.2: Figure (a) displays a test setup. Figure (b) represents a mechanical diagram of the Quarter Car Model.

as described in Section 4.3.2. Therefore, the task of the SDSMC is to track the generated desired state $\tilde{x}_d[k]$ as good as possible. The simulation example in Section 4.4 clearly demonstrates the improvement made by the proposed setup.

4.4 Simulation Example

As a simulation example we have chosen the so called Quarter Car. It represents one quarter of a vehicle placed on a moving base. To improve reproducibility of test procedures for cars as well as durability tests of new developed cars, one would like to reproduce predefined road-profiles exactly. Therefore the goal of the controller is to reproduce a measured road profile and hence give the car on the base exactly the same accelerations in every successive test. A picture of a possible test setup together with a mechanical diagram is presented in Figure 4.2 for which we can obtain the linear model:

$$x[k+1] = Ax[k] + Bu[k] \tag{4.62}$$

$$y[k] = Cx[k] \tag{4.63}$$

The system state, input and output are given by:

$$x = [x_c \quad x_w \quad \dot{x}_c \quad \dot{x}_w]^T \qquad y = x_c \qquad u = x_b$$

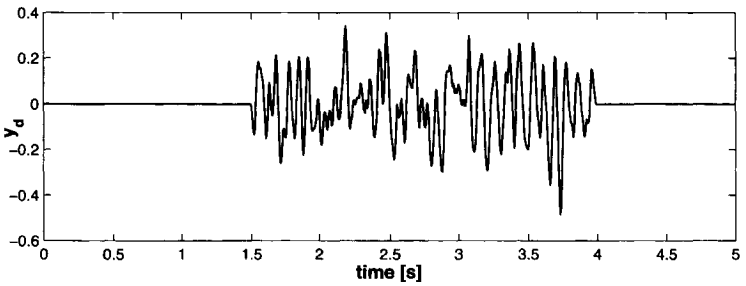


Fig. 4.3: Desired output trajectory y_d.

The variables x_c, x_w, and x_b represent the car, wheel and base displacement respectively. The system matrices are given by:

$$
\begin{aligned}
A_c &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{c_c}{m_c} & \frac{c_c}{m_c} & -\frac{d_c}{m_c} & \frac{d_c}{m_c} \\ \frac{c_c}{m_w} & -\frac{c_c+c_w}{m_w} & \frac{d_c}{m_w} & -\frac{d_c}{m_w} \end{bmatrix} \\
B_c &= \begin{bmatrix} 0 & 0 & 0 & \frac{c_w}{m_w} \end{bmatrix}^T \\
C_c &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

where $m_w = 33\ [kg]$ is the mass of the wheel, $m_c = 200\ [kg]$ is one quarter of the mass of the car, $c_w = 20000\ [N/m]$ is the wheel stiffness, $c_c = 9000\ [N/m]$ the suspension stiffness and $d_c = 1200\ [Nsec/m]$ the suspension damping. The above matrices (A_c, B_c, C_c) are discretised by sample and hold to obtain the matrices (A, B, C). The discretised model has relative degree $r = 1$, zeros $\{-3.54, 0.969, -0.260\}$ and poles $\{0.8323 \pm 0.373i, 0.991 \pm 0.022i\}$, consequently the system is nonminimum phase. The sampling time is taken to be $T_s = 5ms$. Figure 4.3 presents the target signal. The generation of the feedforward signal will be successively solved for the infinite preview time case and the finite preview time case.

- *Infinite Preview Time* In the case that the desired output trajectory is fully known in advance, the input signal can be computed by either SDI or MD. The MD results in perfect tracking ($VAF(y_d, y_{MD}) = 100\ \%$) while ensuring stability of the feedforward control signal, as can be seen in Figure 4.4. Using the SDI procedure, a trade off can be made between the tracking performance and the aggressiveness in the input signal. In some cases perfect tracking is pursued, in other cases the aggressiveness of the input signal should be considered as well. While this trade-off can be easily made in optimal control, SDI offers the possibility as well. Reducing the covariance matrix Q_ζ automatically slows the generated input signal down, of course at a cost of tracking performance. The difference is shown in Figure 4.5 for $Q_\zeta = 1e^1$ and $Q_\zeta = 1e^{10}$. For $Q_\zeta = 1e^{10}$ perfect tracking is achieved ($VAF(y_d, y_{SDI}) = 100\ \%$). Reducing Q_ζ to $Q_\zeta = 1e^1$ reduces the tracking

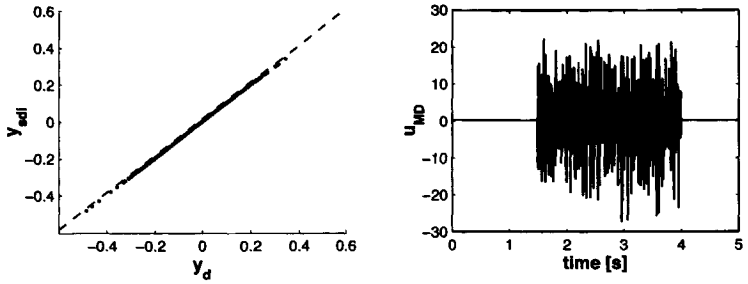


Fig. 4.4: Tracking performance for the MD procedure. The left figure represents the xy-plot of achieved tracking, where the dashed diagonal represents perfect tracking. The right figure represents the input signal.

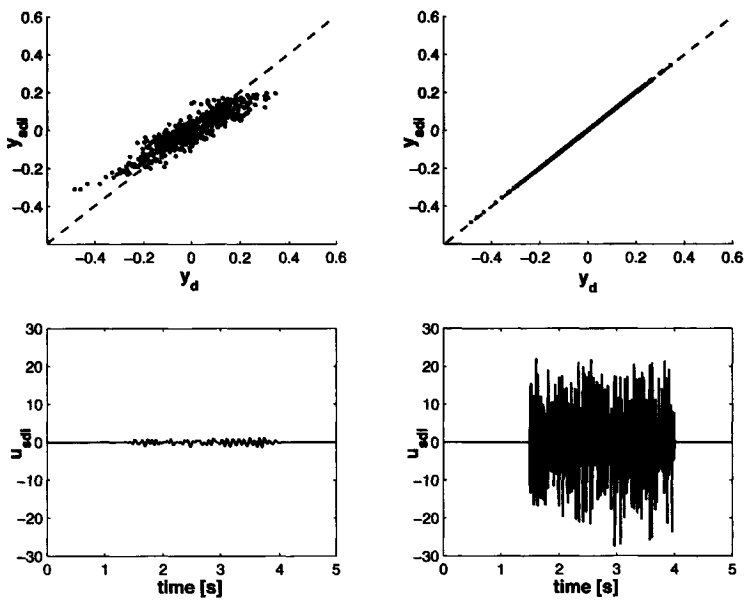


Fig. 4.5: Tracking performance for the SDI procedure with $Q_\zeta = 10$ (left figures) and $Q_\zeta = 1e^{10}$ (right figures). The top figures represent the xy-plots of achieved tracking, where the dashed diagonal represents perfect tracking. The bottom figures represent the input signals.

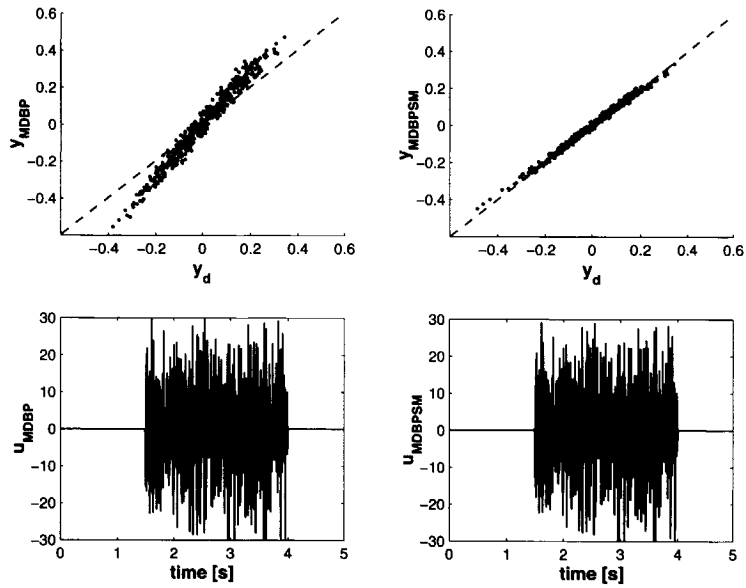


Fig. 4.6: Tracking performance of the BPMD (left figures) and the BPMDSM (right figures) with a preview time of $T_{tot} = 4T_s$. The top figures represent the xy-plot of achieved tracking, where the dashed diagonal represents perfect tracking. The bottom figures represent the input signal.

performance to a $VAF(y_d, y_{SDI}) = 79.8\,\%$. However, as can be seen in Figure 4.5, the amplitude of the control signal is reduced considerable as well.

- *Finite Preview Time* For the finite preview time problem we will consider the BPMD and the BPMDSM methods. The parameters for the BPMDSM are given by $\Phi = 0$ and $S = [0.93 \quad -2.33 \quad -0.48 \quad -0.99]$. The simulation results are shown in Figure 4.6 for a preview time $T_{tot} = 4T_s$. Computing the VAF for the two cases results in:

$$\begin{aligned} VAF(y_d, y_{BPMD}) &= 74.6\% \\ VAF(y_d, y_{BPMDSM}) &= 99.1\% \end{aligned} \tag{4.64}$$

Varying the preview time and recording the obtained tracking performances we obtain table 4.1. Clearly a longer preview time leads to better tracking. Furthermore, BPMDSM gives a better performance with a shorter preview time compared to BPMD.

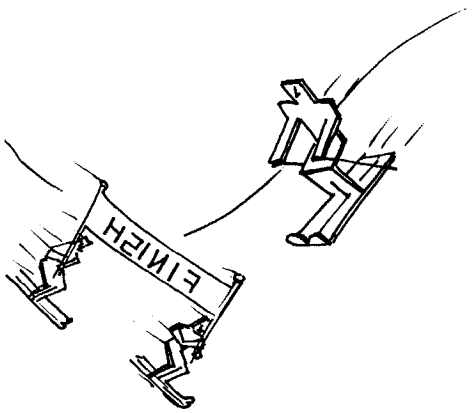
4.5 Summary

This chapter considers the problem of feedforward control for an arbitrary desired output trajectory. Whereas a feedforward controller for minimum phase systems can be obtained by simply inverting the model, this leads to unbounded signals for

T_{tot}	MDBP	MDBPSM
3	-	88.8 %
4	74.6 %	99.1 %
5	98.0 %	99.9 %
6	99.8 %	100.0 %
7	100.0 %	100.0 %

Table 4.1: Tracking performance (represented as VAF) for BPMD and BPMDSM for different preview times.

nonminimum phase systems. In this chapter three methods are described in detail which generate input signals leading to approximate or even perfect tracking while ensuring boundedness of the input signal. First Section 4.1 describes the Steering Along the Zeros Control (SAZC) procedure. Despite the fact that this procedure was originally introduced for systems without zeros on the unit circle it is shown that, under one extra constraint on the desired output trajectory, the method is applicable to systems with zeros on the unit circle as well. Then Section 4.2 introduces the Stable Dynamic Inversion (SDI) technique. Despite the name, this method is in fact not an inversion technique. Instead, the input signal is modeled as an integrator driven with white noise. Then by augmenting the system description with the input model a Kalman filter is constructed to reconstruct the augmented state and hence the input signal. Finally, Section 4.3 introduces the Method of Dichotomies (MD). While this method was introduced for system descriptions without zeros on the unit circle it is shown that under an extra condition the method is applicable to systems with zeros on the unit circle as well. Furthermore, the method is adapted to the finite preview case which results in approximate tracking. To increase the performance for extremely short preview times, Section 4.3.3 proposes a new feedforward controller structure. A DSMC is used within the feedforward controller to minimize the error due to the bounded preview time. The SDI procedure and the MD are compared in Section 4.4 on a simple simulation example.
The results of Section 4.3 and 4.4 have been published in (Monsees and Scherpen 2002*b*).



5

Tracking

In this chapter we consider the control problem where some desired, dynamical, state- or output trajectory needs to be replicated by the system under control. Tracking problems are often encountered in practice. Consider for example the cases where a robot needs to follow a given trajectory, a machine tool which needs to perform certain prescribed actions, or an airplane which needs to follow a certain flight plan. In all these applications, a controller is required which gives the best possible tracking of the given reference trajectory despite any modeling errors or external disturbances. In this chapter we consider the two typical cases of tracking control:

- Target tracking control. For target tracking we assume that some desired output- or state trajectory is defined which needs to be replicated exactly by the system. Consequently, the goal for the closed-loop system is perfect tracking of the desired trajectory.
- Model reference control. Using a model reference controller we do not aim at perfect tracking of some desired trajectory, but we aim at tracking some ideal model.

While in target tracking full attention is given to the tracking performance of the closed-loop system, in a model reference approach attention is given to the dynamics of the closed-loop system. The difference can be explained by considering some specified, step-wise constant, output trajectory (for example a block-wave). Examples of such applications can be found in for example the control of an airplane which is required to track some new course after each way-point. Perfect tracking of such a discontinuous signal is not possible for most real-life systems since it would involve input signals which are far out of the permissible limits. Therefore, it is better to define some desired dynamics by which the system makes the transition

from one output value to another. However, in other applications it is required that the closed-loop system tracks some prescribed state- or output trajectory exactly. Examples can be found in for example robot control where the robot needs to follow some trajectory in order to avoid collision with other objects. In these situations, one has to assume that the trajectory has been specified such that the system is capable of tracking it (i.e. all system variables, such as inputs and states, remain within their boundaries). In other words, the defined trajectory should be tractable. In this chapter we solve the problem of target tracking and of model reference control with a DSMC. For each problem a state-based, as well as an output-based approach is presented. Solutions for the target tracking problem are presented both in combination with and without a feedforward controller (as defined in Chapter 4). The system under control is given by:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] + f(x,k) \\ y[k] &= Cx[k] + Du[k] \end{aligned} \tag{5.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, and the quadruple (A, B, C, D) is both controllable and observable. The vector $f(x,k)$ represents the bounded disturbances. Furthermore, it is assumed that $p \geq m$. As stated before, for perfect tracking the desired output trajectory should be tractable. The following definition gives an interpretation of tractability which is used throughout this chapter. It is not concerned with any system limitations nor with disturbances, instead it is just concerned with boundedness of the resulting input and state signals.

Definition 5.0.1 *A desired output signal $y_d[k]$ is tractable by the system (5.2) if there exists a bounded input signal $u[k]$ and an initial state $x[0]$, satisfying $y[0] = Cx[0] + Du[0]$, which applied to the system (5.2) leads to $y[k] = Cx[k] + Du[k] = y_d[k] \ \forall \ k > 0$. This specific input signal is called the desired input signal denoted by $u_d[k]$. The corresponding state trajectory is called the desired state, denoted by $x_d[k]$.*

Similarly, tracktability of some desired state signal is defined in the following definition.

Definition 5.0.2 *A desired state signal $x_d[k]$ is said to be tractable by the system (5.2) if there exists a bounded input signal $u[k]$ which applied to the system (5.2) leads to $x[k] = Ax[k] + Bu[k] = x_d[k] \ \forall \ k > 0$. This specific input signal is called the desired input signal denoted by $u_d[k]$.*

Although DSMC has been introduced for stabilization in Chapter 3, it was already demonstrated in Examples 3.3.12 and 3.3.13 that DSMC can be applied to tracking problems as well. However, the DSMCs developed in Chapter 3 aim at stabilization only, i.e. the dynamics of the desired state- or output signal are not taken into account. In other words, the desired state- or output signal is assumed to be constant. In contrast to the controllers developed in Chapter 3, the controllers developed in this chapter specifically address the problem of tracking some time varying desired state- or output signal. In every controller structure introduced in

this chapter a preview of the desired state- or output signal can be identified. This preview has the form of a feedforward signal, a reference model, or explicitly a one step preview of the desired state- or output signal.
The main difference between the controllers developed in this chapter and the feedforward controllers developed in Chapter 4 is the presence of uncertainty. For a feedforward controller the model is controlled, it is assumed that the system under control behaves the same as the model. No measurements are required nor used in a feedforward controller. The success of a feedforward controller logically depends on the quality of the model, i.e. to which extend the model predicts the behaviour of the system. In the case of varying parameters, external disturbances, or neglected dynamics, the system behaviour will deviate from the model leading to a decrease in tracking performance. Therefore, this section considers controllers which use the available system information to improve the tracking performance. In some implementations the controller is split into a feedforward controller and a feedback controller. In other controller implementations this division is not as obvious but in principle still present.
As mentioned before, the controllers presented in this chapter are based on the DSMC theory presented in Chapter 3. As will become clear throughout this chapter, there is no substantial difference in the construction of the control laws. Therefore, the disturbance vector $f(x,k)$ is left out of the discussion in this chapter and the interested reader is referred to Chapter 3 for this topic. Therefore, the system representation (5.1) is altered to:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] + f(x,k) \\ y[k] &= Cx[k] + Du[k] \end{aligned} \tag{5.2}$$

5.1 Target Tracking Control

In the following sections we explore the field of tracking some desired output signal perfectly. First the case is considered where the desired output trajectory is translated to a desired state trajectory. A SDSMC is then defined to track this desired state signal. As a second approach an ODSMC is defined to track the desired output trajectory. For both the state-based case and the output-based case a feedforward/feedback configuration and a tracking controller is considered.

5.1.1 State-Based Target Tracking Control

In this section we assume that we have to track some tractable desired output signal. The first step is to generate the desired state signal $x_d[k]$ and the corresponding desired input signal $u_d[k]$ leading to perfect tracking for the ideal system represented by equation (5.2). The solution to this problem can be found in Chapter 4. In this chapter a feedforward controller is developed which results in perfect tracking of some tractable desired output signal. However, perfect tracking can only be accomplished if the desired output signal is known in advance, i.e. the feedforward controller needs a preview of the desired output trajectory . For minimum phase systems this preview time is finite. It is determined by the largest relative degree of the outputs. For nonminimum phase systems, the preview time is infinite. It was

shown in Chapter 4, that in the case of only a finite preview time we can still achieve almost perfect tracking. We assume here that the preview time is sufficiently large, such that we can assume that the generated desired input signal results in perfect tracking of the desired output signal. Therefore, we can write:

$$\begin{aligned} x_d[k+1] &= Ax_d[k] + Bu_d[k] \\ y_d[k] &= Cx_d[k] + Du_d[k] \end{aligned} \tag{5.3}$$

The next step is to design a SDSMC which forces the closed-loop system to track the desired state trajectory. Defining the state error as:

$$e_x[k] = x[k] - x_d[k] \tag{5.4}$$

and using equations (5.2) and (5.3) we can determine the state error dynamics to be:

$$e_x[k+1] = Ae_x[k] + B(u[k] - u_d[k]) \tag{5.5}$$

Equation (5.5) can be brought into regular form by the transformation $T_{r_x} \in \mathbb{R}^{n \times n}$:

$$\begin{bmatrix} \bar{e}_{x_1}[k] \\ \bar{e}_{x_1}[k] \end{bmatrix} = T_{r_x} e_x[k] \tag{5.6}$$

resulting in the following representation for the error dynamics:

$$\begin{bmatrix} \bar{e}_{x_1}[k+1] \\ \bar{e}_{x_1}[k+1] \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{e}_{x_1}[k] \\ \bar{e}_{x_1}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{B}_2 \end{bmatrix} (u[k] - u_d[k]) \tag{5.7}$$

The switching function is defined as:

$$\sigma_{e_x}[k] = Se_x[k] = \bar{S}_1 \bar{e}_{x_1}[k] + \bar{S}_2 \bar{e}_{x_2}[k] \tag{5.8}$$

Assuming that the closed-loop system is in perfect sliding mode, i.e. $\sigma_{e_x} = 0$, equation (5.8) can be rewritten as:

$$\bar{e}_{x_2}[k] = \bar{S}_2^{-1} \bar{S}_1 \bar{e}_{x_1}[k] \tag{5.9}$$

Inserting equation (5.9) into the dynamics for $\bar{e}_{x_1}$, which can be found from equation (5.7), leads to the reduced order dynamics in sliding mode:

$$\bar{e}_{x_2}[k+1] = \left(\bar{A}_{11} - \bar{A}_{12} \bar{S}_2^{-1} \bar{S}_1\right) \bar{e}_{x_1}[k] \tag{5.10}$$

The above dynamics should be stable, which can be accomplished by the correct choice for the design parameters $\bar{S}_1$ and $\bar{S}_2$. Interesting to note is that the above design procedure for the sliding surface is identical to the design procedure for the SDSMC presented in Section 3.1. The only difference is that in Section 3.1 the system state is stabilized at the origin in state space, whereas in this section we aim at stabilizing the error state $e_x[k]$. For the design of the sliding surface this makes no difference. Apparently, the dynamics of the error state are identical to the dynamics of the system state.

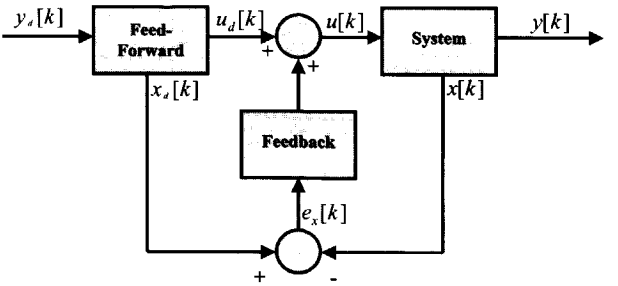


Fig. 5.1: Block diagram of the state-based feedforward/feedback configuration.

Assuming that, by for example LQR design or pole placement, the matrices $\bar{S}_1$ and $\bar{S}_2$ (the matrix S can be obtained from $S = [\bar{S}_1 \ \ \bar{S}_2]T_{r_x}$) have been chosen such that the reduced order dynamics in the sliding mode are stable, we can now design a control law which drives the closed-loop in the sliding mode. In order to do so we use the linear reaching law presented in Section 3.1.4, which is given by:

$$\sigma_{e_x}[k+1] = \Phi\sigma_{e_x}[k] \tag{5.11}$$

For simplicity we assume that the design parameter $\Phi \in \mathbb{R}^{m\times m}$ has been chosen diagonal with all its diagonal entries $\phi_i \quad i = 1 \ldots m$ satisfying $0 \leq \phi_i < 1$. Then, using equations (5.11) and (5.4) we obtain:

$$\Phi\sigma_{e_x}[k] = S\left(x[k+1] - x_d[k+1]\right) \tag{5.12}$$

From equation (5.12) we have two options to come to a controller:

- **Feedforward/Feedback Configuration:** Inserting equations (5.2) and (5.3) in equation (5.12) leads to:

$$\Phi\sigma_{e_x}[k] = S\left(Ax[k] + Bu[k] - Ax_d[k] - Bu_d[k]\right) \tag{5.13}$$

  Realizing that $Ae_x[k] = Ax[k] - Ax_d[k]$ we can rewrite equation (5.13) as:

$$u[k] = u_d[k] + (SB)^{-1}\left(\Phi\sigma_{e_x}[k] - SAe_x[k]\right) \tag{5.14}$$

  The control signal $u_d[k]$ can be seen as a feedforward signal since it is based on off-line computations only. The other terms at the right hand side of equation (5.14) represent the feedback part of the controller since it is based on the measured state error. In Figure 5.1 the block diagram of the state-based feedforward/feedback configuration is displayed.

- **Tracking Controller:** Inserting equation (5.2) in equation (5.12) leads to:

$$\Phi\sigma_{e_x}[k] = S\left(Ax[k] + Bu[k] - x_d[k+1]\right) \tag{5.15}$$

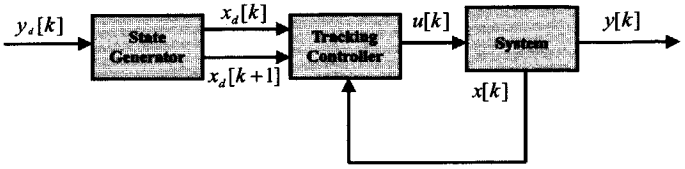


Fig. 5.2: Block diagram of the state-based tracking controller.

From equation (5.15) the control law can instantly be determined to be:

$$u[k] = (SB)^{-1} \left(\Phi\sigma_{e_x}[k] + Sx_d[k+1] - SAx[k]\right) \tag{5.16}$$

In Figure 5.2 the block diagram of the state-based tracking controller is displayed.

The first thing to notice if we compare the Feedforward/Feedback Configuration with the Tracking Controller (equations (5.13) and (5.16) respectively) is that their effect is exactly the same. Both controllers drive the closed-loop system towards the sliding surface according to the linear reaching law $\sigma_{e_x}[k+1] = \Phi\sigma_{e_x}[k]$. In fact, by the correct manipulation, thereby using equation (5.3), equations (5.13) and (5.16) can be rewritten into each others form. The difference is however, that the Feedforward/Feedback Configuration is a straightforward extension of a, possibly already existing, feedforward configuration. In other words, the feedback controller can be omitted in a first test run, thereby avoiding stability problems due to an extremely bad model. Another important reason for using the feedforward/feedback configuration could be that the performance of an existing system set-up should be increased. It could then be preferable that the original feedforward controller is maintained, and only a feedback loop is added. The role of the feedback could then be gradually increased when information is gained for the design of the feedback controller. This avoids the possible danger of instability or of the calculation of an aggressive control signal.

5.1.2 Output-Based Target Tracking Control

In this section we introduce a DSMC tracking controller which is based on the output signals, rather than the system states. For the moment however, we still assume the availability of the systems' state, i.e. only the definition of the switching function is changed. The situation where only the output measurements will be use is addressed at the end of this section. For the application of output-based DSMC it was shown in Chapter 3 that we need to impose the following restrictions to the system (5.2):

- The system is minimum phase
- The matrix product CB has full rank
- The matrix D is zero

In the rest of this section it is assumed that these restrictions apply to the system (5.2) which is therefore reduced to:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] \end{aligned} \tag{5.17}$$

As in the previous section it is assume that there exists some desired input signal $u_d[k]$ and some desired state signal $x_d[k]$ such that:

$$\begin{aligned} x_d[k+1] &= Ax_d[k] + Bu_d[k] \\ y_d[k] &= Cx_d[k] \end{aligned} \tag{5.18}$$

At this point we again refer to Chapter 4 for the generation of the signals $x_d[k]$ and $u_d[k]$, here we simply assume they are available.
The next step is to design an ODSMC which forces the closed-loop system to track the desired state- and output trajectory. Defining the state error and the output error as:

$$e_x[k] = x[k] - x_d[k] \qquad e_y[k] = y[k] - y_d[k] \tag{5.19}$$

and using equations (5.17) and (5.18) we can determine the state error dynamics to be:

$$\begin{aligned} e_x[k+1] &= Ae_x[k] + B(u[k] - u_d[k]) \\ e_y[k] &= Ce_x[k] \end{aligned} \tag{5.20}$$

By the nonsingular transformation $T_r \in \mathbb{R}^{n\times n}$ given by:

$$\begin{bmatrix} \bar{e}_{x_0}[k] \\ \bar{e}_{x_1}[k] \\ \bar{e}_{y_1}[k] \\ \bar{e}_{y_2}[k] \end{bmatrix} = T_\sigma e_x[k] \tag{5.21}$$

system (5.20) is brought into the following form:

$$\begin{aligned} \begin{bmatrix} \bar{e}_{x_0}[k+1] \\ \bar{e}_{x_1}[k+1] \\ \bar{e}_{y_1}[k+1] \\ \bar{e}_{y_2}[k+1] \end{bmatrix} &= \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} & \bar{A}_{14} \\ 0 & \bar{A}_{22} & \bar{A}_{23} & \bar{A}_{24} \\ 0 & \bar{A}_{32} & \bar{A}_{33} & \bar{A}_{34} \\ \bar{A}_{41} & \bar{A}_{42} & \bar{A}_{43} & \bar{A}_{44} \end{bmatrix} \begin{bmatrix} \bar{e}_{x_0}[k] \\ \bar{e}_{x_1}[k] \\ \bar{e}_{y_1}[k] \\ \bar{e}_{y_2}[k] \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \bar{B}_2 \end{bmatrix} (u[k] - u_d[k]) \\ e_y[k] &= \begin{bmatrix} 0_{p\times(n-p)} & T_y \end{bmatrix} \begin{bmatrix} \bar{e}_{x_0}[k] \\ \bar{e}_{x_1}[k] \\ \bar{e}_{y_1}[k] \\ \bar{e}_{y_2}[k] \end{bmatrix} \end{aligned} \tag{5.22}$$

We now define the switching function as:

$$\sigma_{e_y}[k] = Se_y[k] = \bar{S}_1\bar{e}_{y_1}[k] + \bar{S}_2\bar{e}_{y_2}[k] \tag{5.23}$$

Assuming the closed-loop system is in perfect sliding mode, i.e. $\sigma_{e_y}[k] = 0$, equation (5.23) can be written as:

$$\bar{e}_{y_2}[k] = -\bar{S}_2^{-1}\bar{S}_1\bar{e}_{y_1}[k] \tag{5.24}$$

By inserting equation (5.24) into equation (5.22) the reduced order dynamics in the sliding mode can be determined to be:

$$\begin{bmatrix} \bar{e}_{x_0}[k+1] \\ \bar{e}_{x_1}[k+1] \\ \bar{e}_{y_1}[k+1] \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} - \bar{A}_{14}\bar{S}_2^{-1}\bar{S}_1 \\ 0 & \bar{A}_{22} & \bar{A}_{23} - \bar{A}_{24}\bar{S}_2^{-1}\bar{S}_1 \\ 0 & \bar{A}_{32} & \bar{A}_{33} - \bar{A}_{34}\bar{S}_2^{-1}\bar{S}_1 \end{bmatrix} \begin{bmatrix} \bar{e}_{x_0}[k] \\ \bar{e}_{x_1}[k] \\ \bar{e}_{y_1}[k] \end{bmatrix} \tag{5.25}$$

Again we see that the design procedure for the sliding surface of a tracking controller is identical to the design procedure for a stabilizing ODSMC which was presented in Section 3.2. In the rest of this section it is assumed that the matrix $S = [\bar{S}_1 \;\; \bar{S}_2]T_y$ has been chosen (if possible) such that the reduced order dynamics represented by equation (5.25) are stable.

The next task is design a controller which forces the closed-loop system in the sliding mode. In order to do so we use the linear reaching law presented in Section 3.1.4, which is given by:

$$\sigma_{e_y}[k+1] = \Phi\sigma_{e_y}[k] \tag{5.26}$$

We again assume that the design parameter $\Phi \in \mathbb{R}^{m\times m}$ has been chosen diagonal with all its diagonal entries $\phi_i \;\; i = 1 \ldots m$ satisfying $0 \leq \phi_i < 1$. Then, using equations (5.26) and (5.20) we obtain:

$$\Phi\sigma_{e_y}[k] = S\left(y[k+1] - y_d[k+1]\right) \tag{5.27}$$

From equation (5.27) we have two options obtain to a controller:

- **Feedforward/Feedback Configuration:** Inserting equations (5.17) and (5.18) in equation (5.27) leads to:

$$\Phi\sigma_{e_y}[k] = S\left(CAx[k] + CBu[k] - CAx_d[k] - CBu_d[k]\right) \tag{5.28}$$

  Realizing that $CAe_x[k] = CAx[k] - CAx_d[k]$ we can rewrite equation (5.28) as:

$$u[k] = u_d[k] + (SCB)^{-1}\left(\Phi\sigma_{e_y}[k] - SCAe_x[k]\right) \tag{5.29}$$

  The control signal $u_d[k]$ can be seen as a feedforward signal since it is based on off-line computations only. The other terms at the right hand side of equation (5.14) represent the feedback part of the controller since it is based on the actual state error.

- **Tracking Controller:** Inserting equation (5.17) in equation (5.12) leads to:

$$\Phi\sigma_{e_y}[k] = S\left(CAx[k] + CBu[k] - y_d[k+1]\right) \tag{5.30}$$

  From equation (5.30) the control law can instantly be determined to be:

$$u[k] = (SCB)^{-1}\left(\Phi\sigma_{e_y}[k] + Sy_d[k+1] - SCAx[k]\right) \tag{5.31}$$

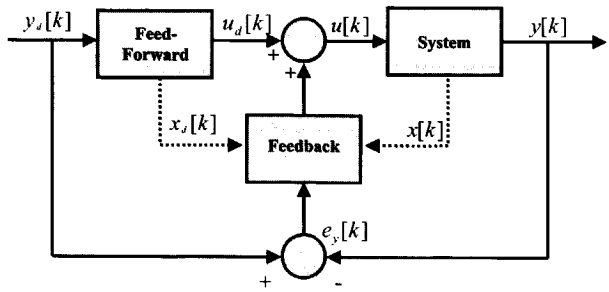


Fig. 5.3: Block diagram of the output-based feedforward/feedback configuration. The dashed arrow represents the optional use of the system state.

Despite the difference between equations (5.29) and (5.31) the resulting control signals are identical, they both satisfy the reaching law $\sigma_{e_y}[k+1] = \Phi\sigma_{e_y}[k]$. In fact by the correct manipulation of equation (5.29) or (5.31), thereby using equation (5.18), each controller representation can be rewritten into the other representation. The obvious difference between the two controller implementations is of course the presence of a feedforward control signal in the feedforward/feedback configuration. The same applied to the state-based implementations. However, another important difference between the two controller structures is that for the target controller no feedforward controller or desired state generator is required. Unlike the state-based controllers and the output-based feedforward/feedback configuration no use is made of the desired state signal nor the desired input signal.

Let us now return to the topic of either using full state information or only output information. Although the sliding surface in this section is defined in terms of the output error, both the feedforward/feedback configuration and the tracking controller assume to have full state access. To obtain a controller which is based on output information only for both the feedforward/feedback configuration we apply the transformation T_{r_y}, defined in equation (5.21), to the error state in equation (5.29) leading to:

$$u[k] = u_d[k] + (SCB)^{-1}\left(\Phi\sigma_{e_y}[k] - SCAT_r^{-1}\begin{bmatrix}\bar{e}_{x_0}[k]\\ \bar{e}_{x_1}[k]\\ \bar{e}_{y_1}[k]\\ \bar{e}_{y_2}[k]\end{bmatrix}\right) \tag{5.32}$$

Applying the same transformation to the system state in equation (5.31) leads to:

$$u[k] = (SCB)^{-1}\left(\Phi\sigma_{e_y}[k] + Sy_d[k+1] - SCAT_r^{-1}\begin{bmatrix}\bar{x}_0[k]\\ \bar{x}_1[k]\\ \bar{y}_1[k]\\ \bar{y}_2[k]\end{bmatrix}\right) \tag{5.33}$$

Still equations (5.32) and (5.33) are a different representation for exactly the same signal. However, if we only use output information and simply truncate all other

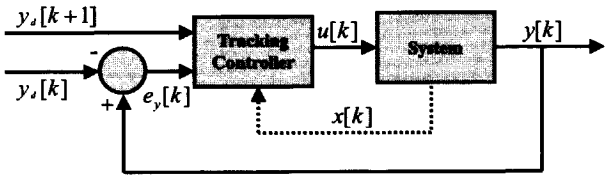


Fig. 5.4: Block diagram of the output-based tracking controller. The dashed arrow represents the optional use of the state signals.

(state-) information, equations (5.32) and (5.33) reduce to:

$$u[k] = u_d[k] + (SCB)^{-1}\left(\Phi\sigma_{e_y}[k] - SCAT_r^{-1}\begin{bmatrix}0\\0\\\bar{e}_{y_1}[k]\\\bar{e}_{y_2}[k]\end{bmatrix}\right) \tag{5.34}$$

$$u[k] = (SCB)^{-1}\left(\Phi\sigma_{e_y}[k] + Sy_d[k+1] - SCAT_r^{-1}\begin{bmatrix}0\\0\\\bar{y}_1[k]\\\bar{y}_2[k]\end{bmatrix}\right) \tag{5.35}$$

Comparing equations (5.32) and (5.34) we see that for the feedforward/feedback configuration the signals $\bar{e}_{x_0}[k]$ and $\bar{e}_{x_1}[k]$, representing the error signals $\bar{e}_{x_0}[k] = \bar{x}_0[k] - \bar{x}_{d,0}$ and $\bar{e}_{x_1}[k] = \bar{x}_1[k] - \bar{x}_{d,1}$ respectively, have been replaced by zero. Comparing equations (5.33) and (5.35) for the tracking controller we see that the state components $x_0[k]$ and $x_1[k]$ have been replaced by zero. In the case that the feedforward signal already gives approximate tracking for the true system, the error signals will be much smaller then the state signals. Therefore, the feedforward/feedback configuration is better suited for the situation where only output information is to be used. In the case where no full state information is available, the state can better be reconstructed by an observer in order to successfully employ the tracking controller. For more advanced output-based control laws the reader is referred to Section 3.2.
Figure 5.3 displays the block diagram of the output-based feedforward controller, whereas Figure 5.4 depicts the output-based tracking controller. In both figures, the dashed arrow represents the optional use of the system state.

5.2 Model Reference Control

The controllers introduced in Section 5.1 aim at perfect tracking of some specified desired output trajectory $y_d[k]$. In this section another approach is taken. Instead of the desire of perfectly replicating some desired output trajectory, this section is devoted to the replication of a desired closed-loop behaviour. This desired closed-loop behaviour is defined by a reference model from which the name model reference control originates. The model reference approach in combination with CSMC can be found, for both the state-based and the output-based case, in for

example (Edwards and Spurgeon 1998). This chapter presents the straightforward discrete-time implementation. The reference model is defined by:

$$w[k+1] = Gw[k] + Hr[k] \tag{5.36}$$

where $w \in \mathbb{R}^n$, $r \in \mathbb{R}^q$, $G \in \mathbb{R}^{n \times n}$, and $H \in \mathbb{R}^{n \times q}$. Furthermore, it is assumed that the pair (G, H) is controllable and the matrix G has stable eigenvalues, hence is the reference system is stable. We now seek a controller which results in perfect tracking by system (5.2) of the reference state $w[k]$. First we determine some criteria for the reference model (G, H) to be tractable by the system (5.2).

Theorem 5.2.1 *The state $w[k]$, generated by the reference model (5.36), is tractable by the system (5.2) if, and only if, the matrices $(A - G)$ and H can be written as:*

$$A - G = B\delta_{AG} \qquad\qquad H = B\delta_H \tag{5.37}$$

for some finite $\delta_{AG} \in \mathbb{R}^{m \times n}$ and $\delta_H \in \mathbb{R}^{m \times q}$.
Proof : *We define the tracking error e_{xw} as:*

$$e_{xw}[k] = x[k] - w[k] \tag{5.38}$$

Using equations (5.2), (5.36), and (5.38) we can determine the error dynamics to be:

$$e_{xw}[k+1] = Ae_{xw}[k] + (A - G)w[k] - Hr[k] + Bu[k] \tag{5.39}$$

If condition (5.37) is fulfilled, equation (5.39) can be written as:

$$e_{xw}[k+1] = Ae_{xw}[k] + B\left(\delta_{AG}w[k] - \delta_H r[k] + u[k]\right) \tag{5.40}$$

By the transformation $T_{r_x} \in \mathbb{R}^{n \times n}$, given by:

$$\begin{bmatrix} \bar{e}_{xw,1}[k] \\ \bar{e}_{xw,2}[k] \end{bmatrix} = T_{r_x} e_{xw}[k] \tag{5.41}$$

we bring the error system (5.39) into the regular form:

$$\begin{aligned} \begin{bmatrix} \bar{e}_{xw,1}[k+1] \\ \bar{e}_{xw,2}[k+1] \end{bmatrix} &= \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{e}_{xw,1}[k] \\ \bar{e}_{xw,2}[k] \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 \\ \bar{B}_2 \end{bmatrix} \left(u[k] + \delta_{AG}w[k] - \delta_H r[k]\right) \end{aligned} \tag{5.42}$$

Assuming that at some initial time k_0 the state error is zero, i.e. $\bar{e}_{xw,1}[k_0] = 0$ and $\bar{e}_{xw,2}[k_0] = 0$, it can be seen from equation (5.42) that it is sufficient to choose for $u[k]$ $\forall$ $k \geq k_0$:

$$u[k] = \delta_H r[k] - \delta_{AG}w[k] \tag{5.43}$$

For the case that condition (5.37) is not fulfilled, applying the transformation T_{r_x} to equation (5.39) leads to:

$$\begin{aligned}\begin{bmatrix} \bar{e}_{xw,1}[k+1] \\ \bar{e}_{xw,2}[k+1] \end{bmatrix} &= \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{e}_{xw,1}[k] \\ \bar{e}_{xw,2}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{B}_2 \end{bmatrix} u[k] \\ &+ \begin{bmatrix} T_{r_x,1}(A-G) \\ T_{r_x,2}(A-G) \end{bmatrix} w[k] - \begin{bmatrix} T_{r_x,1}H \\ T_{r_x,2}H \end{bmatrix} r[k]\end{aligned} \tag{5.44}$$

where T_{r_x} has been partitioned as:

$$T_{r_x} = \begin{bmatrix} T_{r_x,1} \\ T_{r_x,2} \end{bmatrix} \tag{5.45}$$

with $T_{r_x,1} \in \mathbb{R}^{(n-m)\times n}$ and $T_{r_x,2} \in \mathbb{R}^{m\times n}$. Since condition (5.37) is not fulfilled, either $T_{r_x,1}(A-G)$ or $T_{r_x,1}H$ (or both) are nonzero. Therefore, there does not exist a control input which cancels all terms driving the error dynamics. Which completes the proof. □

Theorem 5.2.1 shows that the matrices $(A-G)$ and H should both be matched with the input matrix B. A possible interpretation is to see the signals $(A-G)w[k]$ and $Hr[k]$ as measurable disturbances working on the error system. It is already known from SMC theory that only matched disturbances can be canceled out by the controller, which explains Theorem 5.2.1. In the following sections we define a SDSMC and an ODSMC which forces the closed-loop system to track the reference system (5.36).

5.2.1 State-Based Model Reference Sliding Mode Control

In this section we define a SDSMC which forces the system (5.2) to track the reference model (5.36). It is assumed that condition (5.37) is fulfilled. Consequently the error dynamics in the regular form are given by (see Theorem 5.2.1):

$$\begin{aligned}\begin{bmatrix} \bar{e}_{xw,1}[k+1] \\ \bar{e}_{xw,2}[k+1] \end{bmatrix} &= \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{e}_{xw,1}[k] \\ \bar{e}_{xw,2}[k] \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ \bar{B}_2 \end{bmatrix} (u[k] + \delta_{AG} w[k] - \delta_H r[k])\end{aligned} \tag{5.46}$$

The switching function is defined as:

$$\sigma_{xw}[k] = S(x[k] - w[k]) = \bar{S}_1 \bar{e}_{xw,1}[k] + \bar{S}_2 \bar{e}_{xw,2}[k] \tag{5.47}$$

Then, if we assume that the system is driven into the sliding mode, i.e. $\sigma_{xw}[k] = 0$, we can write equation (5.47) as:

$$\bar{e}_{xw,2}[k] = -\bar{S}_2^{-1} \bar{S}_1 \bar{e}_{xw,1}[k] \tag{5.48}$$

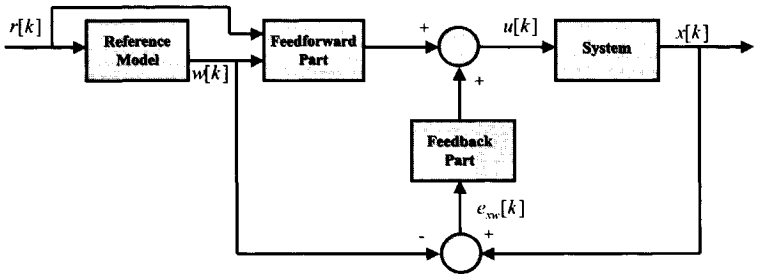


Fig. 5.5: Block diagram of the state-based model reference sliding mode controller.

Inserting equation (5.48) into the representation for $\bar{e}_{xw,1}[k]$, which can be found from equation (5.46), leads to the reduced order dynamics in the sliding mode:

$$\bar{e}_{xw,1}[k+1] = \left(\bar{A}_{11} - \bar{A}_{12}\bar{S}_2^{-1}\bar{S}_1\right)\bar{e}_{xw,1}[k] \tag{5.49}$$

Again, the matrices $\bar{S}_1$ and $\bar{S}_2$ should be chosen such that the above reduced order dynamics are stable. Interesting to notice is that the design procedure for the sliding surface is again identical to the design of the sliding surface for a stabilizing SDSMC, as was the case for a SDSMC tracking controller.
The next step is the design of the controller which forces the closed-loop system in the sliding mode. Therefore, we define the linear reaching law as :

$$\sigma_{xw}[k+1] = \Phi\sigma_{xw}[k] \tag{5.50}$$

where $\Phi \in \mathbb{R}^{m\times m}$ being diagonal with all its diagonal entries $\phi_i \quad i = 1\ldots m$ satisfying $0 \leq \phi_i < 1$. From equations (5.2), (5.36) and (5.50) we can determine the control law to be:

$$\begin{aligned} u[k] &= (SB)^{-1}\left(S(G-A)w[k] + SHr[k]\right) \\ &\quad +(SB)^{-1}\left(\Phi\sigma_{xw}[k] - SAe_{xw}[k]\right) \end{aligned} \tag{5.51}$$

The first part of equation (5.51) is purely based on off-line computed variables ($r[k]$ and $w[k]$) and hence can be considered as a feedforward control signal. The second part of equation (5.51) is based on the measured state error and hence can be considered as the feedback component of the controller. This can be seen in the block diagram presented in Figure 5.5.

5.2.2 Output-Based Model Reference Sliding Mode Control

In this section an output-based model reference DSMC is constructed. Defining the state error and the output error as:

$$e_{xw}[k] = x[k] - w[k] \qquad e_{yw}[k] = y[k] - Cw[k] \tag{5.52}$$

and using equations (5.2) and (5.36) we can determine the state error dynamics to be:

$$\begin{aligned} e_{xw}[k+1] &= Ae_{xw}[k] + Bu[k] - (G-A)w[k] - Hr[k] \\ e_{yw}[k] &= Ce_{xw}[k] \end{aligned} \tag{5.53}$$

Assuming that condition (5.37) is fulfilled, applying the nonsingular transformation $T_r \in \mathbb{R}^{n\times n}$ given by:

$$\begin{bmatrix} \bar{e}_{xw_0}[k] \\ \bar{e}_{xw_1}[k] \\ \bar{e}_{yw_1}[k] \\ \bar{e}_{yw_2}[k] \end{bmatrix} = T_r e_{xw}[k] \tag{5.54}$$

to system (5.53) results in:

$$\begin{aligned} \begin{bmatrix} \bar{e}_{xw_0}[k+1] \\ \bar{e}_{xw_1}[k+1] \\ \bar{e}_{yw_1}[k+1] \\ \bar{e}_{yw_2}[k+1] \end{bmatrix} &= \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} & \bar{A}_{14} \\ 0 & \bar{A}_{22} & \bar{A}_{23} & \bar{A}_{24} \\ 0 & \bar{A}_{32} & \bar{A}_{33} & \bar{A}_{34} \\ \bar{A}_{41} & \bar{A}_{42} & \bar{A}_{43} & \bar{A}_{44} \end{bmatrix} \begin{bmatrix} \bar{e}_{xw_0}[k] \\ \bar{e}_{xw_1}[k] \\ \bar{e}_{yw_1}[k] \\ \bar{e}_{yw_2}[k] \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \bar{B}_2 \end{bmatrix} (u[k] + \delta_{AG} w[k] - \delta_H r[k]) \\ e_{yw}[k] &= \begin{bmatrix} 0_{p\times(n-p)} & T_y \end{bmatrix} \begin{bmatrix} \bar{e}_{xw_0}[k] \\ \bar{e}_{xw_1}[k] \\ \bar{e}_{yw_1}[k] \\ \bar{e}_{yw_2}[k] \end{bmatrix} \end{aligned} \tag{5.55}$$

We now define the switching function as:

$$\sigma_{e_{yw}}[k] = S e_{yw}[k] = \bar{S}_1 \bar{e}_{yw_1}[k] + \bar{S}_2 \bar{e}_{yw_2}[k] \tag{5.56}$$

Assuming the closed-loop system is in perfect sliding mode, i.e. $\sigma_{e_{yw}}[k] = 0$, equation (5.56) can be written as:

$$\bar{e}_{yw_2}[k] = -\bar{S}_2^{-1} \bar{S}_1 \bar{e}_{yw_1}[k] \tag{5.57}$$

By inserting equation (5.57) into equation (5.55) the reduced order dynamics in the sliding mode can be determined to be:

$$\begin{bmatrix} \bar{e}_{xw_0}[k+1] \\ \bar{e}_{xw_1}[k+1] \\ \bar{e}_{yw_1}[k+1] \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} - \bar{A}_{14}\bar{S}_2^{-1}\bar{S}_1 \\ 0 & \bar{A}_{22} & \bar{A}_{23} - \bar{A}_{24}\bar{S}_2^{-1}\bar{S}_1 \\ 0 & \bar{A}_{32} & \bar{A}_{33} - \bar{A}_{34}\bar{S}_2^{-1}\bar{S}_1 \end{bmatrix} \begin{bmatrix} \bar{e}_{xw_0}[k] \\ \bar{e}_{xw_1}[k] \\ \bar{e}_{yw_1}[k] \end{bmatrix} \tag{5.58}$$

Again we see that the design procedure for the sliding surface of a tracking controller is identical to the design procedure for a stabilizing ODSMC which was presented in Section 3.2. In the rest of this section it is assumed that the matrix $S = [\bar{S}_1 \;\; \bar{S}_2]T_y$ has been chosen (if possible) such that the reduced order dynamics represented by equation (5.58) are stable.

The next task is design a controller which forces the closed-loop system in the sliding mode. In order to do so we use the linear reaching law presented in Section 3.1.4, which is given by:

$$\sigma_{e_{yw}}[k+1] = \Phi \sigma_{e_{yw}}[k] \tag{5.59}$$

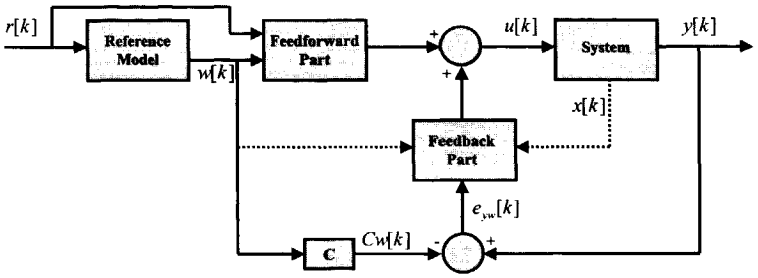


Fig. 5.6: Block diagram of the output-based model reference sliding mode controller. The dashed arrows represent the optional use of the state signals.

We again assume that the design parameter $\Phi \in \mathbb{R}^{m \times m}$ has been chosen diagonal with all its diagonal entries ϕ_i, $i = 1 \ldots m$ satisfying $0 \leq \phi_i < 1$. Then, using equations (5.59) and (5.53) we obtain:

$$\Phi \sigma_{e_{yw}}[k] = S\left(CAe_{xw}[k] + CBu[k] - C(G - A)w[k] - CHr[k]\right) \tag{5.60}$$

Rewriting equation (5.60) leads to the control law:

$$\begin{aligned} u[k] &= (SCB)^{-1}\left(SCHr[k] + SC(G - A)w[k]\right) \\ &\quad +(SCB)^{-1}\left(\Phi\sigma_{e_{yw}}[k] - SCAe_{xw}[k]\right) \end{aligned} \tag{5.61}$$

In the case where only the measured outputs are available the above control law is not implementable. Using transformation T_r, given by equation (5.54), equation (5.61) can be rewritten as:

$$\begin{aligned} u[k] &= (SCB)^{-1}\left(\Phi\sigma_{e_{yw}}[k] + SCHr[k] + SC(G - A)w[k]\right. \\ &\quad \left. -SCAT_r^{-1}\begin{bmatrix} \bar{e}_{xw_0}[k] \\ \bar{e}_{xw_1}[k] \\ \bar{e}_{yw_1}[k] \\ \bar{e}_{yw_2}[k] \end{bmatrix}\right) \end{aligned} \tag{5.62}$$

In the above equation, all time signals except $\bar{e}_{xw_0}[k]$ and $\bar{e}_{xw_1}[k]$ are known, either by measurement or off-line computations. The unknown entries could be approximated by zero, leading to:

$$\begin{aligned} u[k] &= (SCB)^{-1}\left(SCHr[k] + SC(G - A)w[k]\right) \\ &\quad +(SCB)^{-1}\left(\Phi\sigma_{e_{yw}}[k] - SCAT_r^{-1}\begin{bmatrix} 0 \\ 0 \\ \bar{e}_{yw_1}[k] \\ \bar{e}_{yw_2}[k] \end{bmatrix}\right) \end{aligned} \tag{5.63}$$

For more advanced output-based control laws the reader is referred to Section 3.2. In Figure 5.6 the block diagram of the output-based model reference controller can be found.

Variable	Model	System	unit
m_c	200	300	[kg]
m_w	33	30	[kg]
c_c	9000	7000	[N/m]
c_w	$20e^3$	$22e^3$	[N/m]
d_c	1200	1100	[Nsec/m]

Table 5.1: Variable values of the model and the system, Section 5.

5.3 Simulation Examples

5.3.1 Target Tracking Simulation Example

As a simulation example we again consider the target tracking problem presented in Section 4.4 for the feedforward controller. In Section 4.4 a feedforward controller was designed which leads to perfect tracking of the desired output trajectory if the trajectory is known fully in advance. However, due to modeling errors and external disturbances, the obtained input signal does not suffice to obtain perfect tracking for the closed-loop system. Therefore, we extend the controller setup with a feedback controller, as presented in Section 5.1,

For the purpose of feedback control we assume that two output signals are available, hence the system may be represented by:

$$\begin{aligned} x(t) &= A_c x(t) + B_c u(t) \\ y(t) &= C_c x(t) \end{aligned} \tag{5.64}$$

The system state, input and output are given by:

$$x(t) = [x_c(t) \quad x_w(t) \quad \dot{x}_c(t) \quad \dot{x}_w(t)]^T \quad y(t) = [\dot{x}_c(t) \quad \ddot{x}_c(t)] \quad u(t) = x_b(t) \tag{5.65}$$

The variables x_c, x_w, and x_b represent the car, wheel and base displacement respectively. The system matrices are given by:

$$\begin{aligned} A_c &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{c_c}{m_c} & \frac{c_c}{m_c} & -\frac{d_c}{m_c} & \frac{d_c}{m_c} \\ \frac{c_c}{m_w} & -\frac{c_c+c_w}{m_w} & \frac{d_c}{m_w} & -\frac{d_c}{m_w} \end{bmatrix} \\ B_c &= \begin{bmatrix} 0 & 0 & 0 & \frac{c_w}{m_w} \end{bmatrix}^T \\ C_c &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ -\frac{c_c}{m_c} & \frac{c_c}{m_c} & -\frac{d_c}{m_c} & \frac{d_c}{m_c} \end{bmatrix} \end{aligned} \tag{5.66}$$

where m_w is the mass of the wheel, m_c is one quarter of the mass of the car, c_w is the wheel stiffness, c_c the suspension stiffness and d_c the suspension damping. In Table 5.1 the parameters for the model and the system under control control can be found. The matrices (A_c, B_c, C_c) are discretised by sample and hold to obtain the matrices (A, B, C) for the discrete-time model (sampling time $T_s = 5$ $[ms]$.:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] \end{aligned} \tag{5.67}$$

The discretised model has relative degree $r = 1$, and there are no unstable zeros. We use the output-based feedforward/feedback control configuration, given by (see equation (5.29) in Section 5.1.2):

$$u[k] = u_d[k] + (SCB)^{-1} \left(\Phi\sigma_{e_y}[k] - SCAe_x[k]\right) \tag{5.68}$$

The feedforward control signal u_d was already obtained in Section 4.4. The signal $e_x[k]$ is given by $e_x[k] = x[k] - x_d[k]$ and the switching function σ_{e_y} is defined as:

$$\sigma_{e_y}[k] = S\left(y[k] - y^d[k]\right) \tag{5.69}$$

where the signal $y^d[k]$ is the desired output trajectory. Worthwhile to mention is that the desired output trajectory in Section 5.1.2 was defined for x_c which is not a measured output signal. However, since in Section 5.1.2 the desired state trajectory x_d is computed, the desired trajectory for the measured output signals is known as well.

We now design an ODSMC to stabilize the error dynamics. In Section 3.2 several ODSMC implementation were presented. Only the direct linear controller was presented in this chapter. For the simulation example we will also use the direct linear controller with disturbance estimation and the direct linear controller with reduced order state observer. For all sliding mode controllers $\Phi = 0$ and $S = [0.123 \quad 0.00058]$ are used. The reduced order observer gain L is given by: $L = [-20.28 \quad -190.25]$ which results in a dead-beat observer (all observer poles at zero). These parameters would lead to the following poles in perfect sliding mode (i.e. $\sigma_{e_y} = 0$): $p_{sm} = \{1 \quad 0.963 \quad 0.898 \quad 0\}$. However, as was already presented in Section 3.2, this will not be the case for the closed-loop system. In fact, computing the eigenvalues of the matrices given in the before mentioned Lemma's, results in the following closed-loop poles:

$$\begin{aligned} p_{smc}(y,r) &= \{0.784 \quad 0.986 \pm 0.025i \quad 0.104\} \\ p_{smcdes}(y,r) &= \{1 \quad 0.960 \quad 0.912 \quad 0.005 \pm 0.301i\} \\ p_{smcseo}(y,r) &= \{1 \quad 0.963 \quad 0.898 \quad 0 \quad 0.001 \pm 0.040i\} \end{aligned} \tag{5.70}$$

where the subscripts *smc* stands for the sliding mode controller without state error observer, *smcdes* stands for the sliding mode controller with disturbance estimation, and *smcseo* for the sliding mode controller with state error observer. Clearly, the closed-loop poles with the direct linear controller differ substantially from the designed sliding mode poles. However, they are still stable. The poles with the direct linear controller with disturbance estimation are closer to the designed sliding mode poles, where the two poles close to zero are the result of the reaching pole ($\Phi = 0$) and the delay due to disturbance estimation. Finally, the poles of the direct sliding mode controller with the reduced order state-error observer match the designed sliding mode poles again. Furthermore there are two poles close to zero added because of the dead-beat observer.

The performance of the controllers for the given target signal is presented in Figure 5.7. To have a measure of performance of the controllers we define the Variance

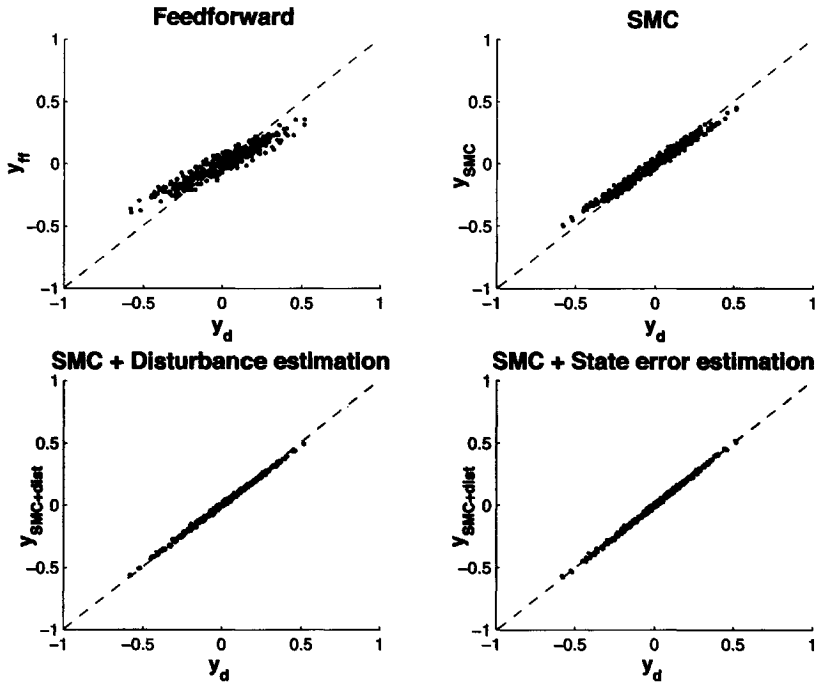


Fig. 5.7: xy-plots for the feedforward controller (top-left), the direct linear controller (top-right), the direct linear controller with disturbance estimation (bottom-left), and the direct linear controller with state error estimation (bottom-right). The dashed diagonal line represents perfect tracking.

Accounted For (VAF) of the output as:

$$\mathrm{VAF}(y, y^d) = \left(1 - \frac{\mathrm{variance}(y - y^d)}{\mathrm{variance}(y)}\right) \tag{5.71}$$

Computing the individual VAF's for the controllers results in:

$$\begin{aligned} \mathrm{VAF}_{ff}(y, y^d) &= \begin{bmatrix} 73.3 \\ 81.6 \end{bmatrix} \qquad \mathrm{VAF}_{smc}(y, y^d) = \begin{bmatrix} 86.8 \\ 96.1 \end{bmatrix} \\ \mathrm{VAF}_{smcdes}(y, y^d) &= \begin{bmatrix} 99.9 \\ 99.6 \end{bmatrix} \quad \mathrm{VAF}_{smcseo}(y, y^d) = \begin{bmatrix} 100.0 \\ 99.7 \end{bmatrix} \end{aligned} \tag{5.72}$$

where the subscript ff stands for the feedforward controller only, and the other subscripts as before. It clearly shows that sliding mode feedback control added to the feedforward controller increases the performance considerably. Especially the direct linear controller with either disturbance estimation or a reduced order state-error observer performs very well, achieving nearly perfect tracking even in the case of severe parameter errors.
Another measure of performance for the sliding mode controllers would be the RMS (Root Mean Square) value of the switching variable σ_{e_y}, which ideally would be zero. From simulation it was computed:

$$\begin{aligned} \mathrm{RMS}_{ff}(\sigma_{e_y}) &= 9.3e^{-4} \qquad \mathrm{RMS}_{smc}(\sigma_{e_y}) = 3.5e^{-4} \\ \mathrm{RMS}_{smcdes}(\sigma_{e_y}) &= 7.3e^{-5} \quad \mathrm{RMS}_{smcseo}(\sigma_{e_y}) = 5.5e^{-5} \end{aligned} \tag{5.73}$$

The above numbers result in the same conclusions as the computed VAF's.
Combining disturbance estimation with a state-error observer is possible as well. For the given desired output and the same parameters as for the already presented controllers the VAF and the RMS values are given by:

$$\mathrm{VAF}_{tot}(y, y^d) = \begin{bmatrix} 100.0 \\ 99.8 \end{bmatrix} \quad \mathrm{RMS}_{tot}(\sigma_{e_y}) = 4.0e^{-5} \tag{5.74}$$

which is again a slight improvement over the previous results.

5.3.2 Model Reference Simulation Example

In this section we consider the pitch-pointing flight controller which was solved using a continuous-time model reference sliding mode controller in (Edwards and Spurgeon 1998). In this section we assume that the airplane is in fact computer controlled and that all control signals are of the sample and hold type. The problem at hand is the decoupling of the pitch angle and the altitude of an aircraft. This implicates that the pitch of the aircraft can be changed without affecting the altitude and conversely the altitude can be changed without affecting the pitch. In (Edwards and Spurgeon 1998) the following model is used which corresponds to an altitude of 3048 m and 0.77 $Mach$:

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{5.75}$$

where the state vector $x \in \mathbb{R}^5$ and the input vector $u \in \mathbb{R}^2$ represent:

$$x(t) = \begin{bmatrix} \gamma(t) \\ q(t) \\ \alpha(t) \\ \eta(t) \\ \delta(t) \end{bmatrix} \begin{array}{l} \text{flight path angle [rad]} \\ \text{pitch rate [rad/sec]} \\ \text{angle of attack [rad]} \\ \text{elevator deflection [rad]} \\ \text{flap deflection [rad]} \end{array} \tag{5.76}$$

$$u(t) = \begin{bmatrix} \eta_c(t) \\ \delta_c(t) \end{bmatrix} \begin{array}{l} \text{elevator command [rad]} \\ \text{flap command [rad]} \end{array}$$

The matrices (A_c, B_c) are given by:

$$A_c = \begin{bmatrix} 0 & 0 & 1.74 & 0.08 & 0.59 \\ 0 & -1.99 & -13.41 & -18.95 & -3.60 \\ 0 & 1.00 & -1.74 & -0.08 & -0.59 \\ 0 & 0 & 0 & -20 & 0 \\ 0 & 0 & 0 & 0 & -20 \end{bmatrix} \quad B_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 20 & 0 \end{bmatrix} \tag{5.77}$$

It is assumed that the aircraft is computer controlled by sample and hold with a sampling time of $T_s = 50$ $[ms]$ therefore we discretize the continuous-time model. In (Edwards and Spurgeon 1998) a reference model is defined which places the eigenvalues at the desired locations:

$$\begin{array}{rcll} \lambda_{1,2} & = & -5.6 \pm 4.2j & \text{short period mode} \\ \lambda_3 & = & -1.0 & \text{flight path mode} \\ \lambda_4 & = & -20 & \text{elevator actuator mode} \\ \lambda_5 & = & -20 & \text{flap actuator mode} \end{array} \tag{5.78}$$

The reference model is discretized with sampling time $T_s = 50$ $[ms]$ as well, leading to:

$$w[k+1] = Gw[k] + Hr[k] \tag{5.79}$$

where (G, H) are found to be:

$$G = \begin{bmatrix} 0 & 0 & 1.74 & 0.08 & 0.59 \\ 0 & -1.99 & -13.41 & -18.95 & -3.60 \\ 0 & 1.00 & -1.74 & -0.08 & -0.59 \\ 38.62 & 11.49 & 42.28 & -29.50 & -2.13 \\ -38.88 & -4.48 & -62.11 & 1.39 & -18.97 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -52.99 & 14.37 \\ 6.76 & 32.12 \end{bmatrix} \tag{5.80}$$

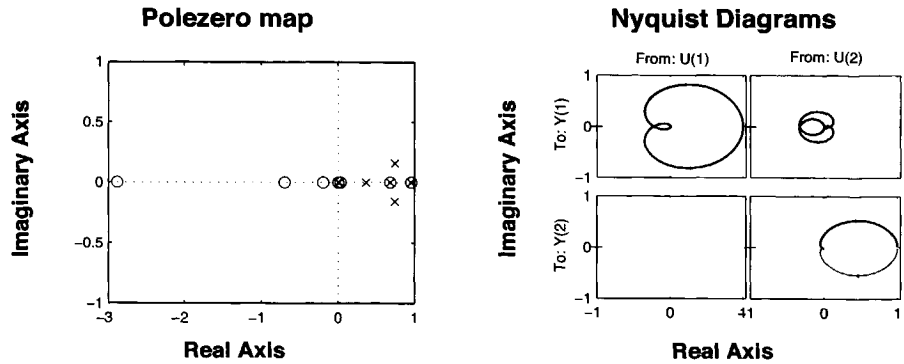


Fig. 5.8: The pole-zero map (left plot) and the Nyquist diagram (right plots) for the closed-loop system including the reference model.

The sliding surface for the DSMC is designed by LQR design leading to (where $Q = I_{n-m}$ and $R = 1e^{-2}I_m$ have been used):

$$S = \begin{bmatrix} -2.99 & -1.64 & 0.39 & 0.72 & -0.10 \\ 7.54 & -0.12 & -7.28 & -0.17 & 1.3112 \end{bmatrix} \tag{5.81}$$

which, in the sliding mode, results in the discrete-time poles for the error dynamics given by $p_{sm} = \{0.9552, 0.6785, 0.0121\}$. The variable Φ has been chosen as $\Phi = 0_{2\times 2}$. Using controller equation (5.51), we can complete the controller design. The pole-zero map and the Nyquist diagrams for the closed-loop system including the reference model are shown in Figure 5.8. The closed-loop system has the poles:

$$p_{cl} = \{0.9552, 0.6785, 0.0121, 0.0000, 0.0000, 0.9513, 0.7390 \pm 0.1575i, 0.3680, 0.3680\}$$

The first three poles corresponds to the poles representing the sliding motion. The two poles in the origin are those assigned by Φ and the remaining poles are the reference model poles. As the pole-zero map in Figure 5.8 there is one zero outside the unit circle ($z_{cl,1} = -2.8902$) making the closed-loop system nonminimum phase. From the Nyquist plots in Figure 5.8 the gain margin G_m and the phase margin P_m can be determined to be:

$$G_m = \begin{bmatrix} \infty & \infty \\ \infty & \infty \end{bmatrix} \qquad P_m = \begin{bmatrix} 179.64^o & \infty \\ \infty & \infty \end{bmatrix}$$

Figure 5.9 presents the simulation results for a stepwise reference change for the pitch signal. While the pitch makes a fast but smooth transition from 0 radians (0 degrees) to 1 radians (57 degrees), the flight-path only marginally deviates from the desired flightpath, with a maximum error of $6e^{-3}$.

5.4 Summary

In this chapter the problem of output tracking is considered. The problem is solved using two different approaches, both using DSMC. The first approach (Section 5.1)

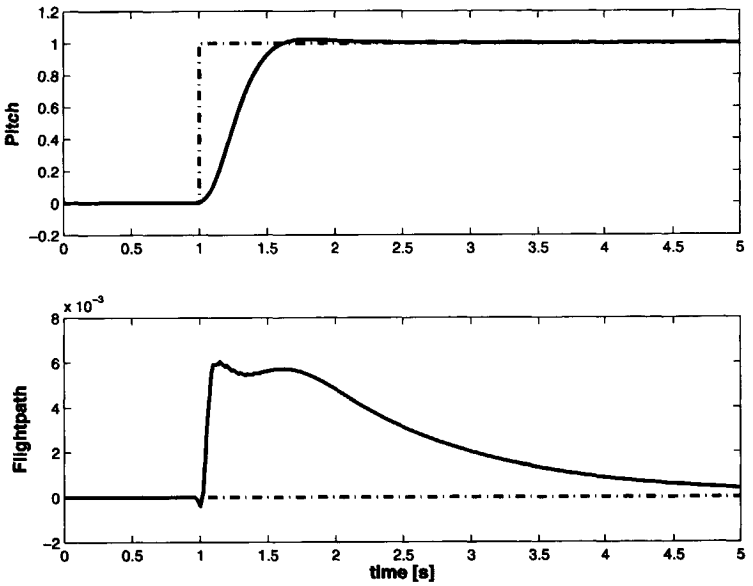
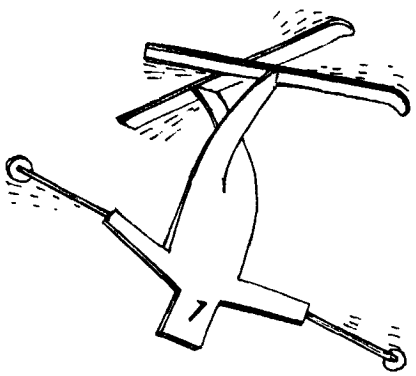


Fig. 5.9: Simulation result for the state-based model reference controller. The dashed line represents the reference signal, the solid line the achieved signal.

aims at perfect tracking of some arbitrary, tractable, output trajectory. Both a state-based and an output-based controller was defined. The second approach (Section 5.2) aims at tracking some desired model, again a state-based and an output-based implementation are presented. The control approach was illustrated by two simulation examples in Section 5.3.
The idea of a feedforward/feedback configuration has been presented in (Monsees et al. 1999) for nonlinear CSMC. The use of a ODSMC for target tracking has been presented in (Monsees and Scherpen 2001*a*) and (Monsees and Scherpen 2001*d*).

6



Laboratory Test Results

In this chapter, three laboratory control problems are solved using the control techniques introduced in Chapters 3, 4, and 5. The first control problem consists of a cart, fixed to a rail, which has to be positioned at some step-wise constant position. The system has one input and is modeled by a second-order linear model. This positioning problem is solved in Section 6.1 by the application of a SDSMC as defined in Chapter 3. The second control problem is formed by a beam, balancing on a fixed pole, which can freely move in the vertical plane (i.e. the pitch direction). The beam is actuated by a propeller, driven by a DC-motor, by which the pitch direction can be influenced. This pitch control problem is solved in Section 6.2 using the following controller structures:

1. Stabilizing SDSMC as introduced presented in Section 3.1.
2. Tracking SDSMC as introduced in Section 5.1.
3. Model Reference SDSMC as introduced in Section 5.2.
4. Model Reference SCSMC. In this case it is assumed that the closed-loop system is continuous-time, thereby neglecting the sampling issue.

As can be seen in the above list, the model reference controller is implemented using SDSMC as well as SCSMC. Both implementations are tested for a relatively high- and a relatively low sampling frequency in order to obtain a comparison between DSMC and CSMC.
Finally, in Section 6.3 the simultaneous control of the pitch and yaw direction of the beam is considered, making it a MIMO control problem. Again the stabilizing-, tracking-, and model reference SDSMC approach are demonstrated.

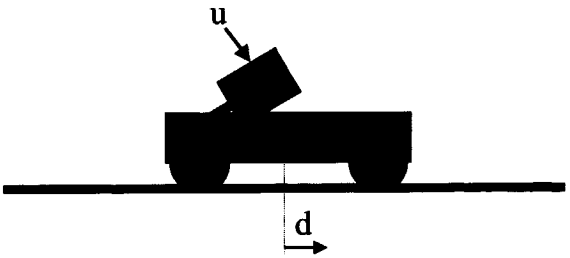


Fig. 6.1: Cart test setup. The position of the cart, denoted by d, is to be stabilized at specified locations.

6.1 Cart Position Control

This section considers the position control of a cart fixed to a rail. Although quite simple, this control problem demonstrates the application of DSMC to a real-life control problem. The cart is driven by a DC-motor. A sketch of the system is presented in Figure 6.1. For the design of the controller, only the dynamics of the cart are considered. The dynamics of the DC-motor, the sensor dynamics, and the stiction are neglected. Instead, the system is described by the second-order linear set of equations:

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{6.1}$$
$$y(t) = C_c x(t) \tag{6.2}$$

where:

$$x = \begin{bmatrix} d(t) \\ \dot{d}(t) \end{bmatrix} \quad y(t) = d(t) \tag{6.3}$$

The variable d denotes the position of the cart, as can be seen in Figure 6.1, and consequently $\dot{d}$ stands for the velocity of the cart. The subscript c for the matrices (A_c, B_c, C_c) denotes the continuous-time, they can be found to be:

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & -c_{fr} \end{bmatrix} \quad B_c = \begin{bmatrix} 0 \\ \frac{k_m}{m} \end{bmatrix} \quad C_c = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{6.4}$$

where $k_m = 5\ [N/V]$ represents the DC-motor gain, $m = 0.49\ [kg]$ the mass of the cart, and $c_{fr} = 7\ [Nm/s]$ the friction coefficient.
The obtained model, as well as the system itself, is continuous-time. However, the closed-loop system will be formed by the continuous-time system together with a discrete-time controller, making it a sampled-data system. Therefore, we discretize the obtained continuous-time system by sample-and-hold with the sampling time $T_s = 10\ ms$, resulting in:

$$x[k+1] = Ax[k] + Bu[k] \tag{6.5}$$
$$y[k] = Cx[k] \tag{6.6}$$

The triple (A, B, C) proofs to be controllable and observable. The open-loop poles are given by $p_{ol} = \{1.0000, 0.8669\}$ and the system has one zero at $z_{ol} = -0.9535$. The task of the controller is to stabilize the cart at step-wise constant positions. Therefore, the desired state trajectory is constructed as:

$$x_d[k] = \begin{bmatrix} r[k] \\ 0 \end{bmatrix} \tag{6.7}$$

where $r[k]$ is the piece-wise constant reference trajectory. The switching function is defined as:

$$\sigma_{e_x}[k] = S(x[k] - x_d[k]) \tag{6.8}$$

The parameter S is set to $S = [110.59 \quad 9.95]$ leading to a pole in the sliding mode at $p_{sm} = 0.9672$. Using the linear reaching law $\sigma[k+1] = \Phi\sigma[k]$, the control law can be determined to be:

$$u[k] = (SB)^{-1}(\Phi\sigma[k] - SAe_x[k]) \tag{6.9}$$

Choosing $\Phi = 0$ reduces the above control law to $u[k] = -SAe_x[k]$. Since setting the parameter Φ to zero implicates that the controller aims at reaching the sliding surface in one time step, excessively large control inputs may be generated. Therefore, the control input is limited to ± 10 $[V]$.

In the presented design procedure it was assumed that the system state is available to the controller. However, since only the cart position is measured, the cart velocity needs to be replicated by an observer. The observer equation is given by:

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] - L(C\hat{x}[k] - y[k]) \tag{6.10}$$

where L is selected as $L = [0.16 \quad 0.48]^T$ leading to the observer poles $p_{obs} = 0.8518 \pm 0.0651i$.

Since the obtained controller structure is completely linear, the resulting closed-loop system can be analyzed using classical control theory. The pole-zero map and the Nyquist diagram are given in Figure 6.2. The pole in the sliding mode was selected to be $p_{sm} = 0.9672$ and Φ was selected to be zero, both poles can be identified in the pole-zero map ($p_{cl,1,2} = \{0.0204, 0.9671\}$). The other two pole are the observer poles, which are cancelled by two zeros at the same location ($p_{cl,3,4} = \{0.8518 \pm 0.0651i\}$). The zero $z_{cl} = -0.9535$ constitutes the open-loop zero.

In Figure 6.3 the test results for a typical reference trajectory are presented. Despite the severe stiction, the results demonstrate a small steady-state error. Furthermore, even though the controller is driven into saturation (maximum control input is provided during the transients), the output does not exhibit any overshoot. This can be explained by observing that, although the system does not stay in the quasi sliding mode during the transitions, it is driven back into the quasi sliding mode before the desired position is reached as shown in Figure 6.4. In the quasi sliding mode, the dynamics of the closed-loop system are approximately described by a (stable) first-order equation. Which explains the fact that the cart position does not have any overshoot.

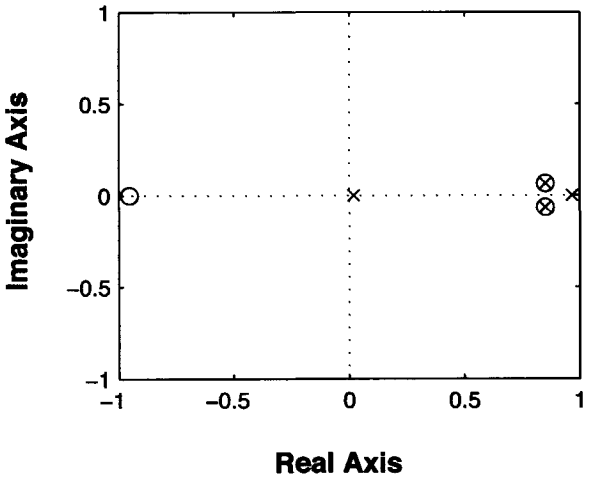


Fig. 6.2: Closed-loop poles and zeros.

6.2 Pitch Control

In this section the control of a beam, actuated by a propeller, which balances on a fixed pole is considered. The propeller is driven by a DC-motor, which is in turn driven by the controller. A sketch of the test setup is given in Figure 6.5. The variables θ, u_θ, and ω_θ, representing the yaw angle, yaw propeller speed and yaw control input respectively, are constrained to zero in this section. The variable φ, representing the pitch angle, is measured. The control input u_φ, representing the voltage applied to the DC-motor, is to be generated by the controller. To come to a DSMC, we first obtain a simple linear model for the system, for which we construct an observer as well to reconstruct the unmeasured system states. Then, the following controller structures are presented and tested on the test-rig:

1. Stabilizing SDSMC. A desired state signal is generated without taking the dynamics of the system into account. Then a SDSMC is designed to stabilize the closed-loop system at this piece-wise constant desired state signal. The design of such a controller was presented in Section 3.1. However, in this case, the closed-loop system is to be stabilized in the origin of the state-error-space, instead of the state-space. This can be accomplished by defining the switching function in terms of the state-error.

2. Tracking SDSMC. A desired state trajectory is obtained by (stable) system inversion which is then to be tracked by a tracking SDSMC (see Section 5.1).

3. Model Reference SDSMC. A reference model is constructed which describes the desired closed-loop dynamics. A model reference SDSMC is then designed

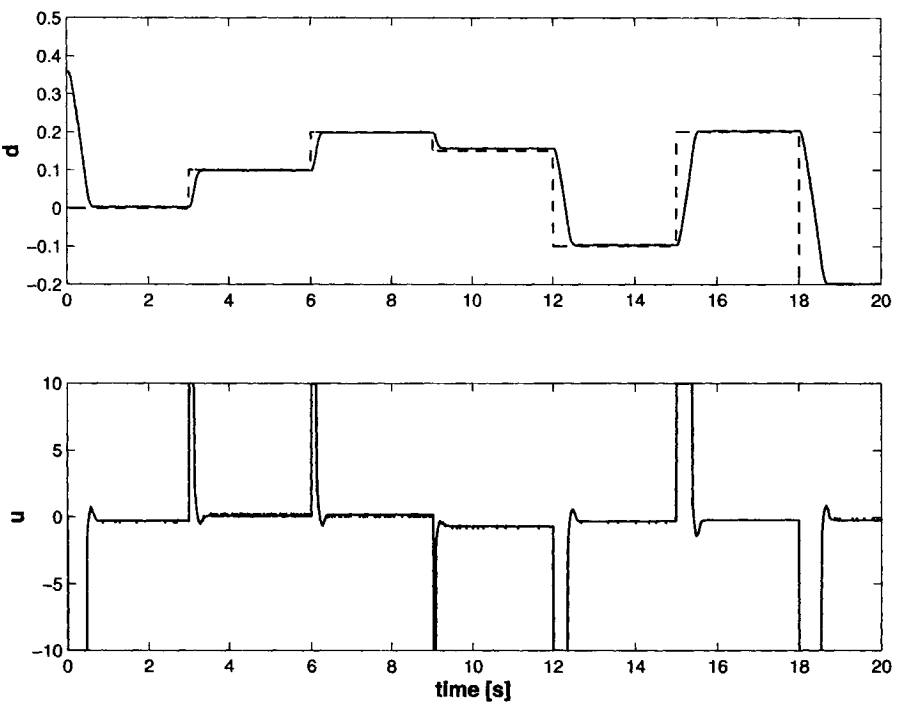


Fig. 6.3: Results for the DSMC applied to the cart test system. The top figure displays the desired- (dashed line) and the achieved (solid line) cart position d and the lower figure presents the control input u.

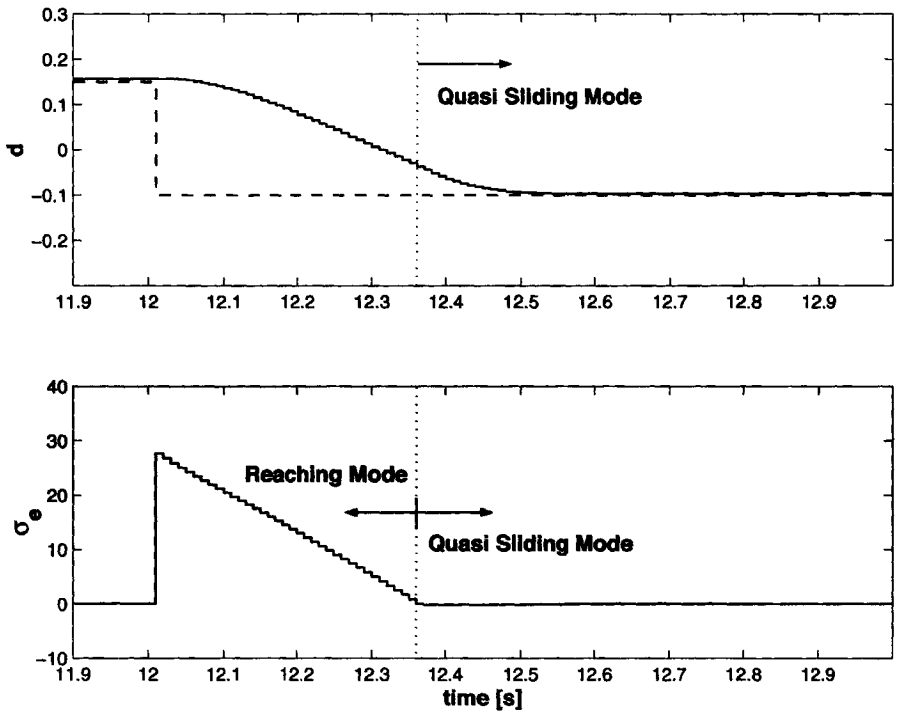


Fig. 6.4: Results for the DSMC applied to the cart test system. The top figure displays the desired- (dashed line) and the achieved (solid line) cart position d and the lower figure presents the switching function σ_{e_x}.

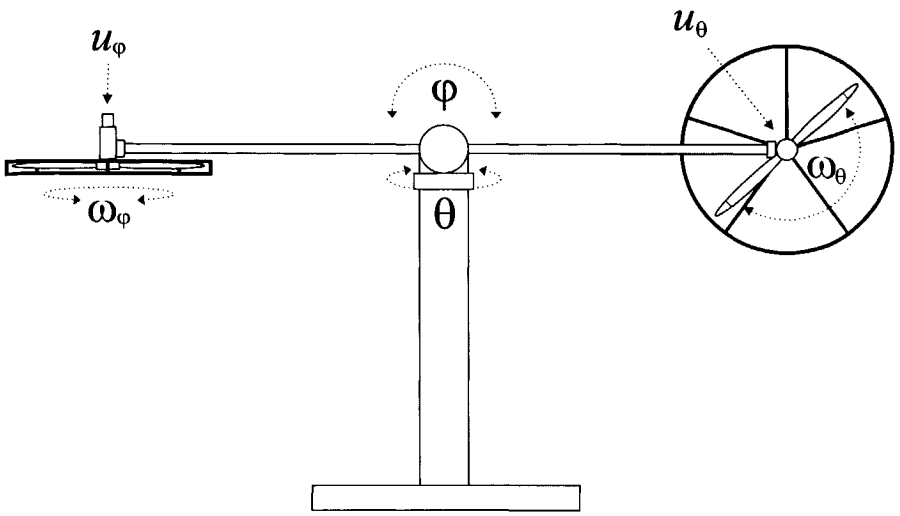


Fig. 6.5: A sketch of the test setup with the essential parameters of the system. The variables θ, ω_θ, and u_θ are for the single input case all fixed to zero.

to force the closed-loop system to track the defined reference model (see Section 5.2).

These three controller implementations are all based on a discrete-time model, leading to a discrete-time controller. In order to obtain a comparison between DSMC and CSMC, the model reference sliding mode controller is also designed using a continuous-time model, leading to:

4. Model Reference SCSMC. A SCSMC is used to force the closed-loop system to track a defined reference model. The sampling behavior of the controller is neglected in the design procedure, leading to a (supposedly) continuous-time closed-loop system. The results are compared with the model reference SDSMC controller for a relatively high- and low sampling frequency.

The first step in the design process is to obtain a model for the system. The DC-motor is modeled by a first-order linear model while the mechanical dynamics of the beam are modeled by a second order model. Consequently, the model can be represented by the block-scheme presented in Figure 6.6. The parameters of the system can be found to be $K_1 = 1.3$, $K_2 = 0.75$, $K_3 = 0.55$, $K_4 = 0.84$, and $T_1 = 0.3$. Using the approximation $\sin(\varphi) \approx \varphi$ the following state-space model can be constructed:

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{6.11}$$
$$y(t) = C_c x(t) + D_c u(t) \tag{6.12}$$

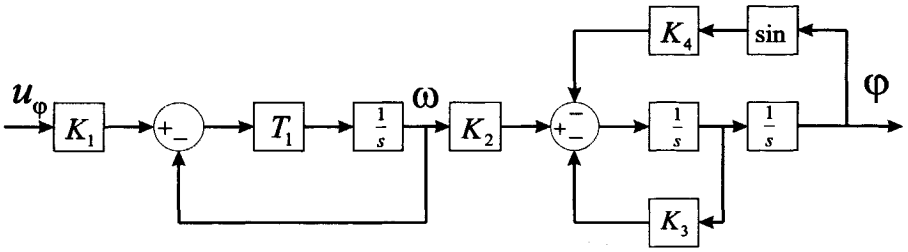


Fig. 6.6: Block-scheme of the system where only the pitch movement is considered.

where the control input u, the measured output y, and the system state x represent:

$$u(t) = u_\varphi(t) \qquad y(t) = \varphi(t) \qquad x(t) = \begin{bmatrix} \varphi(t) \\ \omega(t) \\ \dot{\varphi}(t) \end{bmatrix}$$

The quadruple (A_c, B_c, C_c, D_c) (the subscript c indicates the continuous-time domain) can be found to be:

$$A_c = \begin{bmatrix} 0 & 0 & 1.00 \\ 0 & -3.33 & 0 \\ -0.84 & 0.75 & -0.55 \end{bmatrix} \quad B_c = \begin{bmatrix} 0 \\ 4.33 \\ 0 \end{bmatrix}$$

$$C_c = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad D_c = 0$$

The closed-loop system is formed by the above continuous-time system and a discrete-time controller. All measured output signals and generated control signals are of the sample-and-hold type. Therefore, for the design of the SDSMCs, the continuous-time model is discretised by sample-and-hold, leading to:

$$x[k+1] = Ax[k] + Bu[k] \tag{6.13}$$

$$y[k] = Cx[k] + Du[k] \tag{6.14}$$

The matrices (A, B, C, D) can be easily obtained from (A_c, B_c, C_c, D_d). Unless otherwise specified, the sampling period is taken as $T_s = 10$ $[ms]$. The discrete-time model has its poles at $p_{ol} = \{0.9672, 0.9972 \pm 0.0087i\}$ and its zeros at $z_{ol} = \{-0.2654, -3.6961\}$.

Because only one of the three states is measured, the other two states (the pitch velocity $\dot{\varphi}$ and rotor speed ω_φ) have to be reconstructed. Therefore an observer is constructed as:

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] - L\left(C\hat{x}[k] - y[k]\right) \tag{6.15}$$

where $\hat{x}$ is the observed state and y the measured output. The observer gain L is taken to be:

$$L = \begin{bmatrix} 8.25e^{-2} \\ 0.19 \\ 0.18 \end{bmatrix}$$

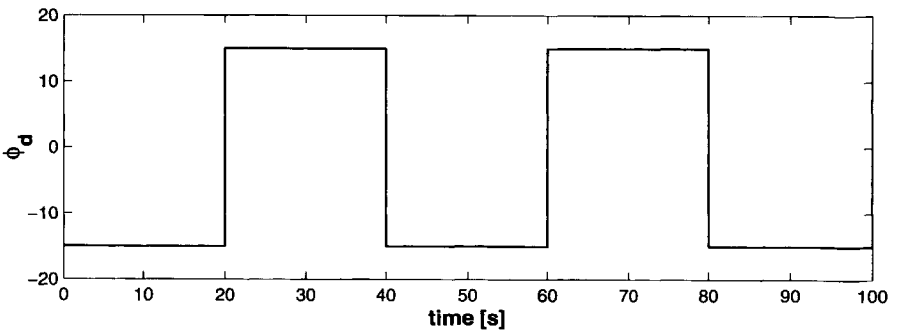


Fig. 6.7: Desired angle φ for the pitch controller.

placing the observer poles at $p_{ob} = \{0.9418, 0.9682 \pm 0.0232i\}$. For the case where we neglect the sampling issue and consider the closed-loop system to be fully continuous-time, an observer can be designed as:

$$\dot{\hat{x}}(t) = A_c\hat{x}(t) + B_c u(t) - L_c\left(C\hat{x}(t) - y(t)\right) \tag{6.16}$$

The observer gain L_c for the continuous-time observer is taken to be:

$$L_c = \begin{bmatrix} 8.52 \\ 20.54 \\ 18.66 \end{bmatrix}$$

placing the observer poles at $p_{ob} = \{-6.0000, -3.2000 \pm 2.4000i\}$.
The task for the controller is to stabilize the beam at certain pitch angles, as presented in Figure 6.7. As stated before, three solutions based on SDSMC are presented and tested. One of these (the model reference approach) is combined with a SCSMC as well to be compared with the discrete-time implementation. These issues will be discussed in the following sections. It was already concluded in Chapter 5 that the design of the sliding surface is identical for stabilizing controllers, tracking controllers, and model reference controllers. Therefore, it suffices to perform the design procedure for the sliding surface for all discrete-time controller implementations only once. For the design procedure we assume that the switching function is given by:

$$\sigma_x[k] = Sx[k] \tag{6.17}$$

The switching function can later be replaced by the correct expression (for example $\sigma_w[k] = S(x[k] - w[k])$ for the model reference controller) which is, however, irrelevant for the design procedure of the sliding surface. Applying the transformation $T_r \in \mathbb{R}^{3\times3}$, given by:

$$T_r = \begin{bmatrix} -1.00 & 2.68e^{-5} & -3.80e^{-3} \\ -3.80e^{-3} & -3.80e^{-3} & 1.00 \\ -1.26e^{-5} & -1.00 & -3.80e^{-3} \end{bmatrix}$$

to the system equation (6.13) and the switching function (6.17) results in the regular form:

$$\left[\begin{array}{c}\bar{x}_1[k+1]\\ \hline \bar{x}_2[k+1]\end{array}\right] = \left[\begin{array}{cc|c} 1.00 & -0.01 & 1.03e^{-4} \\ 8.40e^{-3} & 0.99 & -7.50e^{-3} \\ \hline -3.06e^{-5} & -1.03e^{-4} & 0.97 \end{array}\right] \left[\begin{array}{c}\bar{x}_1[k]\\ \hline \bar{x}_2[k]\end{array}\right] + \left[\begin{array}{c} 0 \\ 0 \\ \hline 0.04 \end{array}\right] u[k] \tag{6.18}$$

$$\sigma_x[k] = \bar{S}_1\bar{x}_1[k] + \bar{S}_2\bar{x}_2[k] \tag{6.19}$$

Assuming that the closed-loop system is driven in the sliding mode, i.e. $\sigma_x[k] = 0$, we can obtain the dynamics in the sliding mode using equations (6.18) and (6.19), resulting in:

$$\bar{x}_1[k+1] = \left[\begin{array}{cc} 1.00 & -0.01 \\ 8.40e^{-3} & 0.99 \end{array}\right] \bar{x}_1[k] - \left[\begin{array}{c} 1.03e^{-4} \\ 7.50e^{-3} \end{array}\right] \bar{S}_2^{-1}\bar{S}_1\bar{x}_1[k] \tag{6.20}$$

Obtaining the product $\bar{S}_2^{-1}\bar{S}_1$ is in fact a standard state-feedback problem of the reduced order system (6.20) and hence can be solved by for example LQR-design. Using the the weighting matrices $Q = I_2$ and $R = 0.05$ results in:

$$\bar{S}_2^{-1}\bar{S}_1 = [3.45 \quad -4.62]$$

The poles in the sliding mode are then given by $p_{sm} = \{0.9881, 0.9714\}$ which clearly are stable poles. The scalar $\bar{S}_2^{-1}$ only acts as a scaling factor on $\bar{S}_1$ and hence can be chosen arbitrary, we choose $\bar{S}_2 = \bar{B}_2^{-1}$. Then S can be found from:

$$S = \left[\bar{S}_1 \quad \bar{S}_2\right] T_r = [80.52 \quad 23.05 \quad 108.85] \tag{6.21}$$

We thus have constructed a sliding surface which results in a stable sliding motion. Since this design procedure is identical for the three discrete-time controller structures used for the tests, we can now focus our attention to the structure of the controllers and their performance.

6.2.1 Stabilizing SDSMC

The stabilizing SDSMC approach is the simplest and most straightforward of the three methods considered. The desired state trajectory is constructed as:

$$x_d[k] = \left[\begin{array}{c} G_{\omega,\varphi}^{-1}\varphi_d[k] \\ \varphi_d[k] \\ 0 \end{array}\right] \tag{6.22}$$

where $G_{\omega,\varphi}$ represents the DC-gain from ω to φ. Obviously, the above desired state trajectory only specifies the desired steady-state values for the state. The (dynamic) transition from one "steady-state" to another is not considered. Consequently, the

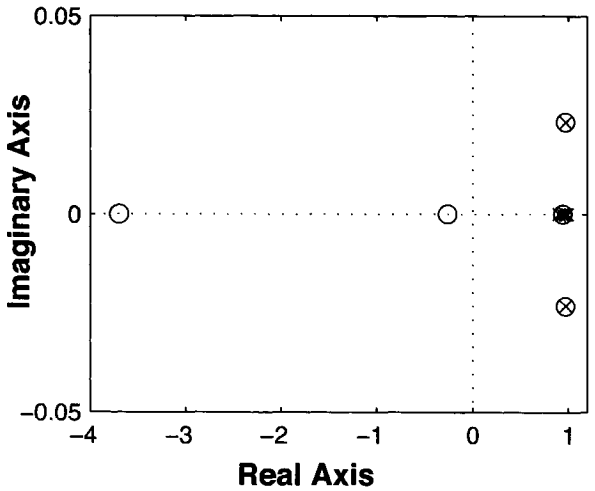


Fig. 6.8: The pole-zero map for the stabilizing SDSMC.

desired state (6.22) is not tractable and thus the closed-loop system will not remain in the sliding mode during a transition.
To come to a stabilizing SDSMC which stabilizes the closed-loop system at the piece-wise constant desired state signal represented by equation (6.22), the switching function (6.17) is altered to:

$$\sigma_{e_x}[k] = S(x[k] - x_d[k]) = Se_x[k] \tag{6.23}$$

The closed-loop system should reach the sliding surface according to the linear reaching law:

$$\sigma_{e_x}[k+1] = \Phi\sigma_{e_x}[k] \tag{6.24}$$

Reaching law (6.24) leads to the control law:

$$u[k] = (SB)^{-1}(\Phi\sigma_{e_x}[k] - SAe_x[k]) \tag{6.25}$$

The parameter Φ is selected as $\Phi = 0.9$. The pole-zero map for the closed-loop system is presented in Figure 6.8. The observer poles can be clearly identified in the pole-zero map because of the zeros at the same locations. The two negative zeros ($z_{cl} = \{-3.6961, -0.2654\}$) are the open-loop zeros. The poles in the sliding mode were assigned at $p_{sm} = \{0.9882, 0.9713\}$, these poles appear in the pole-zero map at about the same locations. The final pole in the pole-zero map lies at $z = 0.9085$ which roughly coincides with the pole for the reaching law $\sigma_{e_x}[k+1] = \Phi\sigma_{e_x}[k]$ which was placed at $\Phi = 0.9$.
The test results are depicted in Figure 6.9. The plot for the pitch angle φ displays a

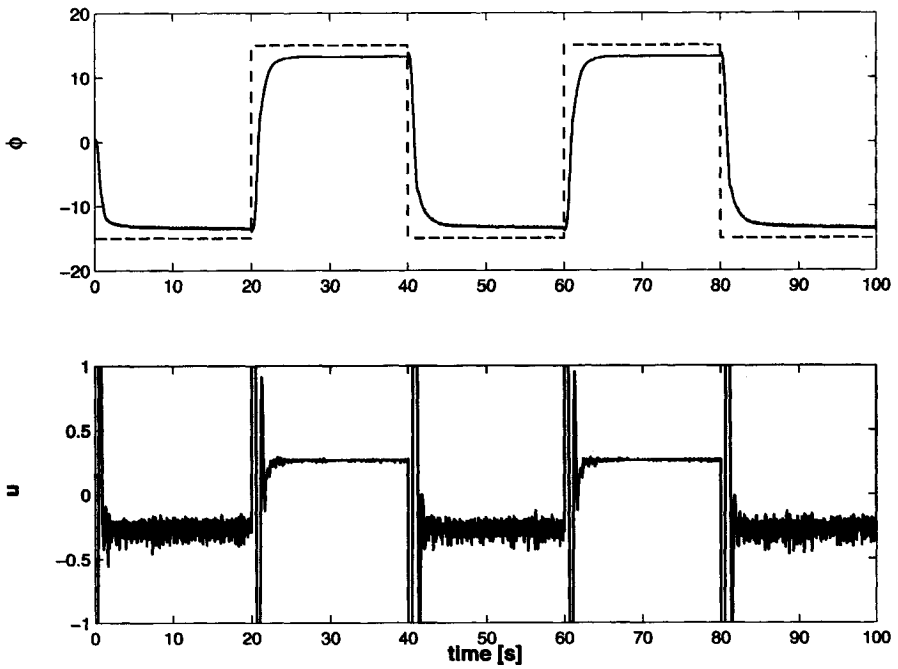


Fig. 6.9: Test results for the stabilization controller. The top plot represents the (piecewise) constant desired angle (dashed) and the achieved angle (solid) while the lower plot represents the applied input signal.

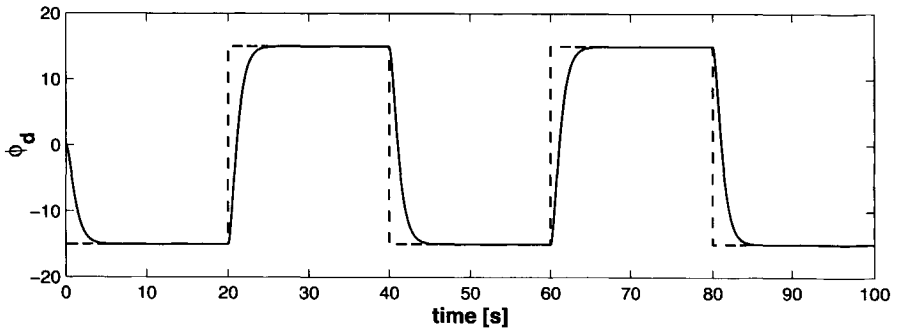


Fig. 6.10: Filtered desired output signal.

rapid response. However, the steady-state error is quite large. The noisy behaviour of the input signal for negative values is due to the sensor noise which is substantially larger for negative angles for our test setup.

6.2.2 Tracking SDSMC

In the previous section we have considered the problem of stabilizing the beam at certain (piece-wise) constant angles as a stabilization problem. Another approach is interpret the problem as a tracking problem, where some specified desired output trajectory needs to be replicated by the closed-loop system. By (stable) system inversion a desired state trajectory can be computed which is to be tracked by the closed-loop system.

The zeros of the open-loop system are located at $z = \{-0.2654, -3.6961\}$. Consequently, the system has one nonminimum phase zero leading to an unstable inverse system. Therefore, the system inversion, and consequently the desired state trajectory generation, should be performed by a stable inversion technique as was described in Chapter 4.

Another problem is of course the step-wise change of the desired output signal. Computing an input signal leading to perfect tracking of some step-wise changing desired output signal leads to excessively large control inputs. Therefore, we filter the piece-wise constant desired output signal resulting in the signal presented in Figure 6.10. The filtered desired output signal is then used to generate the desired state trajectory. We have used the SDI technique, introduced in Section 4.2, to generate the desired state trajectory since for our test case the desired output signal was fully known in advance.

Similar to the stabilizing SDSMC, the sliding surface for the tracking controller is defined as:

$$\sigma_{e_x}[k] = S\left(x[k] - x_d[k]\right) \tag{6.26}$$

However, in this case the desired state trajectory x_d is computed via system inversion for the filtered reference signal. Another important difference compared to the stabilizing controller is that a tracking controller is not only based on the cur-

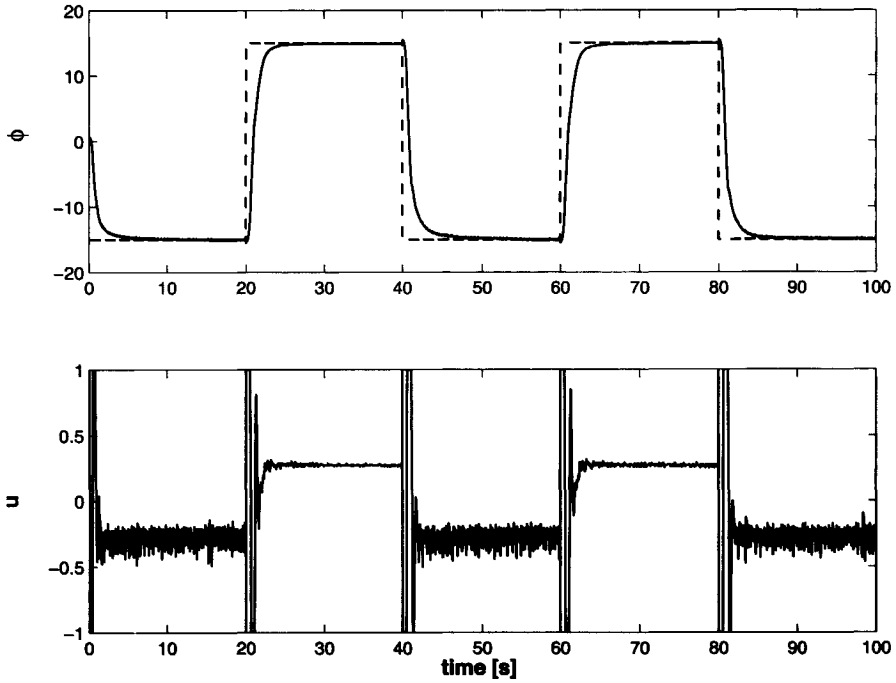


Fig. 6.11: Test results for the tracking controller. The top plot represents the (piecewise) constant desired angle (dashed) and the achieved angle (solid) while the lower plot represents the applied input signal.

rent desired state $x_d[k]$ but also on the desired state at the next sampling instant $x_d[k+1]$, as can be seen in the control law:

$$u[k] = (SB)^{-1} \left(\Phi\sigma_{e_x}[k] + Sx_d[k+1] - SAx[k]\right) \tag{6.27}$$

Hence, the tracking controller does not aim at stabilizing the closed-loop system at some desired position. Instead, it tracks a specified dynamic desired state trajectory thereby using a one-step-ahead prediction.

As stated before, the sliding surface is identical to the sliding surface of the stabilizing controller ($S = [80.52 \;\; 23.05 \;\; 108.85]$). Also the reaching variable Φ is taken exactly the same ($\Phi = 0.9$).

The test results are presented in Figure 6.11. The improvement of the performance of the tracking controller, compared with the stabilizing controller, is remarkable. The response of the closed-loop system during transitions is very fast again. However, the steady-state error had diminished to a large extend. The input signal displays the same behaviour as before, for negative values of the pitch angle the input becomes rather noisy.

6.2.3 Model Reference SDSMC

A model reference SDSMC was presented in Section 5.2 as an alternative for the tracking SDSMC. A reference model is defined to describe the desired closed-loop system behaviour. This reference model is defined by:

$$w[k+1] = Gw[k] + H\varphi_d[k]) \tag{6.28}$$

where $G \in \mathbb{R}^{3\times3}$ and $H \in \mathbb{R}^3$ can be represented by:

$$G = (A - B\delta_G) \tag{6.29}$$

$$H = B\delta_H \tag{6.30}$$

with $\delta_G \in \mathbb{R}^{1\times3}$ and $\delta_H \in \mathbb{R}$. The inclusion of the input matrix B in the design of the matrices (G, H) (see equations (6.29) and (6.30)) ensures tractability of the reference state w as defined in Theorem 5.2.1, Section 5.2. The design of δ_G corresponds with the standard state-feedback problem $(A-BK)$ and hence standard techniques, such as LQR design or pole-placement, can be employed. The scalar δ_H should be chosen such that the DC-gain of the system, formed by equation (6.28) and:

$$\varphi_{ref}[k] = Cw[k], \tag{6.31}$$

is equal to one. This condition ensures that the steady-state error of the (nominal) closed-loop system equals zero if the reference model is tracked. The design parameters δ_G and δ_H are chosen as $\delta_G = [0\ 0\ 1]$ and $\delta_H = 0.8589$, leading to the matrices G and H:

$$G = \begin{bmatrix} 1.00 & 3.70e^{-5} & 1.00e^{-2} \\ 0 & 0.97 & -4.26e^{-2} \\ -8.40e^{-2} & 7.40e^{-2} & 0.99 \end{bmatrix} \qquad H = \begin{bmatrix} 4.60e^{-7} \\ 3.66e^{-2} \\ 1.38e^{-4} \end{bmatrix}$$

The above choice for (G, H) ensures a smooth transition between each constant angle, without any overshoot as can be seen in Figure 6.12. Figure 6.12 shows a close similarity with Figure 6.10, where the reference signal was simply filtered. However, in the case of model reference control the desired closed-loop dynamics are specified whereas for the tracking control the desired output signal is specified. The switching function is now defined as:

$$\sigma_{xw}[k] = S(x[k] - w[k]) \tag{6.32}$$

Again employing the linear reaching law $\sigma_{xw}[k+1] = \Phi\sigma_{xw}[k]$ leads to the control law:

$$u[k] = (SB)^{-1}\left(S(G-A)w[k] + SHr[k] + \Phi\sigma_{xw}[k] - SAe_{xw}[k]\right) \tag{6.33}$$

The parameters S and Φ were again taken the same ($S = [80.55\ \ 23.05\ \ 108.85]$ and $\Phi = 0.9$).

The test results for the model reference SDSMC are presented in Figure 6.13. The performance of the model reference SDSMC is comparable with the tracking

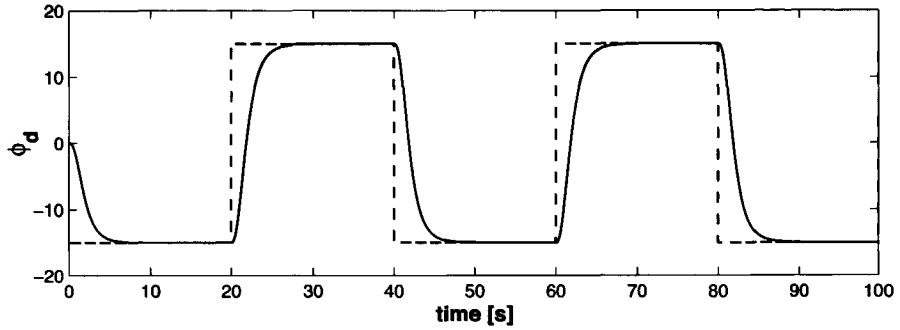


Fig. 6.12: Desired pitch angle filtered by the reference model.

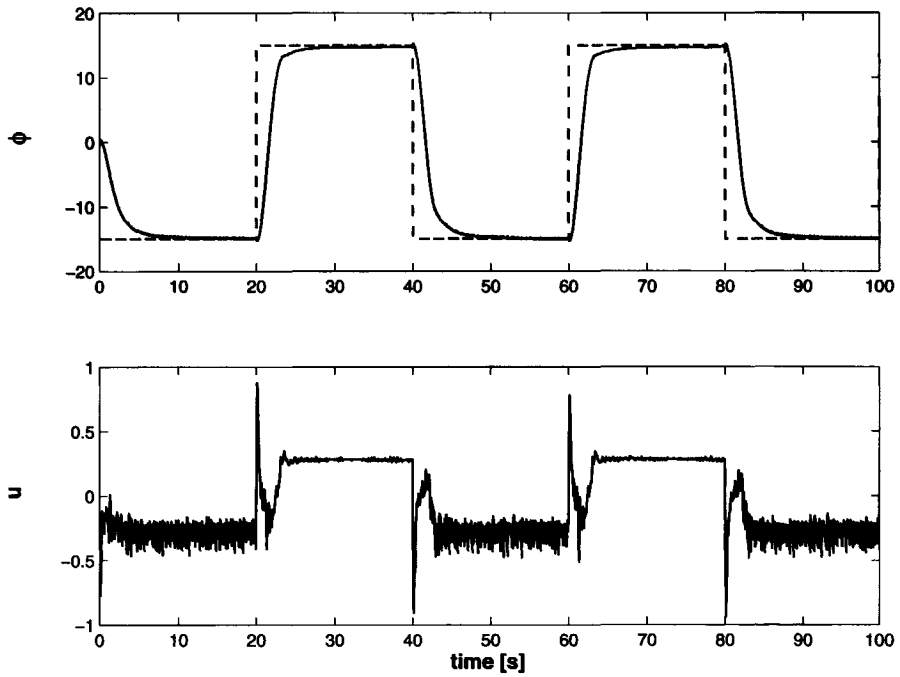


Fig. 6.13: Test results for the model reference controller. The top plot represents the (piece-wise) constant desired angle (dashed) and the achieved angle (solid) while the lower plot represents the applied input signal.

SDSMC. The main difference between the two controllers is the fact that the input signal of the tracking SDSMC is more aggressive during transitions. Apparently, the specified reference model generates a desired state trajectory which is easier to track by the closed-loop system.

6.2.4 Model Reference SCSMC

In this section we define a model reference SCSMC to compare the performance of a SCSMC with a SDSMC in a practical testing environment for a relatively fast- and slow sampling frequency. The continuous-time reference model is given by:

$$\dot{w}(t) = G_c w(t) + H_c r(t) \tag{6.34}$$

The matrices (G_c, H_c) are the continuous-time equivalents of (G, H), i.e. the step response of the continuous-time reference model maps the discrete-time reference model exactly. Also the dynamics in the sliding mode are equivalent to those of the discrete-time sliding mode dynamics. The matrices (G_c, H_c, S_c) are given by:

$$G_c = \begin{bmatrix} 0 & 0 & 1.00 \\ 0 & -3.33 & 0.00 \\ -0.84 & 0.75 & -1.81 \end{bmatrix} \quad H_c = \begin{bmatrix} 0 \\ 3.72 \\ 0 \end{bmatrix}$$

$$S_c = \begin{bmatrix} 3.02 & 0.23 & 3.37 \end{bmatrix}$$

The continuous-time observer, used to reconstruct the systems' state, is designed such that it exhibits the same dynamics as the discrete-time observer.
The control law can be easily established starting from the switching function:

$$\sigma_c(t) = S_c\,(x(t) - w(t)) \tag{6.35}$$

In Chapter 2 it was shown that the following reaching law guarantees convergence towards the sliding surface:

$$\dot{\sigma}_c(t) = -K_p \sigma_c(t) - K_s \text{sign}(\sigma_c(t)) \tag{6.36}$$

if the gains K_p and K_s have been chosen sufficiently large. Using equations (6.11), (6.34), (6.35), and (6.36), we can obtain the following control law:

$$u(t) = (SB)^{-1} \left(SGw(t) + SHr(t) - SAx(t) - K_p \sigma_c(t) - K_s \text{Sat}(\frac{\sigma_c(t)}{\delta}) \right) \tag{6.37}$$

The sign-function present in reaching law (6.36) has been replaced by the saturation-function to reduce chattering, as presented in Section 2.3.1. A more extensive treatment of continuous-time model reference control using sliding modes can be found in for example (Edwards and Spurgeon 1998).
In Figure 6.14 the performance of both the continuous-time and the discrete-time model reference SMCs are presented for a sampling time of $T_s = 10\ ms$. The parameters are in this case given by $K_p = 5$ and $K_s = 0.1$. The figure demonstrates

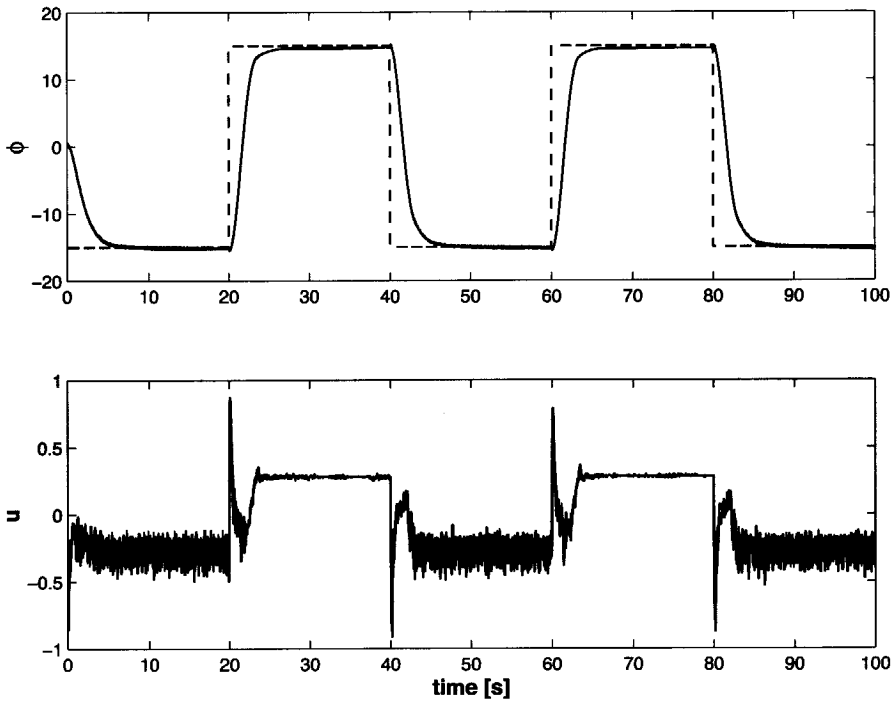


Fig. 6.14: Test results for the model reference SDSMC and the model reference SCSMC for a sampling time of $T_s = 10\ ms$. The top plot represents the (piece-wise) constant desired angle (dashed) and the achieved angle (solid) for both controllers while the lower plot represents the applied input signal for both controllers.

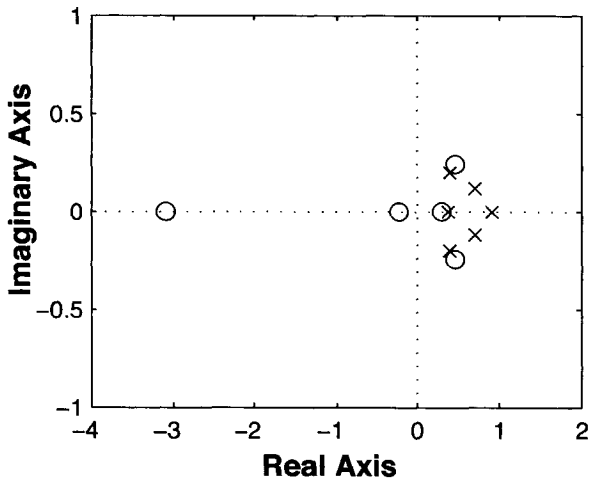


Fig. 6.15: The pole-zero map for the stabilizing SDSMC with a sample time of $T_s = 0.2\ s$.

that the performance of both controllers are comparable. The achieved pitch angles for the continuous-time and the discrete-time controller perfectly match each other.

To run the same test for a sampling time of $T_s = 0.2\ s$ we have to redesign the controllers. For the model reference SDSMC we can use exactly the same design procedure thereby using the same parameters for the LQR-design ($Q = I_2$ and $R = 0.05$), leading to $S = [3.94\ \ 1.15\ \ 5.13]$. The poles in the sliding mode are now given by $p_{sm} = \{0.7656, 0.6006\}$. Again choosing $\Phi = 0.9$ leads to the pole-zero map as shown in Figure 6.15. The observer poles can again easily be identified by their associated zeros, also the open-loop zeros again appear at the left-hand side. The remaining poles are at $z = \{0.9091\}$ which can be associated with Φ and $p = \{0.6752 \pm 0.0719i\}$ which could be associated with the poles in the sliding mode.

Because for the design of the model reference SCSMC the closed-loop system is considered to be continuous-time, the design for the model reference SCSMC can be applied unchanged. However, since this instantly leads to an unstable closed-loop system (as testing points out), we have to alter the controller design anyway. We have chosen to use the discrete-time observer for the model reference SCSMC. This leads to a stable closed-loop system and still provides us with a comparison between the model reference- SCSMC and SDSMC. The results are shown in Figure 6.16 for the sampling time of $T_s = 0.2\ s$. The parameters for the model reference SCSMC have been tuned to $K_p = 0.7$ and $K_s = 0.1$ to obtain a better performance. Φ for the model reference SDSMC has been changed to $\Phi = 0.6$ for the same purpose. The model reference SDSMC shows a decreased performance compared with the shorter sampling time, especially the steady-state error has increased. However, the model reference SCSMC exhibits a severe oscillation in the achieved pitch angle, despite

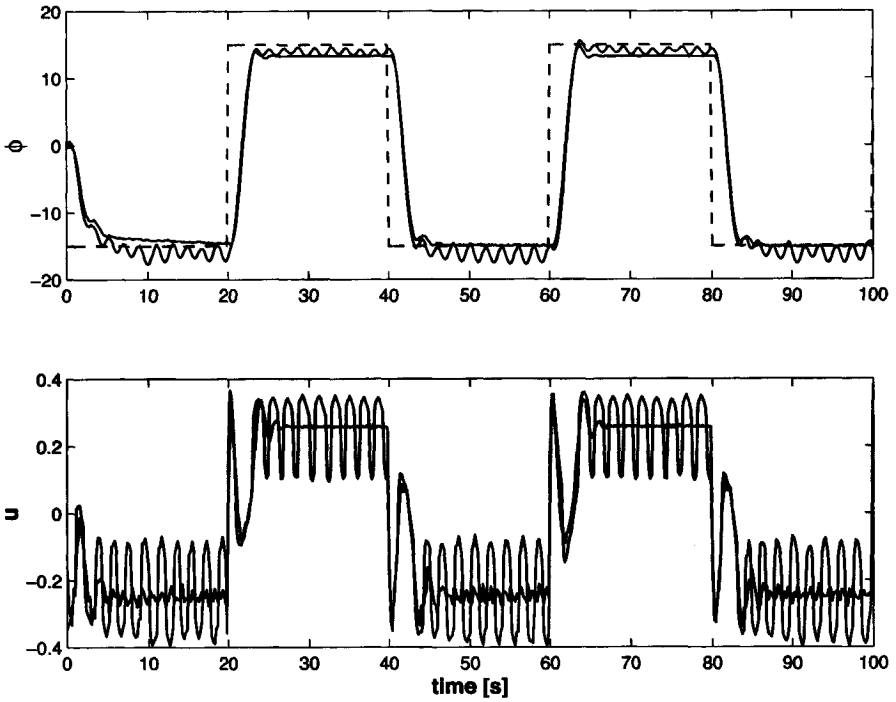


Fig. 6.16: Test results for the model reference SDSMC and the model reference SCSMC for a sampling time of $T_s = 0.2\ s$. The top plot represents the (piece-wise) constant desired angle (dashed) and the achieved angle (solid) for both controllers while the lower plot represents the applied input signal for both controllers.

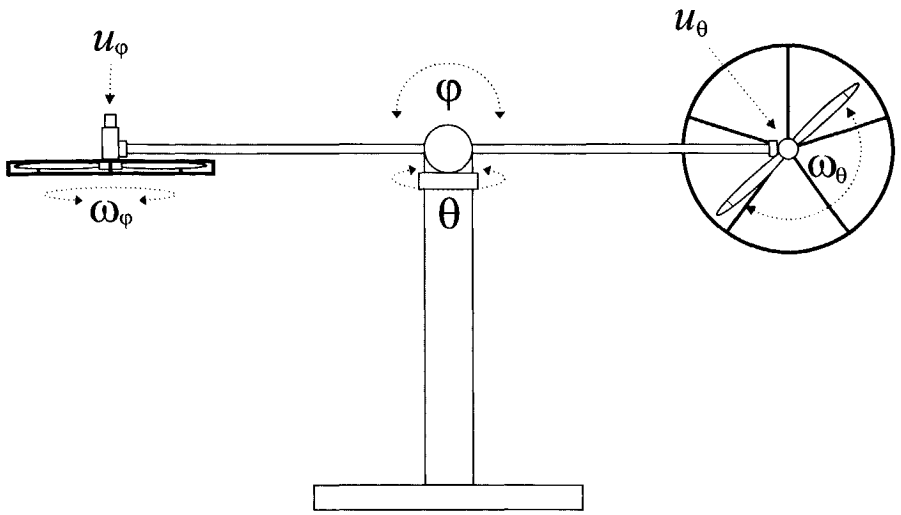


Fig. 6.17: A sketch of the test setup with the essential parameters of the system.

the fact that the proportional gain K_p was decreased substantially. The oscillatory behaviour of the closed-loop system is even clearer in the applied input signal, shown in Figure 6.16 as well. Obviously, the model reference SDSMC is preferable for low sampling frequencies. A fact which could be expected since the error introduced by neglecting the sampling issue becomes larger for lower sampling frequencies. Another important result of this section is the fact that for the relatively high sampling frequency (100 Hz) the performance of the SCSMC and the SDSMC is equal.

6.3 Pitch & Yaw Control

In this section we will control the test setup, introduced in the previous section, in both pitch and the yaw direction. The sketch of the setup is shown again in Figure 6.17. The control problem at hand is in this case multiple input, the yaw- and the pitch-direction can be controlled independently by two propellers. However, as will be demonstrated, there is a severe coupling between the pitch-propeller and the yaw direction, and visa versa. Furthermore, similar to the case where only the pitch direction was controlled, the system exhibits severe stiction.
Similar to the case where only the pitch direction was controlled (previous section), three approaches are presented:

- Stabilizing SDSMC.
- Tracking SDSMC.
- Model reference SDSMC.

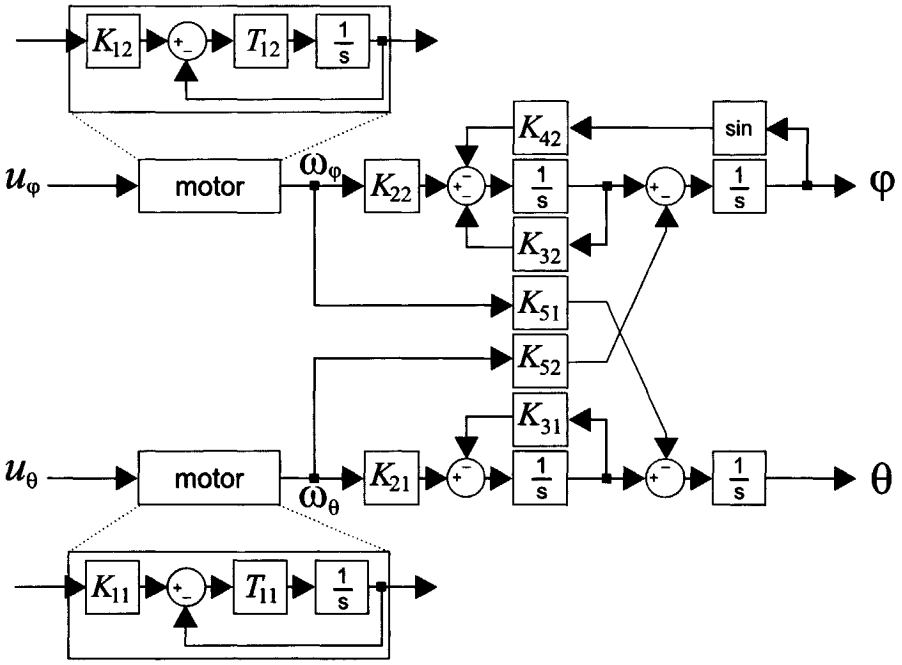


Fig. 6.18: The block diagram of the helicopter setup.

In all cases the controller is state-based, enforcing the use of a state observer to recover the unmeasured states. Before going to the design of the controllers, a model is obtained for this test setup.

The actuators of the system (DC motors) are modeled as first-order linear models with time constants T_{1j} ($j = 1$ represents the yaw actuator and $j = 2$ the pitch actuator) and gains K_{1j}. The beam is modeled as two coupled second order models. The total continuous-time model can be represented by the block-diagram presented in Figure 6.18. The values for the parameters of the model can be found in in Table 6.1. Linearizing in the origin of the state-space and discretizing by sample and hold

T_{11}	0.30	T_{12}	0.30
K_{11}	0.65	K_{12}	1.30
K_{21}	-0.23	K_{22}	0.75
K_{31}	0.50	K_{32}	0.55
K_{41}	0	K_{42}	0.84
K_{51}	-0.05	K_{52}	0.25

Table 6.1: The parameters of the helicopter model.

(sampling time $T_s = 0.01\ s$) leads us to the following sampled data model:

$$x[k+1] = Ax[k] + Bu[k] \tag{6.38}$$
$$y[k] = Cx[k] + Du[k] \tag{6.39}$$

where

$$\begin{aligned} x[k] &= [\theta[k] \ \ \varphi[k] \ \ \omega_\theta[k] \ \ \omega_\varphi[k] \ \ \dot{\theta}[k] \ \ \dot{\varphi}[k]]^T \\ y[k] &= [\theta[k] \ \ \varphi[k]]^T \\ u[k] &= [u_\theta[k] \ \ u_\varphi[k]]^T \end{aligned}$$

The matrices (A, B, C, D) are given by:

$$A = \begin{bmatrix} 1.00 & 0 & -1.11e^{-5} & 4.926e^{-4} & 0.01 & 0 \\ 0 & 1.00 & -2.50e^{-3} & 3.70e^{-5} & 0 & 0.01 \\ 0 & 0 & 0.97 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.97 & 0 & 0 \\ 0 & 0 & -2.20e^{-3} & 0 & 1.00 & 0 \\ 0 & -8.40e^{-3} & 1.03e^{-5} & 7.40e^{-3} & 0 & 0.99 \end{bmatrix}$$

$$B = \begin{bmatrix} -8.05e^{-8} & 1.07e^{-5} \\ -2.68e^{-5} & 5.36e^{-7} \\ 0.02 & 0 \\ 0 & 0.04 \\ -2.41e^{-5} & 0 \\ 7.49e^{-8} & 1.60e^{-4} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

To estimate the system's states, we implement the observer:

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] - L(y[k] - \hat{y}[k]) \tag{6.40}$$

where the observer gain, L has been chosen as:

$$L = \begin{bmatrix} 0.30 & 7.05e^{-5} \\ 1.61e^{-5} & 0.30 \\ -51.64 & -9.93 \\ 47.92 & 15.80 \\ 0.14 & -0.79 \\ -12.91 & 0.05 \end{bmatrix}$$

leading to the observer poles $p_{obs} = \{0.8964 \pm 0.0469i, 0.9704, 0.9418, 0.9471 \pm 0.0248i\}$. The sliding surface is again determined by LQR design with $Q = I_4$ and $R = 1e^{-2}I_2$,

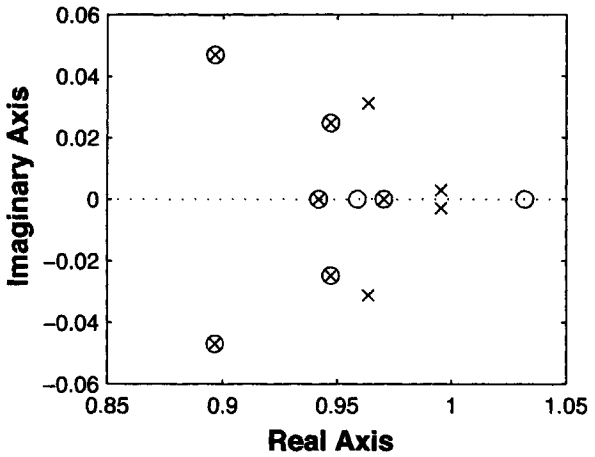


Fig. 6.19: Closed-loop poles and zeros of the closed-loop system.

leading to:

$$S = \begin{bmatrix} -64.31 & -59.07 & 46.75 & 0.03 & -92.32 & -3.43 \\ -54.71 & 661.24 & 0.73 & 22.72 & -91.57 & 198.87 \end{bmatrix}$$

The poles in the sliding mode are given by $p_{sm} = \{0.9636 \pm 0.0312i, 0.9953 \pm 0.0029i\}$. The variables Φ is selected as $\Phi = 0.7I_2$. In Figure 6.19 the pole-zero map in the vicinity of the point $z = 1$ is displayed for the closed-loop system. The closed-loop system has two more zeros at $z = -0.9769$ and $z = -1.0011$, which are invariant zeros of the open-loop system. The pole-zero cancellations present in Figure 6.19 are due to the observer. The other zeros in Figure 6.19 are invariant zeros of the open-loop system. The four plotted complex poles which are not cancelled by zeros are the poles which describe the sliding motion. The pole at $z = 0.9$ is in fact a double pole assigned by the reaching law variable Φ.
In Figure 6.20 the test result for the stabilizing controller is presented, Figure 6.21 presents the test results for the tracking SDSMC, and finally Figure 6.22 presents the test results for the model reference SDSMC. The controllers have been designed in the same way as the single input controllers presented in the previous section. The plots demonstrate that the performance of the tracking SDSMC and the model reference SDSMC are comparable while the performance of the stabilizing SDSMC is much worse. All controller schemes exhibit a steady-state error caused by the stiction in the system.

6.4 Summary

The main purpose of this chapter is to demonstrate the applicability of the control techniques introduced in this thesis, notably in Chapters 3, 4, and 5. Several simu-

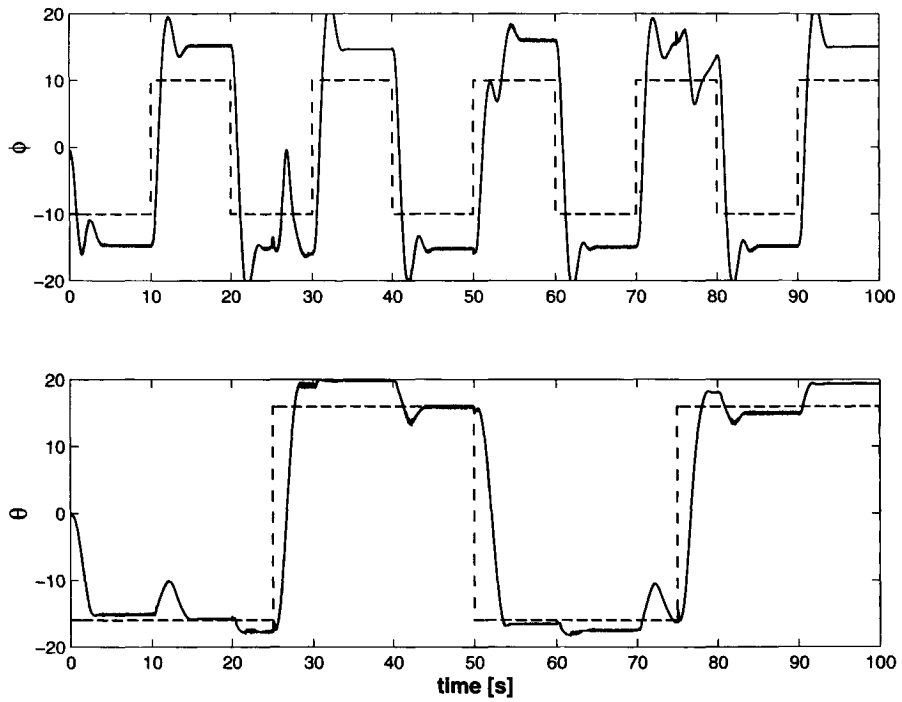


Fig. 6.20: Test results for the stabilizing SDSMC. The dashed line represents the desired output trajectory while the solid line represents the achieved output.

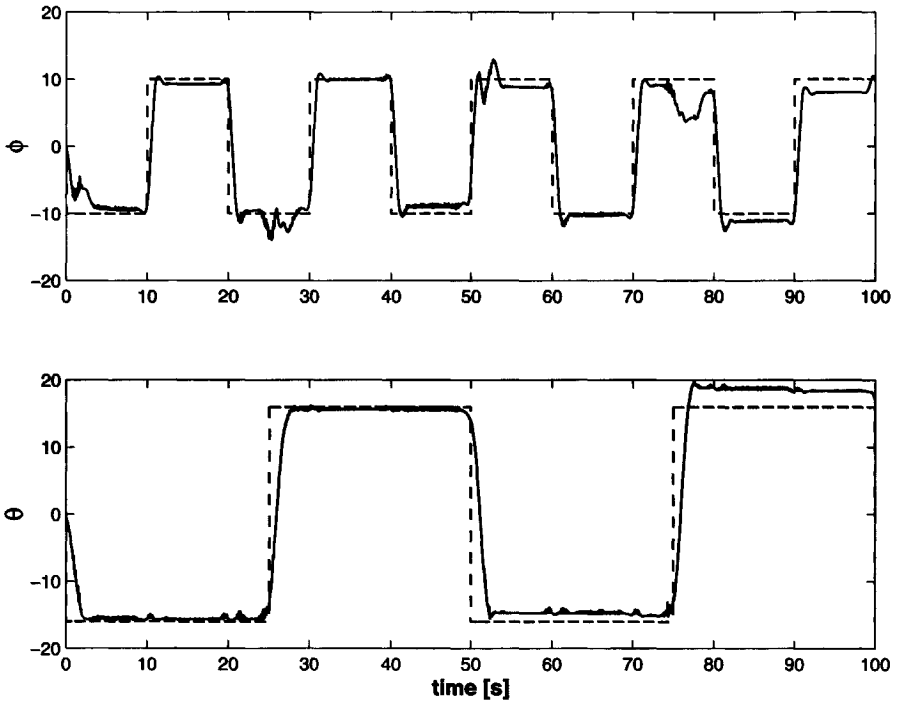


Fig. 6.21: Test results for the tracking SDSMC. The dashed line represents the desired output trajectory while the solid line represents the achieved output.

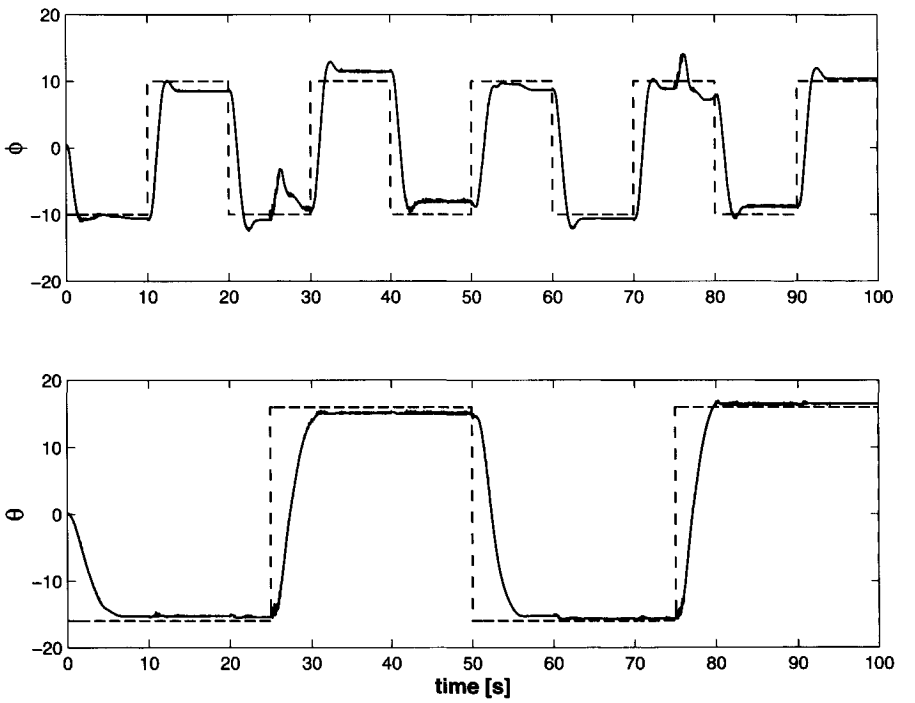
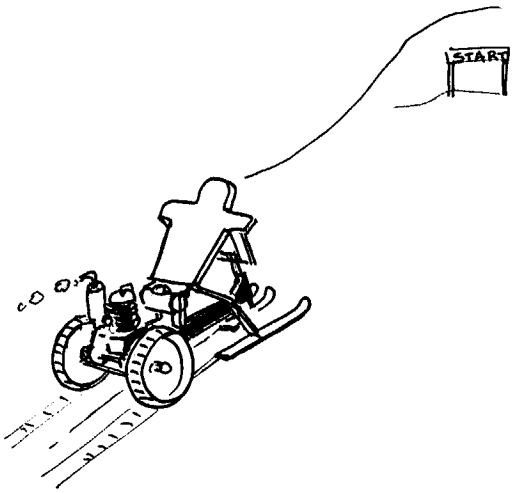


Fig. 6.22: Test results for the stabilizing controller. The dashed line represents the desired output trajectory while the solid line represents the achieved output.

lation examples were already included in those chapters. In this chapter the control theory is applied to three laboratory test setups. First, the position of a cart fixed to a rail is controlled with a SDSMC. Then a beam balancing on a fixed pole is positioned at specified pitch angles using a stabilizing SDSMC, a tracking SDSMC, and a model reference SDSMC. It follows from the tests that the tracking SDSMC and the model reference SDSMC have similar performances while better then the stabilizing SDSMC. The model reference controller is also implemented in continuous-time using SCSMC thereby neglecting the sampling issue. For a relatively high sampling frequency (100 Hz) the performance of the SDSMC and the SCSMC are comparable. However, for a relatively low sampling frequency (5 Hz) the SDSMC performs much better then the SCSMC. Finally, the stabilizing SDSMC, tracking SDSMC, and the model reference SDSMC are used to control the beam in both the pitch and yaw direction. The same can be concluded again, the performance of the tracking SDSMC and the model reference SDSMC are comparable and better then the stabilizing SDSMC.
Two publications using the results in this chapter are in preparation.



7

Automotive Engine Control

The first person to experiment with an internal-combustion engine was the Dutch physicist Christian Huygens, about 1680. It took however until 1888 before Nikolaus A. Otto built a successful four-stroke engine. In the years thereafter, car engines were continuously improved to meet the ever growing demands of drivers and manufacturers. After years of improving the engine-mechanics, a big step in engine efficiency came by the introduction of electronic injection systems. For a wide range of operating conditions the desired injected quantity of fuel was stored in look-up tables. While these methods gave good results for steady-state operation, their performance degraded significantly during fast transitions. The performance degradation included an increase in air pollution and a decrease in drivability. The next step of improvement of engine efficiency came with the introduction of microcomputers in the 1980's. More advanced control strategies could be applied due to the ever increasing computational power of the computers.

An important stimulus for the introduction of engine control is the introduction of cruise control in commercial cars. In this case, the control of the throttle position is taken over from the driver by the microcomputer. Another important reason for the use of microcomputers in automotive applications is the ever increasing environmental demands on car engines. Research has demonstrated (Canale and et. al. 1978),(Falk and Mooney 1980) that the efficiency of catalytic converters is maximized if the ratio between air and fuel in the cylinder is exactly 14.7. Even slight deviations of a few percent from this ratio, called the stoichiometric ratio, result in a drastic degradation of the catalytic conversion efficiency. Variations of greater than 1% above stoichiometry results in a dramatic increase in CO and HC emission levels, while a decrease of more than 1% involves NO_x emissions increasing by up to 50%. For this reason we seek a controller setup which maintains the air fuel ratio well within a $\pm$ 1 % boundary around the stoichiometric air fuel ratio, while

steering the car towards the desired speed. Ultimately the air fuel ratio should be kept within 0.1 % from stoichiometry.
The control task at hand can be divided in two parts:

- Control the throttle position such that the engine speed, and consequently the car speed, follow a prescribed trajectory.

- Control the injected fuel flow such that the air fuel ratio is maintained at the stoichiometric value of 14.7.

In the control literature example publications on engine control are (Cho and Hedrick 1988), (Moskwa and Hedrick 1990), (Cho and Hedrick 1991), (Cho 1991), (Majors et al. 1994), (Vesterholm and Hendricks 1994), (Shiraishi et al. 1995), (Choi and Hedrick 1996), (Shim et al. 1996), (Kim et al. 1998), (Powell et al. 1998), (Won et al. 1998), (Pieper and Mehrotra 1999), (Balluchi, Benedetto, Pinello, Rossi and Sangiovanni-Vincentenlli 1999), (Balluchi, Benvenuti, Benedetto, Cardellino, Rossi and Sangiovanni-Vincentenlli 1999), (Stotsky et al. 2000), (Won and Hedrick 2001). Sliding mode control became one of the most popular control techniques in the field of air fuel ratio control and speed control. A position earned by its robustness properties and capabilities of handling nonlinear models.
In this chapter a design procedure for an air fuel ratio controller and an engine speed controller is presented based on the mean value model presented in Section 7.1. Effects as wall wetting and measurement delays are incorporated in a nonlinear model. However, the engine is modeled as a continuous-time system without taking into account the separate phases of a four-stroke engine (intake, compression, combustion and exhaust).
Both the air fuel ratio controller and the engine speed controller, although completely different in their structure, are based on the sliding mode control methodology. The air fuel controller is constructed in a feedforward/feedback configuration to reduce the controller sensitivity to the measurement delay of the air fuel ratio (Section 7.2.1). The engine speed controller is based on the design procedure developed in (Lu and Spurgeon 1997), (Lu and Spurgeon 1998), (Lu and Spurgeon 1999*a*), (Lu and Spurgeon 1999*b*). This design procedure is applicable to model in the so-called Generalized Controller Canonical Form (Section 7.2.2). The derived controllers are applied to an extensive industrial simulation model in Section 7.3.

7.1 Engine Model

A spark ignition engine is a highly complicated system with an extremely nonlinear behavior. In this section a simplified model is presented which can be found in the literature. The engine is modeled as a continuous-time system without taking into account the separate phases of a four-stroke engine (intake, compression, combustion and exhaust). For this reason, the obtained engine model is known as a mean value model.
The air fuel ratio is normally defined as the mass ratio of air and fuel in the cylinder

during compression, i.e.:

$$\lambda[k] = \frac{m_{ac}[k]}{m_{fc}[k]} \tag{7.1}$$

where the variable $k \in \mathbb{N}$ indicates the k^{th} compression cycle. Clearly the above variable is discrete-time, it is only defined during compression. However, since the model does not take the separate phases of a four-stroke engine into account, we approximate the air fuel ratio by (Cho and Hedrick 1988):

$$\lambda(t) = \frac{\dot{m}_{ac}(t)}{\dot{m}_{fc}(t)} \tag{7.2}$$

As opposed to the earlier discrete-time definition of the air fuel ratio, the above definition models the air fuel ratio as a continuous-time signal, assuming a continuous air- and fuel-flow into the cylinder.
A simple sketch of an engine is presented in Figure 7.1. If we have a closer look at the engine we can identify four more or less independent parts:

- The *fueling system*, including the injector, is introduced in Section 7.1.1.
- The *air flow system*, formed by the throttle and manifold, is presented in Section 7.1.2.
- The *engine speed dynamics system* is described in Section 7.1.3.
- The *sensor system* is described in Section 7.1.4.

The purpose of these systems can be easily explained. The air flow system provides an air flow (controlled by the throttle position) into the cylinder whereas the fueling system provides the fuel flow (controlled by the injector) into the cylinder. The amount of air and fuel, and the relation between the two, determine the generated torque by the engine which in turn results in the engine speed, as described by the engine speed dynamics system. As motivated before, the ratio between the air and fuel in the cylinder play an important role in the catalyzer efficiency as well. For the purpose of control several variables are measured by the sensor system.
In the following sections the four subsystems are introduced to the reader. To keep track of the variables and parameters of the model, these are listed in Table 7.1 and Table 7.2 respectively.

7.1.1 Fueling System

When fuel is injected in the manifold, one part of it will reach the cylinder. The other part will form a liquid film on the manifold wall which is called the fuel puddle mass. The liquid film in turn vaporizes and reaches the cylinder as well. This process can be described by (Aquino 1981), (Beaumont et al. 1992), (Hendricks and Sorenson 1990), (Hires and Overington 1981), (Pieper and Mehrotra 1999):

$$\dot{m}_{fp}(t) = \chi_{fp}\dot{m}_{fi}(t) - \frac{m_{fp(t)}}{\tau_{fp}} \tag{7.3}$$

$$\dot{m}_{fc}(t) = (1 - \chi_{fp})\,\dot{m}_{fi}(t) + \frac{m_{fp}(t)}{\tau_{fp}} \tag{7.4}$$

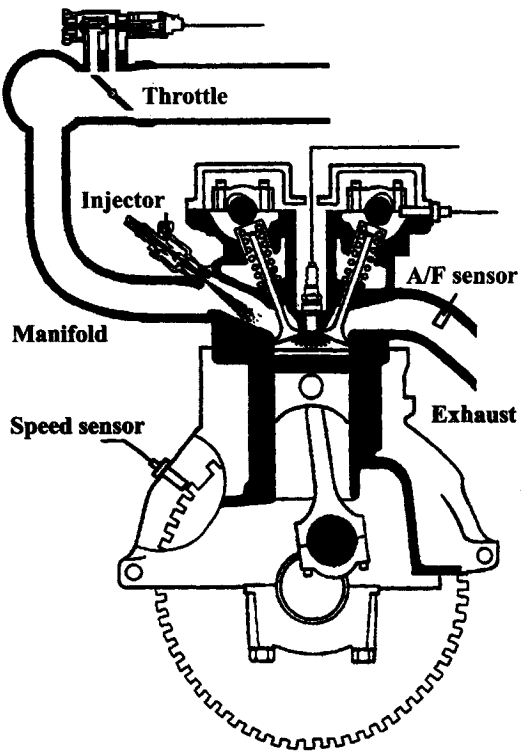


Fig. 7.1: Simple sketch of an spark ignition engine (from (Edwards and Spurgeon 1998)).

Variable	unit	Description
m_{am}	$[kg]$	Total mass of air in the manifold
m_{fp}	$[kg]$	Mass of fuel which forms a film on the manifold surface, known as the fuel puddle mass
$\dot{m}_{ac}$	$[kg/s]$	Mass flow of air in the cylinder
$\dot{m}_{fc}$	$[kg/s]$	Mass flow of fuel in the cylinder
$\dot{m}_{fi}$	$[kg/s]$	Injected fuel mass flow
$\dot{m}_{at}$	$[kg/s]$	Mass flow of air through the thottle
θ	$[rad]$	Throttle angle
A_{th}	$[m^2]$	Throttle open area
FPW	[ ]	Fuel Pulse width control input
λ	[ ]	Air fuel ratio
$\tilde{\lambda}$	[ ]	Measured air fuel ratio
N	$[rad/sec]$	Engine speed
p_m	$[Pa]$	Manifold air pressure
$\tilde{p}_m$	$[Pa]$	Measured manifold air pressure
T_{eng}	$[Nm]$	Produced torque by the engine
T_l	$[Nm]$	Torque load
T_r	$[Nm]$	Rolling friction torque
T_f	$[Nm]$	Engine friction torque
T_a	$[Nm]$	Aerodynamical drag torque
J	$[kgm^2]$	Moment of inertia of the engine

Table 7.1: Variables of the engine model.

Parameter	unit	Description
χ_{fp}	[ ]	Proportion of the injected fuel which forms the fuel film on the manifold surface
τ_{fp}	$[s]$	Evaporation time constant of fuel puddle mass
FPW_{offset}	[ ]	Fuel pulse-width offset
K_{inj}	$[kg/s]$	Injector gain
N_{cyl}	[ ]	Number of cylinders
C_d	[?]	Discharge coefficient
p_o	$[Pa]$	Ambient air temperature
T_o	$[K]$	Ambient air temperature
γ	[ ]	Gas constant
R	$[J/kg/K]$	Gas constant
η_{vol}	[ ]	Volumetric efficiency
D	$[m]$	Throttle diameter
V_{swept}	$[m^3]$	Effective volume per cylinder per stroke
V_m	$[m^3]$	Manifold volume
τ_λ	$[s]$	λ-sensor time constant
δ_λ	$[s]$	λ-sensor time delay
τ_{p_m}	$[s]$	Manifold air pressure sensor time constant
δ_a	$[s]$	Spark advance

Table 7.2: Parameters of the engine model.

where m_{fp} represents the fuel puddle mass, $\dot{m}_{fi}$ the injected fuel mass flow, and $\dot{m}_{fc}$ the fuel mass flow into the cylinder. The proportion of the injected fuel which forms the fuel film on the manifold surface is χ_{fp}, the parameter τ_{fp} determines the amount of evaporation. Both parameters strongly depend on the working conditions. For example the amount of evaporation will be strongly depending on the manifold temperature and the air flow along the fuel film. Therefore, the parameters χ_{fp} and τ_{fp} are by no means constants, they will vary over a large range depending on the working conditions.
The fuel is injected by an electronically operated device (the injector) which is triggered by a pulse-width signal. The actual injected fuel mass flow rate is given by (Pieper and Mehrotra 1999), (Powell et al. 1998):

$$\dot{m}_{fi}(t) = \frac{(FPW(t) - FPW_{offset})K_{inj}N_{cyl}}{2}\frac{N(t)}{60} \tag{7.5}$$

where FPW is the pulse width modulated signal applied to the injector, FPW_{offset} is the fuel pulse offset, K_{inj} the injector gain, N_{cyl} is the number of cylinders and N the engine speed.

7.1.2 Air Flow System

The air flow system is most probably the most complex subsystem of a spark ignition combustion engine. It is built up around the manifold. The manifold can be

considered as a buffer which is filled by the throttle and emptied by the cylinder, leading to the net flow of air into the manifold $\dot{m}_{am}$ given by:

$$\dot{m}_{am}(t) = \dot{m}_{at}(t) - \dot{m}_{ac}(t) \tag{7.6}$$

where $\dot{m}_{at}$ is the mass air flow past the throttle and $\dot{m}_{ac}$ is the mass air flow into the cylinder. By a very course approximation, the total air mass in the manifold m_{am} can be related to the manifold air pressure p_m by the gas law:

$$p_m(t)V_m = m_{am}(t)RT_m(t) \tag{7.7}$$

where V_m represents the volume of the manifold, T_m the manifold air temperature, and R is a gas-constant for air. Differentiating equation (7.7) with respect to time while assuming a constant manifold air temperature, and substituting equation (7.6) results in the following relation between the air manifold pressure and the air flow in- and out of the manifold:

$$\dot{p}_m(t) = \frac{RT_m}{V_m}\left(\dot{m}_{at}(t) - \dot{m}_{ac}(t)\right) \tag{7.8}$$

It should be mentioned that equation (7.8) is highly simplified. For a more accurate model we refer to for example (Cipollone and Sughayyer 2001).
The throttle regulates the amount of air entering the manifold. A simplified model for this process is given by (Heywood 1988), (Crossley and Cook 1991):

$$\dot{m}_{at}(t) = g(p_m)A_{th}(\theta) \tag{7.9}$$

with p_m the manifold air pressure and A_{th} the throttle open area which is controlled by the throttle position θ. The function $g(p_m)$ is given by:

$$g(p_m) = \frac{C_d p_o}{\sqrt{RT_o}}\left(\frac{2\gamma}{\gamma-1}\right)^{\frac{1}{2}}\left\{\left(\frac{p_m(t)}{p_o}\right)^{\frac{2}{\gamma}} - \left(\frac{p_m(t)}{p_o}\right)^{\frac{\gamma+1}{\gamma}}\right\}^{\frac{1}{2}} \tag{7.10}$$

where p_o and T_o represent the ambient air pressure and temperature respectively, C_d the discharge coefficient, and the gas constants γ and R. As can be seen in equation (7.9), the throttle open area is a function of the throttle angle denoted by θ. An approximation for the function A_{th} can be found to be (Kim et al. 1998):

$$\begin{aligned} A_{th}(\theta) \quad = \quad & \frac{\pi D^2}{4}\left\{\left(1 - \frac{\cos(\theta)}{\cos(\theta_c)}\right) + \frac{2}{\pi}\left[\frac{\theta}{\cos(\theta)}\sqrt{\cos^2(\theta) - \theta^2\cos^2(\theta_c)}\right]\right. \\ & \left. - \frac{\cos(\theta)}{\cos(\theta_c)}\sin^{-1}\left(\frac{\theta\cos(\theta_c)}{\cos(\theta)}\right) - \theta\sqrt{1-\theta^2} + \sin^{-1}(\theta)\right\} \end{aligned} \tag{7.11}$$

where θ is the throttle angle, θ_c is the throttle angle when closed, and D is the throttle diameter. From equation (7.11) it is not easy to determine which throttle angle is results in the required throttle open area. Therefore the open throttle is often approximated by a Taylor expansion or by means of a look-up table.

Finally, the air mass flow fed away by the cylinder is given by:

$$\dot{m}_{ac}(t) = \eta_{vol} \frac{V_{swept}}{2} \frac{N(t)}{60} \frac{1}{RT_m} p_m(t) \tag{7.12}$$

In this equation, η_{vol} represents the volumetric efficiency. The volumetric efficiency depends on many factors such as for example engine speed and the manifold air pressure.

7.1.3 Engine Speed Dynamics

The dynamics of the engine speed are closely related to the car dynamics and the gear-ratio. Without taking the different gear ratios into account, the speed dynamics can be modeled by (Pieper and Mehrotra 1999):

$$J(t)\dot{N}(t) = T_{eng}(t) - T_l(t) \tag{7.13}$$

where T_{eng} is the torque produced by the engine and T_l is a variable load torque. Bearing in mind that the gear-shift operation and different gear ratios are not taken into consideration, the torque generated by the engine depends mainly on the engine speed, the mass air flow into the cylinder, the air fuel ratio, and the spark advance which we consider to be constant. The load torque T_l can be represented by (Cho and Hedrick 1988):

$$T_l(t) = T_r(t) + T_f(t) + T_a(t) \tag{7.14}$$

where T_r represents the rolling friction, T_f the engine friction torque and T_a the aerodynamical drag.

7.1.4 Sensors

For the controller design the manifold air pressure p_m, the engine speed N and the air fuel ratio λ are assumed to be measured. The speed dynamics of the engine are assumed to be much slower then the time constant of the speed sensor. Therefore, the speed is assumed to be known exactly. Since this does not apply to the manifold air pressure sensor, the measured manifold air pressure $\tilde{p}_m$ can be represented by:

$$\tilde{p}_m(t) + \tau_{p_m}\dot{\tilde{p}}_m(t) = p_m(t) \tag{7.15}$$

where τ_{p_m} is the time constant of the manifold air pressure sensor.
For the measurement of the air fuel ratio, two sensor types are available which reproduce the air fuel ratio from exhaust gas measurements. The first type is the HEGO-sensor (Heated Exhaust Gas Oxygen), also called the λ-sensor, which only indicates whether the air fuel ratio is above or under the stoichiometric value. In this chapter we assume the availability of an UEGO-sensor (Universal Exhaust Gas Oxygen), also called the linear air fuel ratio sensor. The UEGO-sensor measures the exact air fuel ratio. However, the sensor exhibits a measurement time constant and a measurement time delay. If the UEGO-sensor is placed close enough to the

exhaust valve it is valid to assume that the measurement delay is constant. The dynamics of the UEGO-sensor can then be described by (Chang et al. 1993):

$$\tilde{\lambda}(t) + \tau_\lambda \dot{\tilde{\lambda}}(t) = \lambda(t - \delta_\lambda) \tag{7.16}$$

where λ is the real air fuel ration, $\tilde{\lambda}$ the measured air fuel ratio, τ_λ is the sensor time, and δ_λ is the constant measurement delay.

7.2 Controller Design

The controller design is split in two independent steps. First, Section 7.2.1 presents the air fuel ratio controller design, then Section 7.2.2 describes the speed controller design.

7.2.1 Air Fuel Ratio Controller

The air fuel ratio controller is constructed in two parts: a feedforward component $u_{\lambda,ff}$ and a feedback part $u_{\lambda,fb}$. The control input fed to the system is given by:

$$u_\lambda(t) = u_{\lambda,ff}(t) + u_{\lambda,fb}(t) \tag{7.17}$$

For controller design we assume that the air fuel ratio λ, the engine speed N and the manifold air pressure p_m are known. The system thus only has one unmeasurable state, namely the fuel puddle mass m_{fp}. The fuel puddle mass will be reconstructed by means of an observer for use in the feedforward controller. In the following text the parameters η_{vol}, χ_{fp}, and τ_{fp} are for simplicity denoted by constants. However, at a later stage they will be approximated by polynomials to give a better performance.

For simplicity the injected fuel flow $\dot{m}_{fi}$ will be considered to be the input to the system. As mentioned earlier, the actual input is a pulse width modulated signal which, by manipulation of equation (7.5), can be found to be:

$$FPW(t) = \frac{FPW_{offset} + u_\lambda(t)120}{K_{inj} N_{cyl} N(t)} \tag{7.18}$$

In this section a straightforward feedforward controller is derived. From equation (7.12) we can estimate the air mass flow rate into the cylinder $\dot{m}_{ac}$. Dividing this value by the stoichiometric air fuel ratio we obtain the ideal set-point for the fuel mass flow rate into the cylinder $\dot{m}_{fc}$. Using equation (7.4) we can compute the amount of fuel that needs to be injected. Combining these computations in a single equation leads to the following feedforward control component:

$$u_{\lambda,ff}(t) = \frac{\eta_{vol} V_{swept} N(t) p_m(t)}{120 R T_m (1 - \chi_{fp}) 14.7} - \frac{m_{fp}(t)}{(1 - \chi_{fp})\tau_{fp}} \tag{7.19}$$

A very important advantage of a feedforward controller is that it does not depend on the delayed air/fuel ratio measurement. In equation (7.19), we assume knowledge of the manifold air pressure p_m, the manifold air temperature T_m, the engine speed

N, the fuel puddle mass m_{fp} as well as all parameters. These assumptions can of course not be justified. The manifold air temperature can be either measured or approximated by for example the ambient air temperature. The manifold air pressure and the engine speed are measurable as well, however the fuel puddle mass cannot be measured. The unavailability of the fuel puddle mass can be circumvented by the use of an observer. For the observer the following model can be created from equations (7.2), (7.3), (7.4), (7.12), (7.17),:

$$\dot{\hat{m}}_{fp}(t) = \chi_{fp}\left(u_\lambda(t) - L_\lambda e(t)\right) - \frac{1}{\tau_{fp}}\hat{m}_{fp}(t) \tag{7.20}$$

$$e(t) = \frac{u_\lambda(t) - Le(t) - \dot{\hat{m}}_{fp}(t)}{\eta_{vol}V_{swept}N(t)p_m(t)} - \frac{1}{\lambda(t)} \tag{7.21}$$

where $\hat{m}_{fp}$ is the estimated fuel puddle mass. The observer gain $L_\lambda \in \mathbb{R}$ should be sufficiently large that the measured output is tracked.
Another problem is the fact that some of the parameters which were assumed to be constant may in fact vary over a wide range under different working conditions. This problem is traditionally solved by creating lookup tables which yield appropriate values for different circumstances. Another solution is to estimate the parameters as functions of, for example, the manifold air pressure, temperature and engine speed. Indeed, incorporating a robust feedback controller may reduce the sensitivity of the closed-loop system to variations in the appropriate parameters used in the feedforward signal.
We can expect the feedforward controller, with constant parameters, to yield substantial errors, particularly in the steady-state. We thus introduce a feedback controller to compensate for the error. The feedback controller we consider is a sliding mode controller. The output λ can be found from equations (7.2),(7.4), and (7.12) to be:

$$\lambda(t) = \frac{\eta_{vol}V_{swept}p_m(t)}{2RT_m\left((1-\chi_{fp})\dot{m}_{fi}(t) + \frac{m_{fp}(t)}{\tau_{fp}}\right)}\frac{N(t)}{60} \tag{7.22}$$

In order to make the system affine in the input we define a hypothetical output which is the inverse of λ, which leads to the affine system:

$$\frac{1}{\lambda(t)} = \frac{120RT_m}{\eta_{vol}\tau_{fp}V_{swept}N(t)p_m(t)}m_{fp}(t) + \frac{120RT_m(1-\chi_{fp})}{\eta_{vol}V_{swept}N(t)p_m(t)}\dot{m}_{fi}(t) \tag{7.23}$$

Now we define the switching function to be:

$$\sigma_\lambda(t) = \frac{1}{\lambda(t)} - \frac{1}{\lambda_d(t)} \tag{7.24}$$

where λ_d is the desired (stoichiometric) air fuel ratio. It was presented in Chapter 2 that convergence towards the sliding surface is guaranteed if the following reaching law is satisfied::

$$\sigma_\lambda(t)\dot{\sigma}_\lambda(t) < -\eta|\sigma_\lambda(t)| \tag{7.25}$$

for some positive constant scalar η. The same reaching law can be used to establish a controller for the air fuel ratio control. As can be concluded from equation (7.23) the system has relative degree zero (i.e. the input appears directly in the output). So in order to satisfy the above condition the first derivative of the system input will be discontinuous, which effectively results in an integral sliding mode controller. The disadvantage of this structure is that all model uncertainties and disturbances are unmatched with the input. Therefore, the sliding mode cannot be maintained under all circumstances. Especially during fast transients the closed-loop system will not remain in the sliding mode. However, invariance against these disturbances and uncertainties cannot be guaranteed in any way due to the measurement delay of the air fuel ratio.

As introduced in Chapter 2, a sliding mode controller is normally designed in two parts. The first part is the continuous control or equivalent control part which is based on the model knowledge and can be obtained by solving the equation $\dot{\sigma}_\lambda = 0$ for the input u_λ. The second part is the discontinuous control part which is used to compensate for modeling errors and disturbances. Since the aim of the feedforward controller is to steer the output to the sliding surface we may interpret this as the continuous control part according to the interpretation presented in (Monsees et al. 1999). Therefore the sliding mode feedback controller will only consist of the discontinuous control part.

Since we can expect a considerable effect of unmatched disturbances during transients, we cannot expect the controller to maintain sliding mode. In order to speed up the reaching phase we introduce a proportional term in the control law. Furthermore, to reduce chattering we replace the sign function by the saturation function. This results in the following sliding mode feedback law:

$$\dot{u}_{\lambda,fb}(t) = -K_{\lambda,d}\mathrm{Sat}\left(\frac{\sigma_\lambda(t)}{\phi}\right) - K_{\lambda,p}\sigma_\lambda(t) \tag{7.26}$$

where $K_{\lambda,d}$ is the switching gain, $K_{\lambda,c}$ the proportional gain, and the positive constant ϕ specifies the linear band of the saturation function. We note that the integral sliding mode feedback signal may be found by integrating the above equation:

$$u_{\lambda,fb}(t) = u_{\lambda,fb}(0) + \int_{\nu=0}^{t}\left(-K_{\lambda,d}\mathrm{Sat}\left(\frac{\sigma_\lambda(\nu)}{\phi}\right) - K_{\lambda,p}\sigma_\lambda(\nu)\right)d\nu \tag{7.27}$$

7.2.2 Speed Controller

For the design of the speed controller we use the approach presented in (Lu and Spurgeon 1997), (Lu and Spurgeon 1998), (Lu and Spurgeon 1999*a*), (Lu and Spurgeon 1999*b*). This design procedure is based upon a model description which is expressed in the Generalized Controller Canonical Form (GCCF) where essentially the system output of interest and its derivatives up to some appropriate order are used as the system states. The sliding mode design approach is then a two-stage approach involving the selection of an appropriate sliding surface to yield desirable performance and the selection of a controller to ensure the sliding mode is attained. The development of the GCCF together with an outline of the control design phase

is now presented. It is assumed that a model in GCCF is readily available. This assumption is by no means trivial and therefore will be assessed, specifically for the simulation example in Section 7.3. It involves numerous nonlinear transformations and should be performed using some symbolic software package.
The engine model, introduced in Section 7.1, has as far as the speed control is concerned two states (engine speed and manifold air pressure), one input (throttle angle), and one output (engine speed). However, by the correct state transformation the system can be represented by the engine speed and the engine acceleration as well. For the controller design we assume that the model is brought into the following GCCF description:

$$\dot{z}_1(t) = z_2(t) \tag{7.28}$$
$$\dot{z}_2(t) = \eta_0(z) + \eta_u(z)u_N(t) \tag{7.29}$$

where:

$$z(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} N(t) \\ \dot{N}(t) \end{bmatrix} \tag{7.30}$$
$$y(t) = N(t) \tag{7.31}$$
$$u_N(t) = A_{th}(\theta) \tag{7.32}$$

Before considering the selection of the sliding surface, it is important to note that the GCCF of the automotive engine satisfies the basic assumptions of the robust design method developed in (Lu and Spurgeon 1997). These are:

- The nominal system, described by equations (7.28) and (7.29), is minimum phase, i.e. its zero dynamics are asymptotically stable.
- Equation (7.29) is continuous and differentiable with respect to time.
- Equation (7.29) is affine in the input.

It is straightforward to prove that these conditions are satisfied in the operating range of the engine given by:

$$\begin{array}{rrl} 0 < N_{min} \leq & N(t) & \leq N_{max} \\ 0 < p_{m,min} \leq & p_m(t) & \leq p_{m,max} < p_o \end{array} \tag{7.33}$$

The first step in the design for a SMC is the selection of a sliding surface.The following sliding surface is chosen:

$$\sigma_N(t) = s_1 \left(z_1(t) - N_{ref}(t)\right) + \left(z_2(t) - \dot{N}_{ref}(t)\right) \tag{7.34}$$

where $\sigma_N \in \mathbb{R}$ is the switching function of the speed controller, N_{ref} is the speed reference, and s_1 is a tuning parameter that determines the dynamics in the sliding mode. In fact, s_1 is the pole of the reduced order system. This forces the parameter s_1 to be positive if stable first order dynamics are desired. Such a restriction is the only theoretical constraint for the sliding surface design.

From Chapter 2 we know that convergence towards the sliding surface is guaranteed if the following reaching law is satisfied:

$$\dot{\sigma}_N(t) = -K_{N,p}\sigma_N(t) - K_{N,s}\text{sign}\left(\sigma_N(t)\right) \tag{7.35}$$

where $K_{N,p} \in \mathbb{R}$ and $K_{N,s} \in \mathbb{R}$ are positive constant design parameters. The tuning of such parameters determines the controller robustness, and strongly depends on the system uncertainties. Considering the GCCF of the nominal system, the corresponding real system can be represented by adding an extra term to equation (7.29) that models any source of uncertainty, such as model uncertainties and external disturbances, leading to the following representation:

$$\dot{z}_1(t) = z_2(t) \tag{7.36}$$

$$\dot{z}_2(t) = \eta_0(z) + \eta_u(z)u_N(t) + \Delta_{GCCF}(t) \tag{7.37}$$

Differentiating equation (7.34) with respect to time, and substituting equations (7.36) and (7.37), results in:

$$\begin{aligned}\dot{\sigma}_N(t) = {} & s_1 z_2(t) - s_1\dot{N}_{ref}(t) + \eta_0(z) + \eta_u(z)u_N(t) \\ & -\ddot{N}_{ref}(t) + \Delta_{GCCF}(t)\end{aligned} \tag{7.38}$$

Combining the reaching law (7.35) and equation (7.38), and solving for u_N (thereby omitting the unknown term Δ_{GCCF}) results in the control law:

$$\begin{aligned}u_N(t) = {} & \frac{1}{\eta_u(z)}\{-K_{N,p}\sigma_N(t) - K_{N,s}\text{sign}\left(\sigma_N(t)\right) \\ & -s_1\left(z_2(t) - \dot{N}_{ref}(t)\right) - \left(\eta_0(z) - \ddot{N}_{ref}(t)\right)\}\end{aligned} \tag{7.39}$$

Inserting control law (7.39) into equation (7.38) results in:

$$\dot{\sigma}_N(t) = \Delta_{GCCF}(t) - K_{N,p}\sigma_N(t) - K_{N,s}\text{sign}\left(\sigma_N(t)\right) \tag{7.40}$$

Consequently, the reaching law:

$$\sigma_\lambda(t)\dot{\sigma}_N(t) < -\eta|\sigma_N(t)| \tag{7.41}$$

is only satisfied if:

$$K_{N,p}\sigma_N(t) + K_{N,s}\text{sign}\left(\sigma_N(t)\right) > \Delta_{GCCF}(t) + \eta\ \text{sign}\left(\sigma_N(t)\right) \tag{7.42}$$

for some arbitrary small positive $\eta \in \mathbb{R}$ determining the speed of convergence towards the sliding surface. It is assumed that the uncertainty term Δ_{GCCF} is bounded by:

$$\Delta_{GCCF}(z, N_{ref}, \dot{N}_{ref}) \leq \rho_1\|z(t)\| + \rho_0 \tag{7.43}$$

Then, to guarantee global exponential stability of the closed loop uncertain system, the controller gains should satisfy:

$$K_{N,p} > \rho_1 + \frac{c^2}{4\phi s_1^2} + \frac{\rho_1\left(1 + \max(1, s_1)\right)^2}{4(1-\phi)} \tag{7.44}$$

$$K_{N,s} > \rho_0 \tag{7.45}$$

with c and ϕ arbitrary constants satisfying $c > 0$ and $0 < \phi < 1$. However, in practice the parameters $K_{N,p}$ and $K_{N,s}$ will be determined by experimentation.
As was previously stated, the engine speed $z_1 = N$ and the manifold pressure p_m are the only measured variables. Therefore, to compute the control action, it is better to maintain η_0 and η_u in terms of z_1 and p_m instead of writing them as functions of z only. For the design of the controller it was convenient to have a state description based on the engine speed and acceleration. However, for the controller implementation it is more convenient to use the variables in the original (measured) coordinates given by the engine speed and the manifold air pressure, thereby avoiding errors caused by the transformation to other than measured coordinates.
Furthermore, to circumvent chattering of the throttle position, the sign function is replaced by the saturation function, leading to the control law:

$$\begin{aligned} u_N(t) &= \frac{1}{\tilde{\eta}_u(z_1, p_m)} \left\{ -K_{N,p}\sigma_N(t) - K_{N,s}\text{sat}\left(\frac{\sigma_N(t)}{\phi_N}\right) \right. \\ &\quad \left. -s_1\left(z_2(t) - \dot{N}_{ref}(t)\right) - \left(\tilde{\eta}_0(z_1, p_m) - \ddot{N}_{ref}(t)\right) \right\} \end{aligned} \tag{7.46}$$

In theory, the control signal (7.39) is identical to the control signal (7.46). However, the latter is written in terms of the measurable signals z_1 and p_m and therefore is easier to implement.
It can be seen from control law (7.46) that the acceleration $z_2 = \dot{N}$ is required. Although the acceleration can be obtained by direct differentiation of the measured engine speed, measurement noise perturbing this measurement will degrade the accuracy of the obtained acceleration significantly. Therefore, we construct the following high-gain observer to estimate the acceleration:

$$\dot{\hat{z}}_1(t) = \hat{z}_2(t) + L_{N,1}\left(z_1(t) - \hat{z}_1(t)\right) \tag{7.47}$$

$$\dot{\hat{z}}_2(t) = \eta_0(z_1, p_m) + \eta_u(z_1, p_m)u_N(t) + L_{N,2}\left(z_1(t) - \hat{z}_1(t)\right) \tag{7.48}$$

where the observer gain $L_N \in \mathbb{R}^2$ ($L_N = [L_{N,1}\ L_{N,2}]^T$) should be chosen sufficiently large to guarantee convergence for the observer.
It should be remarked that the control u is not the practical throttle angle. Instead, the throttle angle θ has to be solved from equation (7.11) such that it provides the required open throttle area. Despite the highly nonlinear structure of the relation between the throttle open area and the throttle angle, this does not impose any severe problems. Since the relation between θ and $A_{th}(\theta)$ is bijective, it can be solved off-line for several values of θ and stored in look-up tables, or alternatively be approximated by some polynomial function.

7.3 Simulation Results

To test the air fuel ratio controller and speed controller, developed in Section 7.2, they are applied to an extensive nonlinear industrial simulation model used for verification and design. In this simulation model, parameters are stored in look-up tables and a much more detailed dynamical model is present compared to the

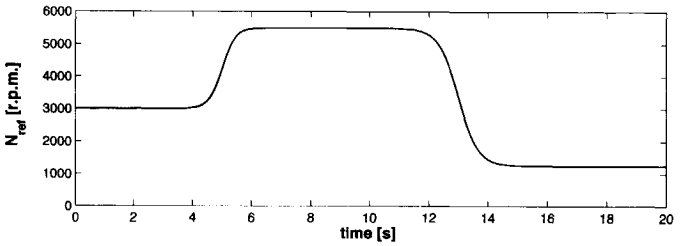


Fig. 7.2: Engine speed reference speed trajectory.

simplified model introduced in Section 7.1. The controllers are designed to maintain the air fuel ratio at the stoichiometric value while tracking the given speed reference trajectory like for example can be found in in Figure 7.2. Despite the fact that these controllers are based on the simplified model introduced in Section 7.1, they will proof to give excellent stabilization of the engine speed and air fuel ratio for the realistic industrial verification model.

To establish the best possible match between the simplified model and the extensive verification model, the parameters of the simplified model are approximated by polynomial functions instead of keeping them constant. Based on the parameter estimations, the air fuel ratio controller can straightforwardly be implemented. For the speed controller design, the model was assumed to be in GCCF. Since the simplified model presented in Section 7.1 is not in GCCF, we present this transformation in this section.

The controllers applied to the extensive nonlinear simulation model presented in this Section are based on the simplified model presented in Section 7.1. Therefore we first determine the parameters for the simplified model, such that it describes the extensive simulation model to its best capabilities. The parameters for each subsystem are given next.

Fueling system

The parameters χ_{fp} and τ_{fp} have been approximated by the polynomial functions:

$$\chi_{fp}(N, p_m) = c_{\chi,0} + c_{\chi,1} p_m(t) + c_{\chi,2} N(t) \tag{7.49}$$

$$\tau_{fp}(N, p_m) = c_{\tau,0} + c_{\tau,1} p_m(t) + c_{\tau,2} N(t) + c_{\tau,3} N(t) p_m(t) \tag{7.50}$$

It should be mentioned however, that for example the manifold air temperature has an important effect on these parameters. However, since the manifold air temperature is assumed to be constant this effect has been neglected. The assumption of a constant manifold air temperature is verified in simulation where it can be seen that the manifold air temperature hardly deviates from the ambient air temperature.

Air flow system

The functions $g(p_m)$ and A_{th}, together specifying the air mass flow past the throttle

(Equation (7.9)), are approximated by:

$$A_{th}(\theta) = c_{th,0} + c_{th,1}\theta(t) + c_{th,2}\theta^2(t) + c_{th,3}\theta^3(t) \tag{7.51}$$

$$g(p_m) = \begin{cases} 1 & p_m(t) \leq \frac{1}{2}p_o \\ \frac{2}{p_o}\sqrt{p_o p_m(t) - p_m^2(t)} & p_m(t) > \frac{1}{2}p_o \end{cases} \tag{7.52}$$

Equation (7.52) approximates Equation (7.10) closely within the range of operation. The air mass flow into the cylinder (Equation (7.12)), is approximated by:

$$\dot{m}_{ac}(t) = c_{ac,0} + c_{ac,1}N(t)p_m(t) + c_{ac,2}N(t)p_m^2(t) + c_{ac,3}N^2(t)p_m(t) \tag{7.53}$$

Engine speed dynamics system
The torque components can be described by (Cho and Hedrick 1988):

$$\begin{aligned} T_{en}(\dot{m}_{ac}, \lambda, \delta_a, N) = {} & c_{en,0} + c_{en,1}\frac{\dot{m}_{ac}(t)}{N(t)} + c_{en,2}\lambda(t) + c_{en,3}\lambda^2(t) \\ & + c_{en,4}\delta_a + c_{en,5}\delta_a^2 + c_{en,6}N(t) + c_{en,7}N^2(t) \\ & + c_{en,8}\delta_a N(t) + c_{en,9}\delta_a\frac{\dot{m}_{ac}(t)}{N(t)} \\ & + c_{en,10}\delta_a^2\frac{\dot{m}_{ac}(t)}{N(t)} \end{aligned} \tag{7.54}$$

$$T_r = c_{tr} \tag{7.55}$$

$$T_f(N) = c_{tf,0} + c_{tf,1}N(t) \tag{7.56}$$

$$T_a(N) = c_{ta}N^2(t) \tag{7.57}$$

Sensor system
The sensor dynamics are neglected in the controller design, therefore no parameters are needed. The design procedure for the speed controller was defined for a model in the form:

$$\dot{z}_1(t) = z_2(t) \tag{7.58}$$

$$\dot{z}_2(t) = \eta_0(z) + \eta_u(z)u_N(t) \tag{7.59}$$

where:

$$z(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} N(t) \\ \dot{N}(t) \end{bmatrix} \tag{7.60}$$

$$y(t) = N(t) \tag{7.61}$$

$$u_N(t) = A_{th}(\theta) \tag{7.62}$$

In the following, the transformation from the simplified model to the GCCF is presented. Obtaining the GCCF is by no means a straightforward task. It involves

Parameter		Value
c_{tr}	=	2.04
$c_{\chi,0}$	=	-0.12
$c_{\chi,1}$	=	$8.86e^{-6}$
$c_{\chi,3}$	=	$3.17e^{-5}$
$c_{\tau,0}$	=	$6.47e^{-6}$
$c_{\tau,1}$	=	$1.08e^{-6}$
$c_{\tau,2}$	=	$6.27e^{-6}$
$c_{\tau,3}$	=	$-2.02e^{-10}$
$c_{en,0}$	=	-543
$c_{en,1}$	=	$1.40e^{7}$
$c_{en,2}$	=	66
$c_{en,3}$	=	-2.60
$c_{en,4}$	=	0.78
$c_{en,5}$	=	$-8.50e^{-3}$
$c_{en,6}$	=	$8.50e^{-3}$
$c_{en,7}$	=	$-3.50e^{-6}$
$c_{en,8}$	=	$1.5e^{-4}$
$c_{en,9}$	=	$9.6e^{4}$
$c_{en,10}$	=	$-1.90e^{3}$
c_{ta}	=	$2.22e^{-4}$

Parameter		Value
$c_{tf,0}$	=	5
$c_{tf,1}$	=	0.036
$c_{th,0}$	=	$2.81e^{-5}$
$c_{th,1}$	=	$8.98e^{-10}$
$c_{th,2}$	=	$1.03e^{-4}$
$c_{th,3}$	=	$-0.63e^{-6}$
$c_{ac,0}$	=	$-3.66e^{-4}$
$c_{ac,1}$	=	$8.98e^{-10}$
$c_{ac,2}$	=	$-3.37e^{-15}$
$c_{ac,3}$	=	$1e^{-12}$
p_o	=	$1e^{5}$
J	=	0.28
δ_a	=	15
R	=	287
T_o	=	298.15
V_m	=	$2.975e^{-3}$
N_{cyl}	=	6
γ	=	1.4
c_{ta}	=	$2.22e^{-4}$

Table 7.3: Parameters values of the model.

the symbolic manipulation of numerous nonlinear equations and thus can best be performed by some symbolic software package, like Maple or Mathematica.
As a first step we obtain the functions $\eta_u(z_1, p_m)$ and $\eta_0(z_1, p_m)$. Using Mathematica, the functions $\eta_u(z_1, p_m)$ and $\eta_0(z_1, p_m)$ can be determined to be:

$$\eta_u(z_1, p_m) = \frac{RT_m}{V_m J} \{c_{ac,1} + 2c_{ac,2} p_m(t) + c_{ac,3} z_1(t)\} g(p_m) \tag{7.63}$$

$$\eta_0(z_1, p_m) = \frac{1}{J^2} \sum_{i=-3}^{3} \sum_{j=0}^{3} C_{i,j} z_1^i(t) p_m^j(t) \tag{7.64}$$

Before defining the coefficients $C_{i,j}$ we introduce the constants:

$$B_1 = c_{en,0} + c_{en,2}\lambda + c_{en,3}\lambda^2 + c_{en,4}\delta_a + c_{en,5}\delta_a^2 \tag{7.65}$$

$$B_2 = \pi(c_{en,1} + c_{en,9}\delta_a + c_{en,10}\delta_a^2) \tag{7.66}$$

$$B_3 = C_{en,6} + C_{en,8}\delta_a \tag{7.67}$$

$$B_4 = J\frac{RT_m}{V_m} \tag{7.68}$$

The coefficients $C_{i,j}$ can then be found to be:

$$C_{-3,0} = -(C_{ac,0}B_2)^2 \tag{7.69}$$

$$C_{-3,1} = 0 \tag{7.70}$$

$$C_{-3,2} = 0 \tag{7.71}$$

$$C_{-3,3} = 0 \tag{7.72}$$

$$C_{-2,0} = -B_2 c_{ac,0}\,(B_1 - c_{tf,0} - c_{tr}) \tag{7.73}$$

$$C_{-2,1} = -c_{ac,0}B_2^2 c_{ac,1} \tag{7.74}$$

$$C_{-2,2} = -c_{ac,0}B_2^2 c_{ac,2} \tag{7.75}$$

$$C_{-2,3} = 0 \tag{7.76}$$

$$C_{-1,0} = 0 \tag{7.77}$$

$$C_{-1,1} = 0 \tag{7.78}$$

$$C_{-1,2} = 0 \tag{7.79}$$

$$C_{-1,3} = 0 \tag{7.80}$$

$$C_{0,0} = (B_3 - c_{tf,1})(B_1 - c_{tr} - c_{tf,0}) + B_2 c_{ac,0}(c_{en,7} - c_{ta} - c_{ac,1}B_4) \tag{7.81}$$

$$C_{0,1} = B_2\{(B_3 - c_{tf,1})c_{ac,1} - 2c_{ac,2}c_{ac,0}B_4 - c_{ac,3}(c_{tf,0} + c_{tr} - B_1)\} \tag{7.82}$$

$$C_{0,2} = B_2\{(B_3 - c_{tf,1})c_{ac,2} + B_2 c_{ac,3}c_{ac,1}\} \tag{7.83}$$

$$C_{0,3} = B_2^2 c_{ac,3}c_{ac,2} \tag{7.84}$$

$$C_{1,0} = (c_{tf,1} - B_3)^2 - B_2 c_{ac,3}c_{ac,0}B_4 + 2(c_{ta} - c_{en,7})(c_{tf,0} + c_{tr} - B_1) \tag{7.85}$$

Parameter		Value
s_1	=	5
$K_{N,p}$	=	41
$K_{N,s}$	=	$1.14e^5$
ϕ_N	=	200
L_N	=	$[50 \;\; 8e^4]^T$

Parameter		Value
$K_{\lambda,p}$	=	1.4
$K_{\lambda,s}$	=	$5e^{-4}$
ϕ_λ	=	$3.3e^{-4}$
L_λ	=	$1e^3$

Table 7.4: Controller parameters. The left table contains the parameters for the speed controller, the right table for the air fuel ratio controller.

$$C_{1,1} = 2B_2\left\{(c_{en,7}-c_{ta})c_{ac,1}-(c_{tf,1}-B_3)c_{ac,3}-\frac{1}{2}c_{ac,1}^2B_4\right\} \quad (7.86)$$

$$C_{1,2} = B_2\left\{2c_{ac,2}(c_{en,7}-c_{ta}-1.5c_{ac,1}B_4)+B_2c_{ac,4}^2\right\} \quad (7.87)$$

$$C_{1,3} = -2B_2c_{ac,2}^2B_4 \quad (7.88)$$

$$C_{2,0} = 4(c_{tf,1}-B_3)(c_{ta}-c_{en,7}) \quad (7.89)$$

$$C_{2,1} = B_2c_{ac,3}\left\{3(c_{en,7}-c_{ta})-ac_{ac,1}B_4\right\} \quad (7.90)$$

$$C_{2,2} = -2B_2c_{ac,3}c_{ac,2}B_4 \quad (7.91)$$

$$C_{2,3} = 0 \quad (7.92)$$

$$C_{3,0} = 2(c_{en,7}-c_{ta}) \quad (7.93)$$

$$C_{3,1} = -B_2c_{ac,3}^2B_4 \quad (7.94)$$

$$C_{3,2} = 0 \quad (7.95)$$

$$C_{3,3} = 0 \quad (7.96)$$

The functions $\eta_u(z)$ and $\eta_0(z)$ can ten be obtained by replacing p_m in equations (7.63) and (7.64) by:

$$p_m(z) = \frac{c_{ac,3}z_1(t)-c_{ac,1}-\sqrt{\Psi(z)}}{-2c_{ac,2}} \quad (7.97)$$

where the function $\Psi(z)$ is defined by:

$$\begin{aligned}\Psi(z) = \; & \tfrac{1}{B_2}\left\{\left(4c_{ac,2}(c_{ta}-c_{en,7})+B_2c_{ac,3}^2\right)z_1^2(t)\right. \\ & +(2B_2c_{ac,3}c_{ac,1}+4c_{ac,2}(c_{tf,1}-B_3)) \\ & (4c_{ac,2}(c_{tf,0}+c_{tr}-B_1)+B_2c_{ac,1}^2) \\ & \left.\tfrac{4B_2c_{ac,2}c_{ac,0}}{z_1(t)}+4c_{ac,2}Jz_2(t)\right\}\end{aligned} \quad (7.98)$$

The last step in the design procedure is to obtain parameters for the controllers. They are given in Table 7.4. The controller parameters have been tuned such that the steady state error of the engine speed is small and the air fuel ratio is close to stoichiometry, even during fast transients. In this tuning procedure we have paid the greatest attention to the air fuel ratio to optimize the catalyzer efficiency, thereby accepting the reduced acceleration performance.

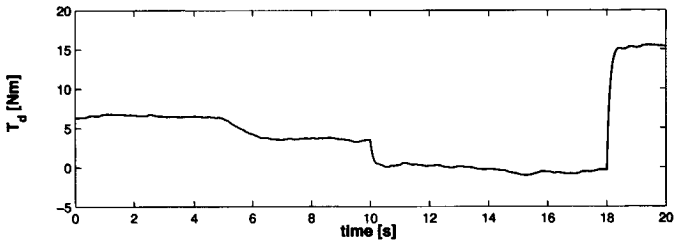


Fig. 7.3: A typical disturbance torque representing sudden activation or deactivation of car accessories, such as air conditioning, windscreenwipers, windows and sunroof, and other random torque disturbances.

The fact that the parameters are approximated and dynamics are neglected, together with aging and variations between engines due to production tolerances, results in significant model uncertainties. Furthermore, the dynamics of the sensors as well as the inherent measurement delay of the air fuel measurements are neglected in the controller design. Therefore, mismatches of up to 25% between parameters of the simulation model and the design model have been included for the tests. In addition to this model- and parameter-uncertainty, a realistic disturbance torque T_d has been added to the load torque T_l to consider sudden activation or deactivation of car accessories (such as air conditioning, windscreenwipers, windows and sunroof) and other unspecified torque disturbances. The disturbance torque is presented in Figure 7.3. The achieved engine speed N and the corresponding throttle open area A_{th} (as a percentage of the maximum open area) under these extreme circumstances are presented in Figure 7.4. The air fuel ratio λ and the injected fuel mass flow $\dot{m}_{fi}$ are depicted in Figure 7.5.
Despite the large engine speed difference over a relatively short time span, especially between $t = 12\ s$ and $t = 15\ s$, the air fuel ratio remains close to the stoichiometric value, almost never crossing the 0.1% error boundary. Only during the fast transient, the air fuel ratio deviates slightly from the stoichiometric value crossing only momentarily the 1% boundary at $t \approx 13\ s$. The fast change of the disturbance torque at $t \approx 18\ s$ (see Figure 7.3) hardly has an effect on the air fuel ratio and the engine speed.
The engine speed does not track the reference trajectory perfectly, however the steady state error is very small. Better tracking of the reference trajectory during fast transients would involve higher controller gains resulting in a chattering behavior of the throttle position. Under these circumstances the air fuel ratio deviates further from the stoichiometric value which is for reasons of catalytic efficiency not desirable. Another attractive feature of the demonstrated simulation results is the smoothness of the control signals, making them attractive for practical application.
In Figure 7.6 the manifold air pressure p_m and the manifold air temperature T_m are plotted. Despite the fact that the manifold air temperature was assumed to be constant it can be seen that it does in fact vary within a range of about 3%. However, it can be seen that in steady-state, the manifold air temperature corresponds with

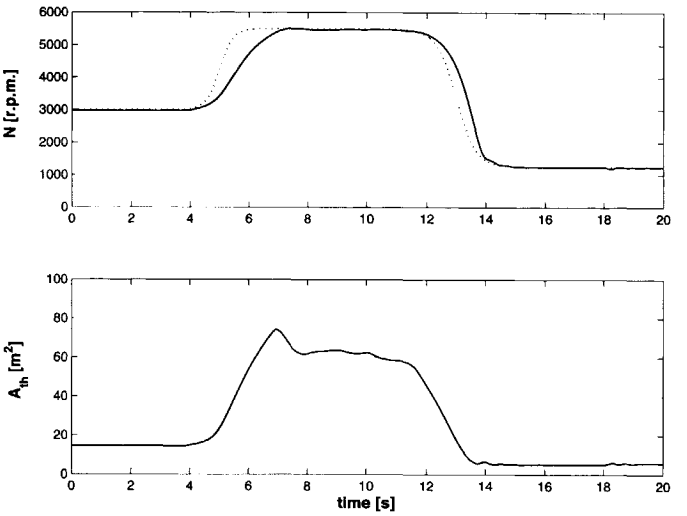


Fig. 7.4: Simulation results for the given speed reference trajectory and disturbance torque. The top figure represents the engine speed (solid line) and engine reference speed (dotted line), the bottom figure represents the throttle position as a percentage of the maximum opening.

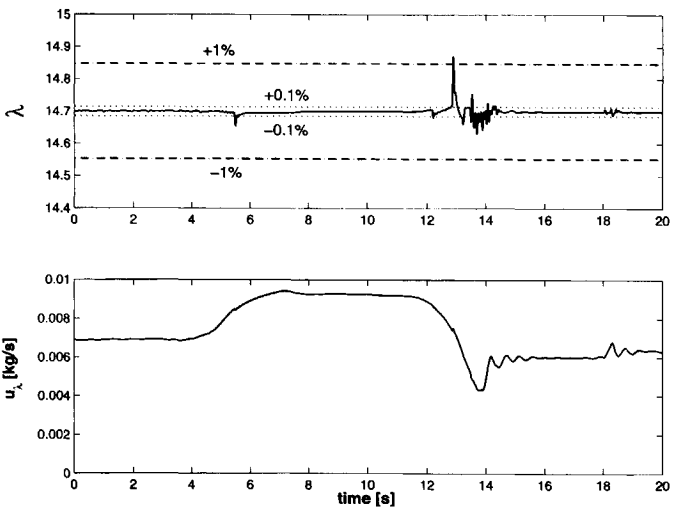


Fig. 7.5: Simulation results for the given speed reference trajectory and disturbance torque. The top figure represents the air fuel ratio (solid line), the $\pm$ 0.1% error boundary (dotted lines), and the $\pm$ 1% error boundary (dashed lines). The bottom figure represents the injected fuel mass flow.

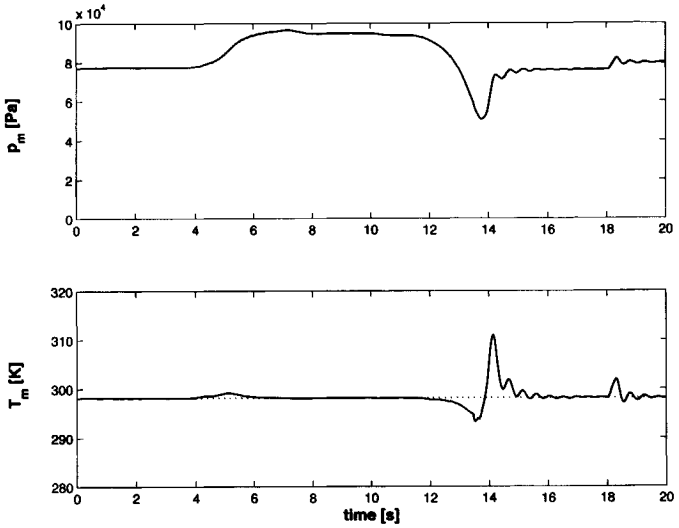


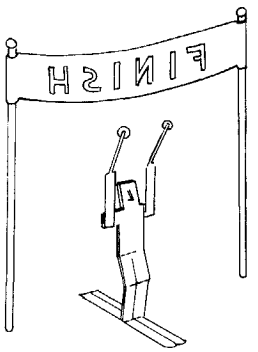
Fig. 7.6: Simulation results for the given speed reference trajectory and disturbance torque. The top figure represents the manifold air pressure, the bottom figure represents the manifold air temperature, where the dotted line represents the ambient air temperature.

the ambient temperature which can be used as an approximation in the absence of a manifold air temperature sensor.

7.4 Summary

In this chapter a design procedure for an air fuel ratio controller and a speed controller have been presented. While both are based on the sliding mode methodology, their structure is different. The measurement of the air fuel ratio is delayed by a, possibly time varying, measurement delay. Therefore, the air fuel ratio controller is divided in a feedforward part and a feedback part. The feedforward part is based on the measured engine speed and manifold air pressure. These values are measured without a time delay, and therefore respond rapidly to throttle changes. The feedback part is based on the measured air fuel ratio and is used to reduce the steady-state error. The speed controller is designed by first transferring the model to the GCCF, from which the design of the controller easily follows. The derived controllers have been applied to an extensive nonlinear simulation model which incorporates many dynamics neglected in the controller design. The simulation results clearly demonstrate the applicability of the proposed controller structure. The air fuel ratio remains within $\pm 0.1\%$ of the stoichiometric value, thereby optimizing the catalytic efficiency. Only during fast transients the air fuel ratio momentarily leaves the $\pm 0.1\%$ error band around stoichiometry. Also the speed controller demonstrates good stabilization of the engine speed. During transients the engine speed does not

perfectly match the reference trajectory. However, improving this tracking performance reduces the performance of the air fuel ratio controller thereby increasing air pollution drastically. The controller signals are smooth and therefore applicable to real-life engine control.
The results presented in this chapter have been published in (Monsees et al. 2000), (Puleston et al. 2001), (Monsees et al. 2001), (Puleston et al. 2002).



8

Conclusions & Recommendations

8.1 Conclusions

This thesis is primarily devoted to the field of Sliding Mode Control (SMC). An important conclusion which may be drawn from the laboratory tests is that presented Discrete-Time SMC (DSMC) can yield a substantial improvement in performance compared to Continuous-Time SMC (CSMC) when applied to a sampled-data system. Laboratory tests have shown that, for relatively high sampling frequencies, the performance of DSMC applied to a sampled data system is comparable to the results of CSMC. Decreasing the sampling frequency degrades the performance of the CSMC significantly, even after tuning the CSMC to the new circumstances. Of course, the DSMC performs better for higher- than for lower sampling frequencies as well. However, the decrease in performance of a DSMC for lower sampling frequencies is not as dramatic as for a CSMC. This difference can be assigned to the fact that DSMC specifically takes the sampling issue into account during the design procedure. Designing a CSMC for a sampled data system with a relatively low sampling frequency will most probably lead to extensive tuning of the controller parameters (switching gain and possibly the proportional gain and quasi sliding mode band) or even may even lead to a total redesign of the controller because the closed-loop system behaves different then expected. Already taking into account the sampling issue during the design phase can save precious time and costs. Another drive for the use of DSMC instead of CSMC could originate from the obtained model. Although models obtained from first principles will be continuous-time, identification will naturally lead to discrete-time models.

Special attention has been given in this thesis to the problem of target tracking, either of some desired output trajectory, some desired state trajectory, or some specified desired dynamical behaviour. DSMCs are developed for either case, both based on the output thereby using Output-Based DSMC (ODSMC) and based on the full state thereby using State-Based DSMC (SDSMC). For the generation of a feedforward control signal, and similarly the generation of a desired state trajectory,

leading to perfect tracking of the desired output signal (for the nominal system), a stable system inversion technique is presented. The cases of zeros on the unit circle, nonminimum phase zeros, and a limited preview of the desired output signal, are addressed as well.

8.2 Recommendations

In Chapter 7 a design study is presented for the speed- and air fuel ratio control of a spark ignition engine. Although the developed controllers indeed are SMCs, they are not based on the DSMC design techniques introduced in this thesis. The discrete-time implementation of the developed air fuel and speed controllers was not an issue considered in the design study. However, thinking of designing a DSMC for the air fuel and speed control leads to the following recommendations for future research:

- **Nonlinear control.** Like many control problems, the engine control problem in Chapter 7 is nonlinear. For those cases where a nonlinear, discrete-time, description of the system can be obtained, the DSMC design methodology should be extended to be capable to employ this knowledge. The same extension should be made for the feedforward controllers introduced in this thesis.

- **Variable sampling interval.** It was already explained in Chapter 7 that the air fuel ratio is in fact a discrete-time variable. It is only defined after the intake cycle has been completed. During the intake and exhaust cycles the air fuel ratio is not defined. Therefore, in some publications (e.g. (Powell et al. 1998)) a discrete-time model is obtained for the engine on which a discrete-time controller is defined. However, since the engine speed operates over a wide range, the time instances where the air fuel ratio is defined are not equidistant. Therefore, the sampling time should not be chosen fixed but should instead depend on the actual engine speed. In this way, the separate phases of the engine could be taken into account. The implications on the DSMC design of this varying sampling time should be studied to improve the applicability of DSMC to air fuel control.

- **Time delays.** It was explained in Chapter 7 that the air fuel ratio measurement has a, possible varying, time delay. Although some time delays can be incorporated in the design by augmenting the (discrete-time) system description with a few states, large- and especially varying time delays will be hard to incorporate in this way. Research has to be performed to find a solution to this problem.

- **Practical tests.** The designed air fuel ratio controller and engine speed controller should be tested in a practical environment. Despite the fact that the industrial simulation model describes the dynamics of a real life car engine very accurately, simulation never matches reality exactly. Therefore, practical applicability of the controllers should be tested on a real car engine.

Other recommendations, not specifically linked to the engine control problem, are:

- **Inter-sampling behaviour.** The discrete-time design procedures introduced in this thesis are all based on discrete-time models of a possibly continuous-time system. These models describe the behaviour of a continuous-time very well at the sampling instances. However, the behaviour of the system between the sampling instances, i.e. the inter-sampling behaviour, is not specified. Unfortunately, this approach does in general not suffice to guarantee that the closed-loop dynamics are satisfactory (see for example (Chen and Francis 1996)). Therefore, the inter-sampling behaviour should be studied.

- **Output-based sliding surface design.** The design of a output-based sliding surface is by no means trivial. Especially for large MIMO systems, this task will be difficult to perform. Recently, a Linear Matrix Inequality (LMI) based method has been developed (Edwards and Spurgeon 2001) which performs this task for OCSMCs in a far more easy way. The extension of this method to ODSMCs should be established.

- **Integrating performance criteria in the design procedure.** The sliding mode control design methodology principally divides the control design problem in two parts:

  - Sliding surface design which determines the dynamics in the sliding motion. with n the dimension of the state and m the number of inputs.
  - The dynamics towards the sliding surface as specified by the reaching law.

  The dynamics towards the sliding surface are typically chosen as fast as possible to approach the sliding surface with high accuracy. In the case that the sliding surface is designed such that each input is connected to exactly one entry of the switching function, these dynamics can be easily selected by hand because the problem consists of defining as many first order equations as there are inputs. The dynamics in the sliding mode are typically selected slower. For state-based SMCs they can most easily be designed by LQR design, for the output-based there is no standard routine to perform this (as the previous recommendation already stated). The disadvantage of the above sketched procedure is that there is no direct link between design requirements for the closed-loop system and the controller design. Instead, the design procedure for a sliding mode controller could be interpreted as try-and-error in the sense that the performance of the closed-loop system can be checked after the design has been completed. Therefore, future research should focus on the integration of design requirements into the design procedure.

- **Integral SMC** The design procedures for the SDSMCs and ODSMCs in this thesis do not incorporate any integrating action. However, for some applications this might be preferable. For this reason it should be investigated in which way the integrating action could be included. Possibilities are for example:

- Extending the original control law with the following integral control term:

$$u_i[k] = u_i[k_1] + K_i\sigma[k]$$

This extension could improve the convergence towards the sliding surface and reduce the quasi sliding mode band.

- Extend the switching function with n_{int} integral states, as:

$$\sigma[k] = Sx[k] + S_{int}x_{int}[k]$$

where the integral states $x_{int}[k]$ are for example defined by:

$$x_{int}[k] = x_{int}[k-1] + F_{int}x[k]$$

The matrix F_{int} can be designed to select states and/or outputs.

- **Higher order DSMC** Similar to the previous idea, discrete-time, higher order, DSMC should be considered. The design of ODSMCs is now constrained to systems where the matrix product CB was invertible. Permitting higher order DSMCs could circumvent this restriction.

Summary

This thesis is mainly devoted to the field of Sliding Mode Control (SMC). SMC has its roots in relay control. It originated in the Soviet Union in the late 1950s, but was not published outside the Soviet Union until the mid 1970s. The principle of SMC can be easily explained by considering the two design-steps. The first step of the design process is to define a new output, either based on the system state or based on the system output leading to State-based SMC (SSMC) and Output-based SMC (OSMC) respectively. This newly defined output is called the *switching function.* The hyperplane in the state-space where the switching function is zero is called the *sliding surface.* The construction of the switching function, and consequently the sliding surface, is not arbitrary and by no means trivial, and is presented in Chapter 2 using a linear continuous-time model. The second step of the design procedure is the construction of a control law which stabilizes the switching function at zero. The control law is derived from the so-called *reaching law* which specifies the movement towards the sliding surface. The reaching law is constructed using Lyapunov's stability theory which, together with the special structure of the switching function, leads to a discontinuous input signal. This discontinuity appears across the sliding surface, consequently the switching function *switches* the input signal which explains the terminology. Chapter 2 presents several controller structures which are based on this reaching law, again for linear continuous-time models. If the conditions set by the reaching law are satisfied, the closed-loop system reaches the sliding surface in finite time. During the phase of approaching the sliding surface the closed-loop system is said to be in the *reaching mode.* Once the sliding surface is reached, the closed-loop system will remain on the surface despite any disturbances. While on the sliding surface, the closed-loop system can only *slide* along the sliding surface in which case the closed-loop system is said to be in the *sliding mode.* It immediately follows that the order of the dynamics in the sliding mode are reduced by the number of components of the switching function, which is chosen equal to the dimension

of the input-vector. The direction of the movement along the sliding surface is determined by the design of the switching function. The switching function, and consequently the sliding surface, have to be constructed such that the system will slide along the sliding surface towards the origin of the state-space. In the case of a faulty design, the closed-loop system will slide away from the origin reflecting an unstable closed-loop system.

An important effect of the discontinuity at the sliding surface is that the input signal has to switch with extremely high frequency in the sliding mode. Although for some applications switching of the input signal is quite natural (for example switching electrical networks), switching is undesirable in some other applications due to for example high wear, high power dissipation, and excitation of unmodeled fast dynamics. In Chapter 2 three approaches are presented to reduce these problems. First of all, a continuous control term based on the available system knowledge, is added to the control signal, an approach which is often called the *equivalent control* approach. Furthermore, the discontinuous switching term can be made smooth by for example a saturation function, in which case the reaching law is only satisfied outside some small region around the sliding surface. Therefore the closed-loop system is only guaranteed to be driven into this, preferably small, region around the sliding surface denoted by the *quasi sliding mode band.* Finally, the amplitude of the discontinuous control term can be made variable and, if desired, adaptive.

The problem of a limited switching frequency for the input signal is even more apparent for a system under the control of a discrete-time controller. Clearly, an input value cannot be changed faster then the sampling frequency. If the sampling is performed much faster than the dynamics of the system under control, the influence of the limited switching frequency can be ignored. It is therefore a common approach to design sliding mode controllers in the continuous-time domain, even though the system is computer-controlled. For those cases where the sampling frequency does not justify such an assumption, Discrete-time SMC (DSMC) may be considered. In Chapter 3 the design procedure for a DSMC is presented which takes the discrete-time issue specifically into account. The design of the switching function can be performed along the same lines as for the CSMC. The design for the control-law however changes substantially. It is shown that the best performance can be achieved by omitting the discontinuous control term completely and solely using a linear controller, still under the assumption of the availability of a linear discrete-time model. Where a discontinuous reaching law (and consequently a discontinuous control law) leads to a switching control signal, a continuous reaching law leads to a smooth control signal thereby circumventing the problem of chattering. It is also shown that disturbance estimation, based on the switching function, can be employed to improve the accuracy of the closed-loop system. Estimating the disturbance using only the switching function, instead of the full system state, permits the application of this approach to both State-based DSMCs (SDSMCs) and Output-based SMCs (ODSMCs). A possible drawback of the proposed linear reaching law for DSMC is that it might lead to excessively large control inputs during the reaching mode. Therefore, a varying reaching law is presented to reduce this problem. If desired, it is still possible to include a discontinuous control term

in the DSMC. For those cases, the adaptive switching gain definitions available for CSMCs have been studied for their applicability to DSMCs. It is shown that those methods do contain some danger of instability, for which reason a new definition is introduced.

While the previous text was entirely focused on stabilization problems, the problem of target tracking is considered in this thesis as well. By target tracking it is meant that some desired state or output trajectory has to be replicated by the system under control. A desired state trajectory leading to perfect tracking of a specified desired output trajectory by the model can be constructed by model inversion. However, direct inversion of a nonminimum phase model leads to an unstable desired state signal. In Chapter 4, three inversion methods for linear discrete-time models are considered which can be employed to minimum phase models as well as nonminimum phase models. The first method, known as Steering Along the Zeros Control (SAZC), uses a transfer function description of the linear discrete-time model to compute the input signal which results in perfect tracking of the desired output trajectory by the model. By applying this input signal to the model we can compute the desired state trajectory. The second method, known as Stable Dynamic Inversion (SDI), augments the linear discrete-time state-space description of the system with an input model. A Kalman filter is constructed to observe the augmented state. In this thesis the input is modeled as integrated white noise, however other input models are possible. The third and last method, known as the Method of Dichotomies (MD), brings the linear discrete-time state-space description into Brunovsky canonical form. The unstable zero dynamics are then determined by solving them backwards in time while the stable zero dynamics are computed forward in time. These three methods in general all require a preview of the desired output trajectory. For the SAZC and the MD approaches, the preview time is in theory infinite. It is shown that only using a limited preview of the desired output trajectory, still an acceptable tracking can be achieved. The tracking performance of the MD approach can be further increased by the inclusion of a DSMC, as can be found in Chapter 4.

The desired state trajectory is now to be replicated by the system. The simplest way to perform this is to compute the input signal from the desired state trajectory and apply it as a feedforward control signal to the system. However, due to modeling errors and disturbances the tracking performance of the closed-loop system will decrease. Therefore, Chapter 5 defines several tracking controllers based on the earlier introduced DSMC techniques. These controllers aim at compensation of any unwanted deviations from the desired state or output trajectory. Both state-based and output-based controllers are presented. Also the problem where the desired closed-loop system dynamics are specified by means of a reference model is considered in Chapter 5. This approach is especially useful for those problems where a piece-wise constant desired output trajectory is specified. Perfect tracking of such an output trajectory is not feasible since it would involve extremely large control effort at the transitions. For these cases, the reference model will generate a smooth transition between the steady-states. Again a state-based and an output-based solution is presented using DSMC.

While earlier chapters include simulation examples to demonstrate the presented control theory, in Chapter 6 the presented control theory is applied to three laboratory test setups. First, the position of a cart fixed to a rail is controlled by a stabilizing SDSMC. Then a beam balancing on a fixed pole is positioned at specified pitch angles using a stabilizing SDSMC, a tracking SDSMC, and a model reference SDSMC. It follows from the tests that the tracking SDSMC and the model reference SDSMC have similar performances and perform better than the stabilizing SDSMC. The model reference controller is also implemented in continuous-time using SCSMC thereby neglecting the sampling issue. For a relatively high sampling frequency (100 Hz) the performance of the SDSMC and the SCSMC are comparable. However, for a relatively low sampling frequency (5 Hz) the SDSMC performs much better then the SCSMC. Finally, the stabilizing SDSMC, tracking SDSMC, and the model reference SDSMC are used to control the beam in both the pitch and yaw direction. The same conclusion can again be drawn, the performance of the tracking SDSMC and the model reference SDSMC are comparable and both perform better than the stabilizing SDSMC.
In Chapter 7 a design procedure for an air fuel ratio controller and a speed controller is presented for an automotive combustion engine. While both are based on the sliding mode methodology, their structure is different. The measurement of the air fuel ratio is delayed by a, possibly time varying, measurement delay. Therefore, the air fuel ratio controller is divided in a feedforward part and a feedback part. The feedforward part is based on the measured engine speed and manifold air pressure. These values are measured without a time delay, and therefore respond rapidly to throttle changes. The feedback part is based on the measured air fuel ratio and is used to reduce the steady-state error. The speed controller is designed by first transferring the model to the GCCF, from which the design of the controller easily follows. The derived controllers have been applied to an extensive nonlinear simulation model which incorporates many dynamics neglected in the controller design. The simulation results clearly demonstrate the applicability of the proposed controller structure. The air fuel ratio remains within $\pm 0.1\%$ of the stoichiometric value, thereby optimizing the catalytic efficiency. Only during fast transients the air fuel ratio momentarily leaves the $\pm 0.1\%$ error band around stoichiometry. Also the speed controller demonstrates good stabilization of the engine speed. During transients the engine speed does not perfectly match the reference trajectory. However, improving this tracking performance reduces the performance of the air fuel ratio controller thereby increasing air pollution drastically. The controller signals are smooth and therefore applicable to real-life engine control.
Despite the fact that for the design of the engine controllers no use have been made of the DSMC techniques introduced in Chapters 3 and 5, the study is very useful to identify some valuable future directions for research on DSMC. These recommendations, together with the conclusions, are presented in Chapter 8.

Samenvatting

Dit proefschrift gaat grotendeels over Sliding Mode Control (SMC). SMC komt voort uit de relaisbesturing. Het is ontstaan aan het einde van de vijftiger jaren, maar is pas buiten de Sovjet Unie gepubliceerd in het midden van de zeventiger jaren. Het principe van SMC laat zich het beste verklaren door de twee ontwerpstappen te beschouwen. In de eerste ontwerpstap wordt een nieuwe uitgang gedefinieerd op basis van de toestand of de uitgang van het systeem hetgeen leidt tot State-based SMC (SSMC) respectievelijk Output-based SMC (SSMC). De nieuw gedefinieerde uitgang noemt men de *switching function*. Het oppervlak in de toestandsruimte waar deze switching function nul is noemt men het *sliding surface*. Het ontwerp van de switching function, en dientengevolge de sliding surface, is zeker niet arbitrair of triviaal. Een ontwerpmethode hiervoor wordt beschreven in Hoofdstuk 2 waarbij gebruik wordt gemaakt van een lineair continue-tijd model. De tweede ontwerpstap bestaat uit het ontwerpen van een regelaar die het gesloten-lus systeem naar de sliding surface stuurt. Deze regelaar wordt ontworpen aan de hand van de zogenaamde *reaching law* welke de beweging naar de sliding surface specificeert. De reaching law is gebaseerd op de stabiliteittheorie van Lyapunov die door de speciale structuur van de switching function resulteert in een discontinue regelaar. Deze discontinuïteit treedt op bij het passeren van de sliding surface, men kan dus zeggen dat de switching function het ingangssignaal schakelt hetgeen de term "switching function" verklaart. In Hoofdstuk 2 worden, wederom voor lineaire continue tijd modellen, verschillende structuren van regelaars beschreven die gebaseerd zijn op de reaching law. Wanneer aan de condities van de reaching law voldaan wordt, zal het gesloten-lus systeem de sliding surface in eindige tijd bereiken. Men zegt dat het gesloten-lus systeem in de *reaching mode* verkeert wanneer deze zich naar de sliding surface toe begeeft. Wanneer de sliding surface eenmaal bereikt is zal het gesloten-lus systeem deze niet meer verlaten, ongeacht eventuele verstoringen. Het gesloten-lus systeem kan zich nu dus alleen nog over de sliding surface bewegen,

het systeem glijdt als het ware over de sliding surface hetgeen de term "sliding surface" verklaart. Wanneer het gesloten-lus systeem zich nog enkel langs de sliding surface beweegt dan zegt men dat het gesloten-lus systeem zich in de *sliding mode* bevindt. Aangezien de dimensie van de bewegingsruimte van het gesloten-lus systeem afgenomen is met de dimensie van de switching function kan men concluderen dat de orde van het gesloten-lus systeem afgenomen is. In welke richting het gesloten-lus systeem glijdt wordt bepaald door het ontwerp van de switching function. De switching function dient dusdanig ontworpen te worden dat het gesloten-lus systeem zich in de richting van de oorsprong in de toestandsruimte beweegt. Bij een foutief ontwerp zal het gesloten lus systeem zich van de oorsprong af bewegen, hetgeen een instabiel gedrag reflecteert.

Een direct gevolg van de discontinuïteit is dat het ingangssignaal in de sliding mode met extreem hoge frequentie moet gaan schakelen. Hoewel dit schakelend gedrag in sommige gevallen voortvloeit uit de applicatie (zoals bijvoorbeeld het geval is bij schakelende elektrische netwerken), zal dit gedrag voor sommige andere toepassingen ongewenst zijn in verband met bijvoorbeeld slijtage en excitatie van niet gemodelleerde snelle dynamica van het systeem. Daarnaast zal de schakelfrequentie begrensd worden door de maximale actuator snelheid. In Hoofdstuk 2 worden drie mogelijkheden gepresenteerd die deze problemen reduceren. Ten eerste kan men een continue regelcomponent toevoegen welke gebaseerd is op de beschikbare systeem kennis, een methode die wordt aangeduid met de term *equivalent control approach*. Verder kan men de discontinuïteit vervangen door een continue benadering zoals bijvoorbeeld de verzadigingsfunctie. Het gevolg hiervan is dat alleen aan de reaching law voldaan wordt buiten een bepaald gebied rond de sliding surface, welke de *quasi sliding mode band* genoemd wordt. Tenslotte kan men de amplitude van de discontinue regelcomponent variabel maken en, indien gewenst, adaptief.

Het probleem van een begrensde schakelfrequentie komt nog duidelijker naar voren bij computerbestuurde systemen. In deze gevallen kan men het ingangssignaal niet sneller schakelen dan de bemonsteringsfrequentie. Wanneer de bemonsteringsfrequentie vele malen sneller is dan de dynamica van het systeem dan zal het bemonsteren een beperkte invloed hebben. In de praktijk neemt men vaak aan dat dit het geval is en beschouwt men het gesloten-lus systeem als een volledig continue-tijd systeem. Echter, wanneer de bemonsteringsfrequentie een dergelijke verwaarlozing niet meer toelaat kan dit tot een degradatie van de prestaties leiden met in het extreme geval instabiliteit van het gesloten-lus systeem. In deze gevallen kan men het gebruik van Discrete-time SMC (DSMC) overwegen. In Hoofdstuk 3 wordt DSMC geïntroduceerd welke het bemonsteringsgedrag van het gesloten-lus systeem in de ontwerpprocedure meeneemt. De ontwerpprocedure van de switching function gaat volgens dezelfde wijze als bij CSMC. Het ontwerp van de regelaar wijkt echter af van de CSMC ontwerpprocedure. Hoofdstuk 3 toont aan dat het volledig achterwege laten van het discontinue gedrag van de regelaar tot het beste resultaat leidt. Daarnaast wordt aangetoond dat met het uitbreiden van de regelaar met een schatter van de verstoring, de prestaties nog verder verbeterd kunnen worden. Doordat de schatting van de verstoring enkel gebruik maakt van de switching function kan deze methode gebruikt worden voor zowel State-based DSMCs (SDSMC)

als voor Output-based DSMCs (ODSMC). Een mogelijk nadeel van de voorgestelde lineaire reaching law is het optreden van exorbitant grote ingangssignalen tijdens de reaching mode. Een variabele reaching law is voorgesteld welke dit probleem reduceert. Indien men toch gebruik wil maken van een discontinue regelaar besteedt Hoofdstuk 3 ook aandacht aan de discrete-tijd implementatie van de adaptieve regelaars die reeds in Hoofdstuk 2 aan bod kwamen. Aangetoond wordt dat deze in de discrete-tijd het gevaar tot instabiliteit hebben, daarom wordt een nieuwe definitie geïntroduceerd.

Hoewel de voorgaande tekst volledig op stabilisatieproblemen is toegespitst komt het probleem van target tracking ook aan bod in dit proefschrift. Met target tracking wordt bedoeld dat het systeem een voorgeschreven toestands- of uitgangssignaal moet volgen. Men kan d.m.v. systeeminversie een gewenst toestandssignaal bepalen welke tot het gewenste uitgangssignaal leidt. Deze procedure leidt echter tot instabiliteit bij niet-minimum fase modellen. In Hoofdstuk 4 worden drie methoden beschreven ter bepaling van het gewenste toestandssignaal en het bijbehorende gewenste ingangssignaal voor zowel minimum fase als niet-minimum fase modellen. De eerste methode, welke bekend staat onder de naam Steering Along the Zeros Control (SAZC), maakt gebruik van een lineaire, discrete-tijd transferfunctie beschrijving van het systeem. De methode levert het gewenste ingangssignaal waarmee het gewenste toestandssignaal berekend kan worden. De tweede methode, bekend onder de naam Stable Dynamic Inversion (SDI), breidt het toestandsmodel uit met een dynamisch model van het ingangssignaal. D.m.v. een Kalman filter wordt het gewenste toestands- en het ingangssignaal bepaald. In dit proefschrift wordt aangenomen dat het ingangssignaal te beschrijven is met een integrator met witte ruis als ingang, andere modellen zijn echter ook mogelijk. De derde en laatste methode, bekend onder de naam Methode of Dichotomies (MD), brengt de toestandsrepresentatie van het systeem in de Brunovsky canonieke vorm. De instabiele nuldynamica wordt dan bepaald door het terugrekenen in de tijd vanaf de bekend veronderstelde eindtoestand terwijl de stabiele nuldynamica op de gebruikelijke voorwaartse manier bepaald wordt. Het gewenste ingangssignaal kan aan de hand van het gewenste toestandssignaal bepaald worden. De drie methoden hebben gemeen dat zij het gewenste uitgangssignaal bekend veronderstellen in zowel het verleden als de toekomst. In Hoofdstuk 4 wordt aangetoond dat wanneer het gewenste uitgangssignaal slechts een aantal tijdstappen in de toekomst bekend is, het nog steeds mogelijk is om een bruikbare schatting van het gewenste ingangs- en toestandssignaal te maken. De gewenste toestandsschatting bij de MD kan in dit geval nog verder verbeterd worden wanneer men gebruik maakt van DSMC.

Nadat het gewenste toestandssignaal is bepaald is uiteraard de volgende stap om het systeem dit signaal te laten volgen. De meest eenvoudige methode is om het systeem te sturen met het gewenste ingangssignaal. Echter, door verstoringen, parameter variaties en niet gemodelleerde dynamica, zal het berekende gewenste ingangssignaal toch niet het gewenste effect leveren. Daartoe introduceert Hoofdstuk 5 verschillende regelaarstructuren welke gebaseerd zijn op de eerder beschreven DSMC technieken, die we tracking DSMCs zullen noemen. Daarnaast behandelt Hoofdstuk 5 ook het geval waar men de gewenste systeemdynamica beschrijft d.m.v.

een referentie model, i.p.v. het gewenste toestands- of uitgangssignaal voor te schrijven. Deze methode is bij uitstek geschikt voor gevallen waar het gewenste uitgangssignaal stapsgewijs constant is. Het perfect volgen van dergelijke signalen is niet mogelijk daar dit tot excessief grote ingangssignalen zou leiden. Voor zowel de tracking DSMC als de model referentie DSMC methoden worden oplossingen geboden gebaseerd op SDSMC en ODSMC.
In voorgaande hoofdstukken zijn simulatievoorbeelden opgenomen welke de werking van de aangeleverde theorie toelichten. In Hoofdstuk 6 worden de resultaten beschreven van de voorgestelde regelstrategieën toegepast op verschillende laboratoriumtestopstellingen. Eerst wordt de positie van een karretje op een rail geregeld d.m.v. een stabiliserende DSMC. Daarna worden de prestaties van een stabiliserende DSMC, een tracking DSMC en de model referentie DSMC met elkaar vergeleken voor het geval waar de hoek van een balancerende balk geregeld moet worden. De hoek van de balk wordt beïnvloed door een propeller die door de computer aangestuurd wordt. Uit de testen volgt dat de tracking DSMC en de model referentie DSMC ongeveer gelijke prestaties leveren welke duidelijk beter zijn dan de stabiliserende DSMC. Om een vergelijking tussen DSMC en CSMC te verkrijgen is de model referentie regelaar ook uitgevoerd met een CSMC. Zoals men intuïtief ook zou mogen verwachten, blijkt uit de testen dat de prestaties van de CSMC en de DSMC voor een relatief hoge bemonsteringsfrequentie (100 Hz) vergelijkbaar zijn terwijl de DSMC duidelijk beter dan de CSMC is voor een relatief lage bemonsteringsfrequentie (5 Hz). Tenslotte zijn de stabiliserende DSMC, de tracking DSMC en de model referentie DSMC ook gebruikt om de hoeken in twee dimensies van de balk te sturen. De resultaten zijn overeenkomstig de voorgaande testen: de tracking DSMC en de model referentie DSMC geven vergelijkbare resultaten welke duidelijk beter zijn dan de stabiliserende DSMC.
In Hoofdstuk 7 wordt een ontwerp gepresenteerd voor de besturing van verbrandingsmotoren in auto's. Het ontwerp bestaat uit een smoorklepregelaar, welke de snelheid van de motor regelt door de luchtstroming naar de cilinder te beïnvloeden, en een injectieregelaar die de brandstofstroming naar de cilinder regelt ten einde de zuurstof/brandstof verhouding in de cilinder op de gewenste waarde te houden. De verkregen regelaars worden getest d.m.v. een zeer uitgebreid simulatiemodel dat in de industrie gebruikt wordt. De simulatieresultaten laten zien dat de zuurstof/brandstof verhouding binnen $\pm$ 0.1 % van de gewenste waarde blijft. Ook de snelheid van de motor volgt de gewenste snelheid goed. Slechts tijdens snelle accelerates wordt de gewenste snelheid minder goed gevolgd. Dit probleem is op te lossen door de smoorklepregelaar sneller te maken hetgeen helaas gepaard gaat met grotere deviaties van de gewenste zuurstof/brandstof verhouding waardoor de katalysator efficiency dramatisch terug zal lopen.
Ondanks het feit dat voor het ontwerp van de regelaars voor verbrandingsmotoren geen gebruik gemaakt wordt van de in Hoofdstuk 3 en Hoofdstuk 5 geïntroduceerde DSMC technieken, motiveert deze ontwerpstudie wel mogelijk toekomstig onderzoek op het gebied van DSMC. Een opsomming van mogelijk toekomstig onderzoek op het gebied van DSMC, alsmede de conclusies die op basis van dit proefschrift getrokken kunnen worden, zijn te vinden in Hoofdstuk 8.

Bibliography

Aly, G. M. and W. G. All (1990), "Digital design of variable structure control systems", *International Journal of Systems Science* **21**, 1709–1720.

Aquino, C. F. (1981), Transient A/F control characteristics of the 5 liter central fuel injection system, *in* "SAE paper no. 810494".

Åstrom, K.J. and B. Wittenmark (1997), *Computer-Controlled Systems, Theory and Design*, Prentice Hall information and System Science Series, 3rd edn, Prentice Hall.

Balluchi, A., L. Benvenuti, M.D. Benedetto, S. Cardellino, C. Rossi and A. Sangiovanni-Vincentenlli (1999), "Hybrid control of the air-fuel ratio in force transients for multi-point injection engines", *Proceeding of the 38th Conference on Decision and Control* pp. 316–320.

Balluchi, A., M.D. Benedetto, C. Pinello, C. Rossi and A. Sangiovanni-Vincentenlli (1999), "Hybrid control in automobile applications: the cut-off control", *Automatica* **35**(3), 519–535.

Bartolini, G., A Ferrara and E. Usai (1998), "Chattering avoidance by second-order sliding mode control", *IEEE Transactions on Automatic Control* **43**(2), 241–246.

Bartolini, G., A. Ferrara and V. Utkin (1995), "Adaptive sliding mode control in discrete-time systems", *Automatica* **32**(5), 796–773.

Bartoszewicz, A. (1996), "Remarks on discrete-time variable structure control systems", *IEEE Transactions on Industrial Electronics* **43**, 235–238.

Bartoszewicz, A. (1998), "Discrete-time quasi-sliding-mode control strategies", *IEEE Transactions on Industrial Electronics* **45**, 633–637.

Beaumont, A. J., A. D. Noble and A. Scarisbrick (1992), "Adaptive transient air-fuel ration control to minimize gasoline engine emissions", *Proceedings of the 1992 FISITA Congress* .

Bondarev, A. G., S. A. Bondarev, N. E. Kostyleva and V. I. Utkin (1985), "Sliding modes in systems with asymptotic state observers", *Automation and Remote Control* **46**(6), 49–64.

Canale, R. and et. al. (1978), "General motors phase II catalyst systems", *SAE Paper No. 780205* .

Chan, C. Y. (1999), "Discrete adaptive sliding-mode control of a class of stochastic systems", *Automatica* **35**, 1491–1498.

Chang, C. F., N. P. Fekete and J. D. Powell (1993), "Engine air-fuel ratio control using an event based observer", *SAE Paper no. 930766* .

Chen, T. and B.A. Francis (1996), *Optimal Sampled-Data Control Systems*, Communications and Control Engineering Series, 2nd edn, Springer-Verlag, London.

Cho, D. (1991), "Research and development needs for engine and power train control systems", *ASME Advanced Automotive Technologies* **40**, 23–34.

Cho, D. and J. K. Hedrick (1988), "A nonlinear controller design method for fuel-injected automotive engines", *Journal of Engineering for Gas turbines and Power, transactions of the ASME* **110**, 313–320.

Cho, D. and J. K. Hedrick (1991), "Sliding mode fuel-injection controller: Its advantages", *Journal of Dynamic Systems, Measurement, and Control* **113**, 537–541.

Choi, S-B. and J.K. Hedrick (1996), "Robust throttle control of automotive engines: theory and experiment", *ASME Journal of Dynamic Systems, Measurement and Control* **118**, 92–98.

Cipollone, R. and M. Sughayyer (2001), "A new modeling of air dynamics inside ICE intake manifolds", *Proceedings of the 7th International Conference & Exhibition, Firenze ATA* .

Crossley, P.R. and J.A. Cook (1991), "A nonlinear engine model for drive train system development", *Proceedings of the IEE International Conference on Control* pp. 921–925.

DeCarlo, R. A., S. H. Żak and G. P. Matthews (1988), "Variable structure control of nonlinear multivariable systems: A tutorial", *Proceedings of the IEEE* **76**(3), 212–232.

Devasia, S. (1997), "Output tracking with nonhyperbolic and near nonhyperbolic internal dynamics: Helicopter hover control", *AIAA Journal of Guidance, Control, and Dynamics* **20**(3), 573–580.

Devasia, S., D. Chen and B. Paden (1996), "Nonlinear inversion-based output tracking", *IEEE Transactions on Automatic Control* **41**(7), 930–942.

Edwards, C. and S. K. Spurgeon (1998), *Sliding Mode Control, Theory and Applications*, Vol. 7 of *Systems and Control Book Series*, Taylor & Francis Ltd.

Edwards, C. and S. K. Spurgeon (2001), "On the design of switching functions for sliding mode output feedback controllers using LMIs", *Proceedings of the European Control Conference* pp. 979–984.

Falk, C. D. and J. J. Mooney (1980), "Three-way conversion catalyst: Effects of closed-loop feedback control and other parameters on catalyst efficiency", *SAE Paper N0. 800462* .

Filippov, A. F. (1964), "Differential equations with discontinuous right hand-sides", *American Mathematical Society Translations* **42**, 199–231.

Fliess, M., H. Sira-Ramírez and R. Marquez (1998), "Regulation of nonminimum phase outputs: a flatness based approach", *In D. Normand-Cyrot (ed), Perspectives in Control - Theory and Applications: a Tribute to Ioan Doré Landau, London, Springer* pp. 143–164.

Furuta, K. (1990), "Sliding mode control of a discrete system", *Systems & Control Letters* **14**, 145–152.

Gao, W., Y. Wang and A. Homaifa (1995), "Discrete-time variable structure control systems", *IEEE Transactions on Industrial Electronics* **42**(2), 117 122.

George, K., M. Verhaegen and J. M. A. Scherpen (1999*a*), "Stable inversion of MIMO linear discrete time nonminimum phase systems", *Proceedings of the 7th IEEE Mediterranean Conference on Control and Automation, Haifa, Israel* pp. 267–281.

George, K., M. Verhaegen and J. M. A. Scherpen (1999*b*), "A systematic and numerically efficient procedure for stable dynamic model inversion of LTI systems", *Proceedings of the 38th IEEE Conference on Decision and Control* pp. 1881–1886.

Ghosh, J. and B. Paden (2001), "Iterative learning control for nonlinear nonminimum phase plants", *ASME Journal of Dynamic Systems, Measurement, and Control* **123**, 21–30.

Golo, G. and C. Milosavljević (2000), "Robust discrete-time chattering free sliding mode control", *Systems & Control Letters* **41**, 19–28.

Gross, E. and M. Tomizuka (1994), "Experimental flexible beam tip tracking control with a truncated series approximation to uncancelable inverse dynamics", *IEEE Transactions on Control Systems Technology* **2**(4), 382–391.

Haskara, İ., Ü Özgüner and V. Utkin (1998), "On sliding mode observers via equivalent control approach", *International Jountrol of Control* **71**(6), 1051–1067.

Hendricks, E. and C. Sorenson (1990), "Mean value modeling of spark ignition engines", *SAE* .

Heywood, J. B. (1988), *Internal Combustion Engine Fundamentals*, McGraw Hill, New York.

Hires, S. D. and M. T. Overington (1981), Transient mixture strength excursions - an investigation of their causes and the development of a constant mixture strength fuelling strategy, *in* "SAE 810495".

Holmberg, U., P. Myszkorowski, Y. Piguet and R. Longchamp (1995), "On compensation of nonminimum-phase zeros", *Automatica* **31**(10), 1433–1441.

Hou, M. and R. J. Patton (1998), "Optimal filtering for systems with unknown inputs", *IEEE Transactions on Automatic Control* **43**(3), 445–449.

Hui, S. and S. H. Żak (1999), "On discrete-time variable structure sliding mode control", *Systems & Control Letters* **38**, 283–288.

Hung, J. Y., W. Gao and J. C. Hung (1993), "Variable structure control: A survey", *IEEE Transactions on Industrial Electronics* **Vol. 40**, 2–21.

Iordanou, H. N. and B. W. Surgenor (1997), "Experimental evaluation of the robustness of discrete sliding mode control versus linear quadratic control", *IEEE Transactions on Control Systems Technology* **5**(2), 254–260.

Isidori, A. (1995), *Nonlinear Control Systems*, 3 edn, Springer-Verlag, London.

Itkis, U. (1976), *Control Systems of Variable Structure*, Wiley, New York.

Jiang, Y. A., D. J. Clements, T. Hesketh and J. S. Park (1994), "Adaptive learning control of robot manipulators in task space", *Proceedings of the American Control Conference* pp. 207–211.

Jung, S. L. and Y. Y. Tzou (1996), "Discrete sliding mode control of a PWM inverter for sinusoidal output waveform synthesis with optimal sliding curve", *IEEE Transactions on Power Electronics* **11**, 567–577.

Kaynak, O. and A. Denker (1993), "Discrete-time sliding mode control in the presence of system uncertainty", *International Journal of Control* **57**, 1177–1189.

Kim, Y-W., G. Rizzoni and V. Utkin (1998), "Automotive engine diagnosis and control via nonlinear estimation", *IEEE Control Systems* **18**(5), 84–99.

Lenz, H., R. Berstecher and M. K. Lang (1998), Adaptive sliding-mode control of the absolute gain, *in* "Nonlinear Control Systems Design Symposium 1998", pp. 667–672.

Leung, T., Q. Zhou and C. Su (1991), "An adaptive variable structure model following control design for robot manipulators", *IEEE Transactions on Automatic Control* **36**(3), 347–352.

Levant, A. (1997), "Higher order sliding: collection of design tools", *Proceedings of the 4th European Control Conference* .

Lu, X.Y. and S.K. Spurgeon (1997), "Robust sliding mode control of uncertain nonlinear systems", *Systems and Control Letters* **32**, 75–90.

Lu, X.Y. and S.K. Spurgeon (1998), "Output feedback stabilisation of siso nonlinear systems via dynamic sliding modes", *International Journal of Control* **70**(5), 735–759.

Lu, X.Y. and S.K. Spurgeon (1999*a*), "Control of nonlinear non-minimum phase systems using dynamic sliding mode", *International Journal of Systems Science* **30**(2), 183–198.

Lu, X.Y. and S.K. Spurgeon (1999*b*), "Output feedback stabilisation of mimo nonlinear systems via dynamic sliding mode", *International Journal of Robust and Nonlinear Control* **9**, 275 306.

Majors, M., J. Stori and D. Cho (1994), "Neural network control of automotive fuel-injection systems", *IEEE Control Systems* **14**(3), 32 36.

Marconi, L., G. Marro and C. Melchiorri (2001), "A solution for almost perfect tracking of nonminimum-phase, discrete-time linear systems", *International Journal of Control* **74**(5), 496–506.

Marro, G. (1996), "Multivariable regulation in geometric terms: old and new results", *C. Bonivento, G. Marro and R. Zanasi (Eds) Colloquium on Automatic Control* .

Marro, G. and F. Hamano (1996), "Using preactuation to eliminate tracking error in feedback control of nonminimum-phase systems", *Proceedings of the IEEE Conference on Decision and Control, Kobe, Japan* **4**, 4549–4551.

Marro, G. and L. Marconi (1997), "Using the diophantine equation in the design of a digital perfect tracking compensator", *Proceedings of the 16th Benelux Meeting on Systems and Control* .

Milosavljević, C. (1985), "General conditions for the existence of a quasi-sliding mode on the switching hyperplane in discrete variable structure systems", *Automation and Remote Control* **3**, 36–44.

Monsees, G. and J. M. A. Scherpen (2000), "Adaptive switching gain for a discrete-time sliding mode controller", *Proceedings of the American Control Conference, Chicago, U.S.A.* pp. 1639–1643.

Monsees, G. and J. M. A. Scherpen (2001*a*), "Discrete-time output-based target tracking using sliding modes", *Submitted* .

Monsees, G. and J. M. A. Scherpen (2001*b*), "Discrete-time sliding mode control with a disturbance estimator", *Proceedings of the European Control Conference, Porto, Portugal* pp. 3270–3275.

Monsees, G. and J. M. A. Scherpen (2001*c*), "Discrete-time sliding mode control with disturbance estimation", *Submitted* .

Monsees, G. and J. M. A. Scherpen (2001*d*), "Output tracking using a discrete-time sliding mode controller with reduced-order state-error estimation", *Proceedings of the 1st IFAC Symposium on System Structure and Control, Prague, Czech Republic.* .

Monsees, G. and J. M. A. Scherpen (2002*a*), "Adaptive switching gain for a discrete-time sliding mode controller", *International Journal of Control* **75**(4), 242–251.

Monsees, G. and J. M. A. Scherpen (2002*b*), "Nonminimum phase output tracking using sliding modes", *To appear in Nonlinear and Adaptive Control, Editors A. Zinober and F. Labnabhi-Lagarrique, Springer Verlag* .

Monsees, G., K. George, J. M. A. Scherpen and M. Verhaegen (1999), "A feedforward-feedback interpretation of a sliding mode control law", *Proceedings of the 7th IEEE Mediterranean Conference on Control and Automation, Haifa, Israel* pp. 2384–2398.

Monsees, G., S. K. Spurgeon and P. F. Puleston (2000), "A combined feedforward/feedback approach to air/fuel control using sliding modes", *Proceedings of the International Conference on Systems Engineering, Coventry, U.K.* .

Monsees, G., S. K. Spurgeon and P. F. Puleston (2001), "Air fuel ratio control using sliding modes", *Systems Science* **26**(3), 97–108.

Moskwa, J. J. and J. K. Hedrick (1990), "Nonlinear algorithms for automotive engine control", *IEEE Control Systems* **10**(3), 88–93.

Pieper, J. K. and R. Mehrotra (1999), "Air/fuel control using sliding mode methods", *Proceedings of the American Control Conference* **1**, 1027–1031.

Pieper, J. K., S. Baillie and K. R. Goheen (1996), "Linear quadratic optimal model-following control of a helicopter in hover", *Optimal Control Applications and Methods* **17**, 123–140.

Powell, J. D., N. P. Fekete and C-F. Chang (1998), "Observer-based air-fuel ratio control", *IEEE Control Systems* **18**(5), 72–83.

Puleston, P. F., G. Monsees and S. K. Spurgeon (2002), "Air/fuel ratio and speed control for low emission vehicles based on sliding mode techniques", *To appear in the Journal of System and Control Engineering* .

Puleston, P. F., S. K. Spurgeon and G. Monsees (2001), "Automotive engine speed control: A robust nonlinear control framework", *IEE Proceedings - Control Theory and Applications* **148**(1), 81–87.

Roh, Y. and J. Oh (2000), "Sliding mode control with uncertainty adaption for uncertain input-delay systems", *Proceedigs of the American Control Conference* pp. 636–640.

Sarpturk, S. Z., Y. Istefanopulos and O. Kaynak (1987), "On the stability of discrete-time sliding mode control systems", *IEEE Transactions on Automatic Control* **32**(10), 930–932.

Scherpen, J. M. A., D. Jeltsema and B. Klaassens (2000), "Langrangian modeling and control of switching networks with integrated coupled magnetics", *Proceedings of the 39th IEEE Conference on Decision and Control* pp. 4054–4059.

Sha, D. and V. B. Bajić (2000), "Robust discrete adaptive input-output-based sliding mode controller", *International Jounal of Systems Science* **31**(12), 1601 1614.

Shim, D., J. Park, P. Khargonekar and W. Ribbens (1996), "Reducing automotive engine speed fluctuation at idle", *IEEE Transactions On Control Systems Technology* **4**(4), 404–410.

Shiraishi, H., S. L. Ipri and D. Cho (1995), "CMAC nearal network controller for fuel-injection systems", *IEEE Transactions on control systems technology* **3**(1), 32–38.

Sira-Ramírez, H. (1991), "Non-linear discrete variable structure systems in quasi-sliding mode", *International Jounal of Control* **54**, 1171–1187.

Sira-Ramírez, H. (1993), "A dynamical variable structure control strategy in asymptotic output tracking problems", *IEEE Transactions on Automatic Control* **38**(4), 615–620.

Sira-Ramírez, H. (1999), "On the control of the variable length pendulum", *Proceedings of the Conference on Decision and Control* pp. 1188–1189.

Slotine, J. J. E. (1984), "Sliding controller design for non-linear systems", *International Journal of Control* **40**(2), 421 434.

Slotine, J. J. E. and W. Li (1991), *Applied Nonlinear Control*, Prentice Hall.

Souverijns, W. and D. Vaes (2001), "Ontwerp van een regelaar voor multi-variabele niet-lineare systemen", *Masters Thesis, Katholieke Universiteit Leuven* .

Spurgeon, S. K. (1992), "Hyperplane design techniques for discrete-time variable structure control systems", *International Journal on Control* **55**(2), 445–456.

Stotsky, A., B. Egardt and S. Eriksson (2000), "Variable structure control of engine idle speed with estimation of unmeasurable disturbances", *ASME Jounal of Dynamic Systems, measurement, and Control* **122**, 599 603.

Su, C., Q. Zhou and T. Leung (1991), "Adaptive sliding mode control of constrained robot manipulators", *Proceedings of the 30th conference on Decision and Control* pp. 1382–1384.

Su, W., S. V. Drakunov and Ü. Özgüner (2000), "An $O(T^2)$ boundary layer in sliding mode for sampled-data systems", *IEEE Transactions on Automatic Control* **45**(3), 482–485.

Trentini, M. and J. K. Pieper (2001), "Mixed norm control of a helicopter", *AIAA Journal of Guidance, Control, and Dynamics* **24**(3), 555–565.

Utkin, V. (1977), "Variable structure systems with sliding mode", *IEEE Transactions on Automatic Control* **22**(2), 212–222.

Utkin, V. (1992), *Sliding Modes in Control Optimization*, Communications and Control Engineering Series, Springer-Verlag, London.

Utkin, V., J. Guldner and J. Shi (1999), *Sliding Mode Control in Electromechanical Systems*, Taylor & Francis Ltd.

Vesterholm, T. and E. Hendricks (1994), "Advanced nonlinear engine speed control systems", *Proceedings of the American Control Conference* **2**, 1579–1580.

Wang, W. and Y. Fan (1993), "Output feedback in variable structure systems with a simple adaption law", *Proceedings of the 32nd Conference on Decision and Control* pp. 422–423.

Wheeler, G., C. Su and Y. Stepanenko (1998), "A sliding mode controller with improved adaption laws for the upper bounds on the norm of uncertainties", *Automatica* **34**(12), 1657–1661.

Won, M. and J. K. Hedrick (2001), "Disturbance adaptive discrete-time sliding control with application to engine speed control", *ASME Journal of Dynamic Systems, Measurement, and Control* **123**, 1–9.

Won, M., S. N. Choi and J. K. Hedrick (1998), "Air-to-fuel ratio control of spark ignition engines using gaussian network sliding control", *IEEE Transactions on Control Systems Technology* **6**(5), 678–687.

Xia, J. Z. and C. H. Menq (1995), "Precision tracking control of nonminimum phase systems with zero phase error", *International Journal of Control* **61**(4), 791–807.

Young, K. D. and Ü. Özgüner (1999), "Sliding mode: Control engineering in practice", *Proceedings of the American Control Conference* pp. 150–162.

Acknowledgements

Despite my initial reluctance to become a PhD student some people motivated me to write this thesis. I therefore especially owe my gratitude to Jacquelien Scherpen for being both patient and inspiring enough to keep me going during the years and for reading my chapters over and over again searching for the numerous mayor and minor errors. Also many thanks to Michel Verhaegen for his inspiration and almost infinite number of ideas. I would also like to thank Sarah Spurgeon for the opportunity to stay six months in Leicester and her support in the progress of this thesis. The discussions with Paul Puleston, Koshy George, and Joris de Cuyper played an important role in the development of my knowledge and consequently the book that is in front of you. Also many thanks to Robert Babuška for the discussions and Johannes Hidajat for the laboratory work. Furthermore, I would like to mention Chris Edwards, Matthew Turner and Guido Hermann for helping me to get settled down in Leicester and forgiving me all those offensive remarks concerning the British railway system.

Those people who did not have a direct input in the scientific contents of this thesis played an immense role as well. First of all I would like to thank Ger Honderd who gave me my first position in Delft. But, of course even more important, De Biercommissie! An anarchy formed by C.L. Wams, C.L. Lazeroms, C.L. Bloemen, C.L. Osterom, C.L. Jeltsema, and fortunately myself. It has been an honor to be a member of this team, an honor which is luckily everlasting. Closely connected, our "Fanclub" formed by De Vrienden van de Biercommissie: Vr. Babuška, Vr. Roubos, Vr. Hegyi, Vr. Duindam, and Vr. Keukens. I think it has been a missed opportunity for Rob de Vries, Ben Klaassens, Jacquelien Scherpen, and Kitty Dukker for not having obtained the title Vr. yet, you would indeed have earned it! I would also like to mention De Feestcommissie, formed by Bart Wams, Marcel Oosterom and myself. It has been an enormous amount of fun to organize the two *little department's onions*, walking around in a silver tuxedo and a traditional golf outfit, rebuilding

the 5th floor into a six hole golf course, and especially generating those few-thousand ideas which, in the end, where just that little bit too "special" to bring into practice. Finally, my friends and family further away from the scientific world, but just as important for the person I am and for the things I have done. My gratitude to Carlo Verleun and Marco Joosten for being best friends for the past thirty-something years. Many thanks to Ruud Doggen and John van Nijnatten for those evenings which certainly did not improve the contents of this thesis! The COM-group, what's left of it, for adopting me. Anko Gils for the organization of all those sailing weekends. Barbara Muijtjens, Cor Miltenburg, Erik Manneke, Henry de Kuijer, Jasper Braakhuis, Peter Glas, Rutger Bevaart, Stef Loffeld, Toby Nanne, and Vincent Verdult for not loosing contact after my numerous schools. Frutsel for being so eager to be in this list and van Delft for being a "lelluken droak". Furthermore, my family and in particular my father for the beautiful drawings which certainly enlightened the appearance of this thesis.
Besides all those important people, I feel truly privileged with my paranimfen Bart Wams and Marco Joosten.
Last but certainly not least Liesbeth. Thank you for stepping back from the flashing world of advertisement into my world, for sharing the joy of living on a beautiful boat in Amsterdam, for the enormous amount of countries we have seen together, and all those other tiny little things which truly make a difference.

Curriculum Vitae

Govert Monsees was born on the 4th of may 1970, in Uden, The Netherlands. He received the B.Sc degree in Avionics from the Algemene Hogeschool Amsterdam, The Netherlands, in 1993, and in 1997 the M.Sc degree in Electrical Engineering from the Delft University of Technology, Delft, The Netherlands.
In the period between 1997 and 2001 he has mainly worked at the Control Laboratory at the Delft University of Technology, with intermezzo's at ITM cv (as part of a design team), University of Leicester (as a research associate), and South America and Asia (as a backpacker). He has been an active member of De Biercommissie since 1998. Since March 2002 he has been working at Tebodin Consultants & Engineers.
His interest include traveling, sailing, playing saxophone and guitar, squash, and photography. He lives together with Liesbeth on the house boat "De Kleine Beer" in Amsterdam, The Netherlands.