Modern tools for control of nonlinear systems

Johan Hamberg, Markus Högberg, Ulrik Nilsson,
John W.C. Robinson, Daniel Skoogh

FOI-R--1696--SE
ISSN-1650-1942

Systems Technology
Methodology report

September 2005

Modern tools for control of nonlinear systems

Johan Hamberg, Markus Högberg, Ulrik Nilsson,
John W.C. Robinson, Daniel Skoogh

FOI-R--1696--SE
ISSN-1650-1942

Systems Technology
Methodology report

September 2005

Issuing organisation	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Systems Technology SE-164 90 STOCKHOLM	FOI-R--1696--SE	Methodology report
	Month year September 2005	**Project number** I60123
	Research area code Strike and Protection	
	Sub area code Weapons and Protection	
	Sub area code 2	

Author(s)	
Johan Hamberg, Markus Högberg, Ulrik Nilsson, John W.C. Robinson, Daniel Skoogh	**Project manager** Markus Högberg
	Approved by Monica Dahlén Division Head, Systems Technology
	Scientifically and technically responsible Martin Hagström
	Sponsoring agency Intern

Report title

Modern tools for control of nonlinear systems

Abstract

In military as well as industrial applications there are complex machines with a dedicated task to perform. Substantial benefits can be drawn from improving the performance and efficiency of these machines, and the modern tools of nonlinear control theory can make this possible in many applications. Regardles of the purpose of the system to be controlled, there are three main ingredients that are needed. A good model of the complete system including uncertanties, a good and accurate measure of the objectives to achieve and a tool for control synthesis and system modeling and analysis.

This report presents advances in methods for model order reduction, which is a key ingredient in the modeling phase of controller design for complex systems. Further a number of synthesis methods for stability and tracking, including nonlinear dynamic inversion (NDI), sliding modes and backstepping, are reviewed along with a simple example. A new result on tracking properties for the zero-dynamics of a backstepping controller are also included. A brief introduction to optimisation based control including a problem formulation for open-loop robust control and dynamic games follows. The final chapter presents an example of control law design based on the powerful techniques for block/vector backstepping recently introduced in the literature. Utilizing these techniques it is possible to quickly synthesize a controller with desirable properties for simultaneous control in pitch, yaw and roll which is illustrated for a realistic model of a fighter aircraft.

Keywords

Nonlinear systems, control, backstepping, NDI, flight control

Further bibliographic information

ISSN	Pages	Language
ISSN-1650-1942	52	English
Distribution	**Price**	
By sendlist	According to price list	
	Security classification	
	Unclassified	

Utgivare

Totalförsvarets Forskningsinstitut – FOI
Systemteknik
164 90 STOCKHOLM

Rapportnummer, ISRN

FOI-R--1696--SE

Klassificering

Metodrapport

Månad år

September 2005

Projektnummer

I60123

Forskningsområde

Bekämpning och skydd

Delområde

VVS med styrda vapen

Delområde 2

Författare

Johan Hamberg, Markus Högberg, Ulrik Nilsson,
John W.C. Robinson, Daniel Skoogh

Projektledare

Markus Högberg

Godkänd av

Monica Dahlén
Avdelningschef, Systemteknik

Tekniskt och/eller vetenskapligt ansvarig

Martin Hagström

Uppdragsgivare/kundbeteckning

Intern

Rapporttitel

Moderna metoder för styrning av ickelinjära system

Sammanfattning

I militära såväl som industiella tillämpningar finns komplexa maskiner med en specifik uppgift. En förbättring av prestanda och effektivitet hos dessa maskiner kan innebära avsevärda fördelar, och de verktyg som tilhandahålls genom modern styrteori för ickelinjära system möjliggör sådana förbättringar i många fall. Oavsett vad syftet är med systemet som skall styras så behövs huvudsakligen tre ingredienser. En bra och noggrann modell av det kompletta systemet inklusive osäkerheter, ett bra och korrekt mått på hur systemet presterar i förhållande till målet och slutligen verktyg för syntes av styrlagar, systemmodellering och analys.

Denna rapport beskriver framsteg inom modellreduktion, vilket är en nyckelingediens i modellerings fasen då en styrlag skall tas fram för ett komplext system. Vidare så presenteras kort ett antal syntesmetoder för stabilisering och följning, bland annat “backstepping” och “nonlinear dynamic inversion (NDI)” med enkla exempel. Dessutom innehåller rapporten nya resultat angående nolldynamik och följningsegenskaper för backstepping regulatorer. En kort introduktion till optimeringsbaserade metoder för styrning följs av problemformuleringar för robust styrning och dynamiska spel. I det sista kapitlet presenteras ett exempel på styrlagsdesign baserat på en utveckling av de kraftfulla tekniker för block/vektorbackstepping som nyligen introducerats i litteraturen. Med dessa tekniker kan man snabbt ta fram en regulator med goda egenskaper för samtidig reglering i roll-, tipp och girled, vilket illustreras för en realistisk modell av ett jaktflygplan.

Nyckelord

Ickelinjära system, styrning, backstepping, NDI, flygstyrsystem

Övriga bibliografiska uppgifter

ISSN

ISSN-1650-1942

Antal sidor

52

Språk

Engelska

Distribution

Enligt missiv

Pris

Enligt prislista

Sekretess

Öppen

Contents

1 Introduction

1.1 The project

In 2002 a strategic competence project in the area of control systems was initiated at the Division of Systems Technology of the Swedish Defense Research Agency (FOI). The project has been carried through at the department of Autonomous Systems during two years. The purpose has been to strengthen the competence within the department and to increase the knowledge of modern techniques used for application of linear as well as nonlinear control theory and systems analysis.

In this report the focus is to highlight a few modern methods for control synthesis that are mature and have a potential for large scale industrial application. Starting of with a glance at the history of the research area, a few fundamental theories and their potential and limitations are discussed. In the last chapter an example of application of a modern synthesis method for flight control is given.

1.2 The research area

1.2.1 Origin

Extremely advanced control systems have been around since before mankind, and still the state of the art of man made control systems utilizing the most advanced technology is not even in the same ballpark as those produced through evolution. Controlling a machine as complex as the human body with the agility of an acrobat is a today only a dream of control engineers and scientists.

Early efforts in the field were motivated by practical problems requiring skillful engineering solutions. Skipping through history until mathematics was introduced as a tool for control system analysis and design, we end up in the middle of the 19th century. Control theory is thus a fairly young field of research focused on system analysis and modelling for the purpose of control design. This admittedly narrow definition distinguishes the field from pure mathematics through specification of an application, like any other area of research in applied mathematics. A characteristic feature of control theory is that it is by nature an interdisciplinary field where problems from e.g. biology, chemistry, physics, economics and mechanics are formulated to fit in a standard framework for which the tools of control theory are developed. The aim of this section is to provide a very coarse grain overview of the research area. Note that many active areas that have had significant impact on the evolution of the field are not considered in this brief review e.g. digital control, system identification, control of distributed and infinite dimensional systems and adaptive control to name a few.

1.2.2 The Classical era

Initially the focus of analysis of control systems was the stability of differential equations. Through the work of Maxwell, utilizing linearized differential equations to find characteristic equations and analyzing the roots, control theory is considered to be established as a discipline. Routh and Hurwitz developed methods to determine the stability of linear systems. In Russia, Lyapunov analyzed the stability of the nonlinear differential equations using methods that are the foundation of many theories in practical use today.

The classical engineering tools of early control theory are basically graphical criteria for single-input single-output (SISO) systems in the frequency domain used to determine stability of a linear time invariant (LTI) system. The mathematical tool of the classical methods is complex analysis. The methods of Evans, Bode, Nyquist and Nichols are still widely used due to their simplicity and have the advantage of providing low order stabilizing and robust controllers. There are however some limitations of frequency domain methods that make then unsuitable for multiple-input multiple-output (MIMO), time varying and nonlinear systems. Another issue is the lack of systematic methods for obtaining a specified system performance.

1.2.3 Linear state space methods

Modern (or state space based) control theory grew in popularity through successful research in the early 1950's. With these methods the focus shifted towards designing for optimal performance of the controlled (hopefully stable) system. The linear state space synthesis methods have capitalized on advances in linear algebra as well as numerical methods for solving large scale matrix equations. The state space methods allow control designs to achieve optimal performance given a specified objective also for MIMO systems. Tools for stability analysis are often derived from the stability theorems of Lyapunov.

One invention that has been very successful in applications is the Kalman filter, which provide the optimal state estimate based on measurements of the state, even when influenced by noise. One problem is however that the controllers obtained from state space formulations are often of high order and thus not as easily accommodated in applications as those designed using the classical approach. This has motivated efforts in model and, more recently, controller reduction which is the topic of a later section in this report.

A standard form for linear systems of equations has made it possible to develop toolboxes to solve problems of system analysis and control synthesis. The state space formulations enable the use of powerful analysis tools from linear algebra e.g. for determining if it is possible to control/ stabilize or observe/ detect a (linear) system given a set of inputs or outputs. In the last decades researchers have explored linear fractional transformations (LFT) as a standard form of the system equations to enable development of more advanced algorithms and toolboxes for linear systems. Linear parameter varying (LPV) system models are sometimes used to extend the region of validity of a linear model while retaining the simple system description.

1.2.4 Modern robust control

The need for robustness in controlled systems inspired scientists to combine the robustness properties of classical methods with the power of the state space methods. Efforts on extending classical frequency domain methods to MIMO systems was successful in the mid 70's. One example of a theory in this branch is the Quantitative Feedback Theory (QFT) utilizing an extension of the Nichols chart to achieve robust controllers for MIMO systems.

Parallel to the evolution of state space control theory, considerable theoretical advances were made in the field of game theory. The solution of a game is a set of strategies for each player such that no player can benefit from a change in his own strategy. Game theory has found most of its successful applications in economics and bargaining problems, but a subset of the theories, known as non-cooperative games, also apply to the field of robust control. The celebrated $\mathcal{H}_\infty$ theory is an example of such robust control synthesis where the solution of an algebraic Riccati equation (ARE) gives the equilibrium strategies for non-cooperating players.

Many tools for state space stability and robustness analysis have their origins in Lyapunov stability theory and the small gain theorem, both applicable to linear as well as nonlinear systems. For linear systems the criteria for stability can be expressed in terms of eigenvalues, and the degree of robustness is expressed through singular values. One analysis method using the structured singular value of a system to quantify the degree of robustness is known as μ analysis.

Recent research has resulted in a focus on solution of general linear matrix inequalities (LMI). These arise from problem formulations using multiple objectives ("pareto-optimal control") or other measures of the objectives than those in the traditional optimal or robust setting. This shift in focus is mainly due to the advent of efficient methods for numerical solution of the convex optimization problems, making general LMI problems tractable. With these methods one can guarantee not only robustness properties of the stability of a system but also statements regarding performance.

1.2.5 Nonlinear methods

Development and research on methods for control synthesis and analysis of nonlinear systems is far from as mature as the tools developed in the linear domain. A few techniques are however used in the industrial setting due to their great advantages in enabling increased system performance.

Many of the aforementioned theories for stability analysis are not limited to linear systems, or have a nonlinear counterpart or extension. The need for nonlinear methods is natural since most physical systems are in fact nonlinear even though many problems can be linearized and thus formulated and solved in the linear setting with satisfactory results. Some isolated nonlinearities can be handled with describing functions and analysis using the classical methods. One early tool for stability analysis for nonlinear differential equations is the circle criterion by Popov.

Methods for controlling nonlinear systems are often formulated in the terms of an optimization problem, or dynamical program. Bellman formulated the principle of dynamical programming using the Hamilton-Jacobi equations, and this is the foundation of many optimization methods. The application of analytical approaches are often limited to small systems, and for larger systems one has to resort to numerical methods.

Differential geometric methods are playing an important role in much of the present research on nonlinear control systems. In particular the use of Lie groups is an active area of research. By Lie group methods, system symmetries may be exploited.

Neural networks are popular in applications where the system dynamics are not explicitly modelled due to the adaptive nature of the approach. Other methods inspired by nature are the genetic algorithms who seek solutions to optimization problems in a process of "natural" selection. Most theories of adaptive control strategies suffer from lack of robustness properties, and remedies to handle this problem is an active topic in the research community today.

Methods for nonlinear control is the topic of this report where a number of constructive control synthesis methods for nonlinear systems are discussed in chapters 3 and 4 and an example of application in 5.

1.2.6 The future

Many tools necessary for controlling complex systems are becoming mature and thus applicable to real life problems. The main advantages of modern methods for controlling nonlinear systems is that one can allow the system to operate in a range closer to the physical limitations of the equipment. This could mean sharper turns with a missile or allowing a wind turbine to operate in strong gusty winds. This increase in performance can be of substantial tactical and economical value. In order to exploit the possibilities of modern control theory there is a need for integrated development teams when a new system is designed. This allows for accurate modeling and rapid feedback on design limitations from all perspectives. The system performance can often be increased by minute modification of the design motivated by controllability and robustness properties. Such a rapid prototyping approach can aid in avoiding pitfalls and unnecessary costs for achieving desired performance objectives. The advent on an integrated environment with tools to support system design and modeling as well as control synthesis and robustness analysis would allow optimization of performance objectives based on economical metrics for the complex systems of the future.

1.3 This report

Later chapters in this report will cover the state of the art in model order reduction in chapter 2, modern techniques for stabilization and tracking including exact linearisation or Nonlinear dynamic inversion (NDI) and in particular backstepping in chapter 3. Optimization based methods such as robust model predictive control and the connection to differential games is discussed in chapter 4. An example of applying block/vector backstepping to a model of a small fighter aircraft is presented in chapter 5.

2 Model Order Reduction

2.1 Introduction

Consider the linear MIMO system

$$\boldsymbol{\Sigma} : \begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}u(t) \\ y(t) &= \boldsymbol{C}\boldsymbol{x}(t) \end{cases} \tag{2.1}$$

with initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, where $\boldsymbol{x}(t)$ is the state-space vector of dimension m, $u(t)$ is the input and $y(t)$ is the output.

The objective in reduced-order modelling is to construct a reduced order model

$$\tilde{\boldsymbol{\Sigma}} : \begin{cases} \dot{\tilde{\boldsymbol{x}}}(t) &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{B}}u(t) \\ y(t) &= \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}}(t) \end{cases} \tag{2.2}$$

such that the state-space dimension of the reduced order model $\tilde{\boldsymbol{\Sigma}}$ is much smaller than the state-space dimension of the original model $\boldsymbol{\Sigma}$, and in the same time the input/output behaviour for the reduced order model $\tilde{\boldsymbol{\Sigma}}$ approximates the original model sufficiently well given that the input signal belongs to a given class.

The main reason for constructing reduced order models is to obtain a model that approximates the original model sufficiently well, and in the same time is much more efficient than the original model with respect to computation and memory requirements.

The two main applications of reduced order models are

- Replace the original model in large scale simulations in order to reduce the simulation time.
- Used for constructing low dimensional controllers for real time applications.

Applications where reduced order models are used include circuit and mems simulation, structural dynamics analysis and simulation of fluids.

We will review two different methods for linear systems, balanced truncation and Krylov subspace methods. The advantage with balanced truncation is that there exists global error bounds, the disadvantage is that the method is very computationally demanding. The advantage with Krylov subspace methods is that they are computationally efficient, the disadvantage is that there exists no global error bounds. Both methods can be described by means of change of variables and projection. Later on we will briefly describe their extensions to nonlinear systems.

2.2 Projection framework for linear systems

Let $\boldsymbol{V}$ be a basis for a subspace $\mathcal{S}$, and $\boldsymbol{W}$ be a basis for a subspace $\mathcal{P}$, both of dimension $\hat{m}$, $\hat{m} << m$. Further we assume that the basis are biorthogonal, i.e. $\boldsymbol{W}^T\boldsymbol{V} = \boldsymbol{I}$.

The oblique projection onto $\mathcal{S}$ and orthogonal to $\mathcal{P}$ can be represented by

$$\boldsymbol{P} = \boldsymbol{V}\boldsymbol{W}^T \tag{2.3}$$

Make a change of variable $\boldsymbol{x} = \boldsymbol{V}\hat{\boldsymbol{x}}$, and let the projection of the residual equals to zero

$$\begin{aligned} \boldsymbol{P}(\boldsymbol{V}\dot{\hat{\boldsymbol{x}}} - \boldsymbol{A}\boldsymbol{V}\hat{\boldsymbol{x}} + \boldsymbol{B}u(t)) &= \boldsymbol{V}\boldsymbol{W}^T(\boldsymbol{V}\dot{\hat{\boldsymbol{x}}} - \boldsymbol{A}\boldsymbol{V}\hat{\boldsymbol{x}} - \boldsymbol{B}u(t)) \\ &= \boldsymbol{V}(\boldsymbol{W}^T\boldsymbol{V}\dot{\hat{\boldsymbol{x}}} - \boldsymbol{W}^T\boldsymbol{A}\boldsymbol{V}\hat{\boldsymbol{x}} - \boldsymbol{W}^T\boldsymbol{B}u(t)) \end{aligned} \tag{2.4}$$

From the a above we can conclude that

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{W}^T\boldsymbol{A}\boldsymbol{V}\hat{\boldsymbol{x}} + \boldsymbol{W}^T\boldsymbol{B}u(t) \tag{2.5}$$

which is the state space equation of the reduced order model.

Finally, the reduced order model can be defined as

$$\tilde{\boldsymbol{\Sigma}}_{\mathrm{L}} : \left\{ \begin{array}{rcl} \dot{\tilde{\boldsymbol{x}}}(t) &=& \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{B}}u(t) \\ y(t) &=& \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}}(t) \end{array} \right. \tag{2.6}$$

where

$$\tilde{\boldsymbol{A}} = \boldsymbol{W}^T\boldsymbol{A}\boldsymbol{V}, \;\; \tilde{\boldsymbol{B}} = \boldsymbol{W}^T\boldsymbol{B}, \;\; \tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{V} \tag{2.7}$$

Several method for reduced order modelling of linear systems can be seen as change of variable and projection. They differ in the strategies how to choose the subspaces, the construction of the basis, and the construction of the projection.

2.3 Balanced Model Reduction

Consider a stable system of minimal realisation

$$\boldsymbol{\Sigma} \left\{ \begin{array}{rcl} \dot{\boldsymbol{x}}(t) &=& \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}u(t) \\ y(t) &=& \boldsymbol{C}\boldsymbol{x}(t) \end{array} \right. \tag{2.8}$$

with the corresponding controllability and observability Gramians $\boldsymbol{P} > 0$, $\boldsymbol{Q} > 0$. The controllability and observability Gramians are the solution to the Lyapunov equations

$$\boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{0} \tag{2.9}$$

$$\boldsymbol{A}^T\boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{A} + \boldsymbol{C}^T\boldsymbol{C} = \boldsymbol{0} \tag{2.10}$$

Since we assume that the system is stable, all eigenvalues of $\boldsymbol{A}$ have negative real part. Since we also assume that the system is a minimal realisation, the gramians $\boldsymbol{P}$ and $\boldsymbol{Q}$ are positive definite.

The first step in balanced model order reduction is to construct a balanced realisation, the balanced realisation is then truncated in order to construct a reduced order model. A balanced realisation is a realisation where the controllability and observability gramians are equal and diagonal.

$$\boldsymbol{P} = \boldsymbol{Q} = \boldsymbol{\Sigma}, \;\; \boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n), \;\; \sigma_1 \geq \sigma_2 \geq \sigma_3 \cdots \geq \sigma_n \tag{2.11}$$

The values $\sigma_1, \ldots, \sigma_n$ are called Hankel singular values. A balanced realisation can be found by a appropriate coordinate transformation

$$\hat{\boldsymbol{x}} = \boldsymbol{T}\boldsymbol{x} \tag{2.12}$$

By this coordinate transformation, the system turns into

$$\hat{\boldsymbol{\Sigma}} \left\{ \begin{array}{rcl} \dot{\hat{\boldsymbol{x}}}(t) &=& \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{B}}u \\ y(t) &=& \hat{\boldsymbol{C}}\hat{\boldsymbol{x}}(t) \end{array} \right. \tag{2.13}$$

where

$$\hat{\boldsymbol{A}} = \boldsymbol{TAT}^{-1}, \;\; \hat{\boldsymbol{B}} = \boldsymbol{TB}, \;\; \hat{\boldsymbol{C}} = \boldsymbol{CT}^{-1} \tag{2.14}$$

and the gramians in the new coordinates

$$\hat{\boldsymbol{P}} = \boldsymbol{TPT}^T, \;\; \hat{\boldsymbol{Q}} = (\boldsymbol{T}^{-1})^T \boldsymbol{QT}^{-1} \tag{2.15}$$

The goal is to choose $\boldsymbol{T}$ such that the gramians in the new coordinate are equal and diagonal

$$\begin{aligned} &\hat{\boldsymbol{P}} = \boldsymbol{TPT}^T = \boldsymbol{\Sigma}, \;\; \hat{\boldsymbol{Q}} = (\boldsymbol{T}^{-1})^T \boldsymbol{QT}^{-1} = \boldsymbol{\Sigma} \\ &\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n), \;\; \sigma_1 \geq \sigma_2 \geq \sigma_3 \cdots \geq \sigma_n \end{aligned} \tag{2.16}$$

Multiply $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{Q}}$

$$\hat{\boldsymbol{P}}\hat{\boldsymbol{Q}} = \boldsymbol{TPQT}^{-1} \tag{2.17}$$

The eigenvalues of the product $\hat{\boldsymbol{P}}\hat{\boldsymbol{Q}}$ are invariant under the transformation $\hat{\boldsymbol{x}} = \boldsymbol{Tx}$, it is a similarity transformation. We want to choose $\boldsymbol{T}$ such that

$$\boldsymbol{TPQT}^{-1} = \boldsymbol{\Sigma}^2 \tag{2.18}$$

Factorise $\boldsymbol{P}$ by a cholesky factorisation

$$\boldsymbol{P} = \boldsymbol{R}^T\boldsymbol{R}, \;\; \boldsymbol{R} - \text{upper triangular} \tag{2.19}$$

(2.18) can be rewritten as

$$(\boldsymbol{RQR}^T)\boldsymbol{R}^{-T}\boldsymbol{T}^{-1} = \boldsymbol{R}^{-T}\boldsymbol{T}^{-1}\boldsymbol{\Sigma}^2 \tag{2.20}$$

Since $\boldsymbol{P}$ and $\boldsymbol{Q}$ are positive definite, $\boldsymbol{RQR}^T$ will also be positive definite.

Compute the eigenvalues and eigenvectors to $\boldsymbol{RQR}^T$

$$(\boldsymbol{RQR}^T)\boldsymbol{U} = \boldsymbol{U\Lambda}, \;\; \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \;\; \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} \tag{2.21}$$

The eigenvectors are normalized by (2.21). In order for (2.16) and (2.20) to be fulfilled, choose

$$\boldsymbol{T}^{-1} = \boldsymbol{R}^T\boldsymbol{U}\boldsymbol{\Sigma}^{-\frac{1}{2}} \tag{2.22}$$

and as a consequence

$$\boldsymbol{T} = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{U}^T\boldsymbol{R}^{-T} \tag{2.23}$$

Note that $\boldsymbol{U}$ is orthonormal, and thus $\boldsymbol{U}^{-1} = \boldsymbol{U}^T$.

We are now ready to give an algorithm for computing a balanced realisation, given an stable system of minimal realisation.

Balanced Realisation
1 Compute the controllability and observability Gramians $\boldsymbol{P}$ and $\boldsymbol{Q}$
2 $\boldsymbol{P} = \boldsymbol{R}^T\boldsymbol{R}$ Factorize $\boldsymbol{P}$
3 $\boldsymbol{M} = \boldsymbol{RQR}^T$ Compute the product
4 $\boldsymbol{MU} = \boldsymbol{U\Lambda}$ Compute the eigenvectors $\boldsymbol{U}$ and eigenvalues $\boldsymbol{\Lambda}$ of $\boldsymbol{M}$
5 $\boldsymbol{\Sigma} := \boldsymbol{\Lambda}^{\frac{1}{2}}$ Set
6 $\boldsymbol{T}^{-1} := \boldsymbol{R}^T\boldsymbol{U}\boldsymbol{\Sigma}^{-\frac{1}{2}}$ Set
7 $\hat{\boldsymbol{A}} := \boldsymbol{TAT}^{-1}, \;\; \hat{\boldsymbol{B}} := \boldsymbol{TB}, \;\; \hat{\boldsymbol{C}} := \boldsymbol{CT}^{-1}, \;\; \hat{\boldsymbol{D}} := \boldsymbol{D}$ Set

The balanced realisation in state space is defined as

$$\hat{\boldsymbol{\Sigma}}_{\mathrm{L}}: \left\{ \begin{array}{rcl} \dot{\hat{\boldsymbol{x}}}(t) & = & \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{B}}u(t) \\ y(t) & = & \hat{\boldsymbol{C}}\hat{\boldsymbol{x}}(t) \end{array} \right. \tag{2.24}$$

A reduced order model of state-space dimension k based on balanced realisation (2.24) is constructed by taking the part of the system corresponding to the k largest Hankel singular values,

Finally, the reduced order model can be defined as

$$\tilde{\boldsymbol{\Sigma}}_{\mathrm{L}}: \left\{ \begin{array}{rcl} \dot{\tilde{\boldsymbol{x}}}(t) & = & \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{B}}u(t) \\ y(t) & = & \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}}(t) \end{array} \right. \tag{2.25}$$

where

$$\tilde{\boldsymbol{A}} = \hat{\boldsymbol{A}}_{1:k,1:k};\ \tilde{\boldsymbol{B}} = \hat{\boldsymbol{B}}_{1:k,:}\ \ \tilde{\boldsymbol{C}} = \hat{\boldsymbol{C}}_{:,1:k},\ \ \tilde{\boldsymbol{D}} = \hat{\boldsymbol{D}} \tag{2.26}$$

Let $\boldsymbol{G}(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}$ be the transfer matrix of the original linear system (2.24), and let $\tilde{\boldsymbol{G}}(s) = \tilde{\boldsymbol{C}}(s\tilde{\boldsymbol{I}} - \tilde{\boldsymbol{A}})^{-1}\tilde{\boldsymbol{B}}$ be the transfer matrix of the reduced order linear system (2.25). A global error bound is given by

$$\| \boldsymbol{G}(s) - \tilde{\boldsymbol{G}}(s) \|_{\infty} \leq 2(\sigma_{k+1} + \sigma_{k+2} + \ldots \sigma_n) \tag{2.27}$$

See [43] for a exact statement of condition and proof.

In terms of the projection framework, the projection matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ are identified as

$$\begin{aligned} \boldsymbol{V} &= \boldsymbol{T}^{-1}(:, 1:k) \\ \boldsymbol{W} &= \boldsymbol{T}^{T}(:, 1:k) \end{aligned} \tag{2.28}$$

Note that $\boldsymbol{W}^T\boldsymbol{V} = \boldsymbol{I}$.

2.4 Moment matching through Krylov subspaces

2.4.1 Moment matching

In this section, we will review how to construct reduced order models based on implicit moment matching. The moments are implicitly matched by constructing Krylov subspaces as the projection subspaces.

Consider the SISO linear system

$$\Sigma_{\mathrm{L}}: \left\{ \begin{array}{rcl} \dot{\boldsymbol{x}}(t) & = & \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t) \\ y(t) & = & \boldsymbol{c}\boldsymbol{x}(t) \end{array} \right. \tag{2.29}$$

make an Laplace transform

$$\Sigma_{\mathrm{L}}: \left\{ \begin{array}{rcl} s\boldsymbol{x}(s) & = & \boldsymbol{A}\boldsymbol{x}(s) + \boldsymbol{b}u(s) \\ y(s) & = & \boldsymbol{c}\boldsymbol{x}(s) \end{array} \right. \tag{2.30}$$

and construct the transfer function

$$\boldsymbol{G}(s) = \boldsymbol{c}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} \tag{2.31}$$

Expand the transfer function $\boldsymbol{G}(s)$ around $s = 0$ in a series expansion

$$\boldsymbol{G}(s) = (-1)\sum_{l=1}^{\infty} s^{l-1}\boldsymbol{c}\boldsymbol{A}^{-l}\boldsymbol{b} \tag{2.32}$$

The coefficients $m(l) = -\boldsymbol{c}\boldsymbol{A}^{-l}\boldsymbol{b}, l = 1, \ldots, \infty$ are the (low-frequency) moments. We want to construct a reduced order model in such a way that the first k moments are matched. This is done in an explicit way through the AWE algorithm [29]. It turns out that explicit moment matching is a numerically unstable process, it is much more numerically stable to construct the reduced order model in such a way that the moments are matched implicitly through constructing basis for Krylov subspaces and utilise projection [11]. We will now describe such a procedure, this procedure is build on [26].

First we build up an orthonormal basis $\boldsymbol{V}$ for the right Krylov subspace

$$\mathcal{K}_k(\boldsymbol{A}^{-1}, \boldsymbol{A}^{-1}\boldsymbol{b}) = \text{span}\{\boldsymbol{A}^{-1}\boldsymbol{b}, \boldsymbol{A}^{-2}\boldsymbol{b}, \dots, \boldsymbol{A}^{-k}\boldsymbol{b}\} \tag{2.33}$$

then the reduced order model is constructed through projection.

Let $\boldsymbol{P}$ be the orthogonal projection matrix onto the Krylov subspace $\mathcal{K}_k(\boldsymbol{A}^{-1}, \boldsymbol{A}^{-1}\boldsymbol{b})$, and let $\boldsymbol{V}$ be an orthonormal basis for the same subspace. Then $\boldsymbol{P}$ can be represented by

$$\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^T \tag{2.34}$$

Make a change of variable $\boldsymbol{x} = \boldsymbol{V}\tilde{\boldsymbol{x}}$, and let the projection of the residual be equal to zero

$$\begin{aligned} 0 &= \boldsymbol{P}(\boldsymbol{V}\dot{\tilde{\boldsymbol{x}}} - \boldsymbol{A}\boldsymbol{V}\tilde{\boldsymbol{x}} - \boldsymbol{b}u) \\ &= \boldsymbol{V}\boldsymbol{V}^T(\boldsymbol{V}\dot{\tilde{\boldsymbol{x}}} - \boldsymbol{A}\boldsymbol{V}\tilde{\boldsymbol{x}} - \boldsymbol{b}u) \\ &= \boldsymbol{V}(\dot{\tilde{\boldsymbol{x}}} - \boldsymbol{V}^T\boldsymbol{A}\boldsymbol{V}\tilde{\boldsymbol{x}} - \boldsymbol{V}^T\boldsymbol{b}u) \end{aligned} \tag{2.35}$$

From the above one can identify a reduced order model

$$\Sigma_{\mathrm{L}} : \begin{cases} \dot{\tilde{\boldsymbol{x}}}(t) &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{b}}u(t) \\ y(t) &= \tilde{\boldsymbol{c}}\tilde{\boldsymbol{x}}(t) \end{cases} \tag{2.36}$$

where

$$\tilde{\boldsymbol{A}} = \boldsymbol{V}^T\boldsymbol{A}\boldsymbol{V}, \;\; \tilde{\boldsymbol{b}} = \boldsymbol{V}^T\boldsymbol{b}, \;\; \tilde{\boldsymbol{c}} = \boldsymbol{c}\boldsymbol{V} \tag{2.37}$$

The first k moments of the reduced order model matches the first k moments of the original model.

$$\begin{aligned} \tilde{m}(l) &= \tilde{\boldsymbol{c}}\tilde{\boldsymbol{A}}^{-l}\tilde{\boldsymbol{b}}, \;\; l = 1, \dots, k \\ &= \boldsymbol{c}\boldsymbol{A}^{-l}\boldsymbol{b} \\ &= m(l) \end{aligned} \tag{2.38}$$

for a proof see [26].

2.4.2 Construction of a basis for a Krylov subspace

In this section we will give an algorithm for computing an orthonormal basis $\boldsymbol{V}_k$ for the Krylov subspace (2.33) used as the projection subspace, the Arnoldi algorithm, here formulated with operations with the inverse of $\boldsymbol{A}$.

Arnoldi algorithm

$\boldsymbol{v}_1 = \boldsymbol{A}^{-1}\boldsymbol{b}$ start vector
  for j=1:k
    $\boldsymbol{r} = \boldsymbol{A}^{-1}\boldsymbol{v}_j$ operate with $\boldsymbol{A}^{-1}$
    $\boldsymbol{f}_j = \boldsymbol{V}_j^T\boldsymbol{r}$
    $\boldsymbol{r} = \boldsymbol{r} - \boldsymbol{V}_j\boldsymbol{f}_j$
    $f_{j+1,j} = \parallel \boldsymbol{r} \parallel$ if $f_{j+1,j} = 0$ stop
    $\boldsymbol{v}_{j+1} = \boldsymbol{r}/f_{j+1,j}$
  end

- The operation with the inverse on line 3, for efficiency should be implemented by using (sparse) matrix factorisation and solvers.

- Lines $4-7$ are the orthogonalisation steps, the resulting basis $\boldsymbol{V}_{k+1}$ is orthonormal by construction.

- To ensure that the basis is orthonormal to working accuracy, the orthogonalisation steps should be implemented with reorthogonalisation, for example, see the procedure `gsreorthog` on page 287 in [39].

- If $f_{j+1,j}=0$, then stop, the resulting basis $\boldsymbol{V}_j$ span an invariant subspace under $\boldsymbol{A}^{-1}$.

2.4.3 System properties

Apart from the moment matching capabilities, in many cases it is important that system property such as stability is preserved. For example, if $\boldsymbol{A}$ is negative definite, the system is stable. The congruence transformation

$$\tilde{\boldsymbol{A}}=\boldsymbol{V}^T\boldsymbol{A}\boldsymbol{V} \tag{2.39}$$

preserves the matrix property of negative definiteness, and thus the system property stability is preserved.

2.4.4 Extensions and further readings

For an overview of Krylov subspace methods for linear systems, see [12, 18] and the references therein. We will here discuss some important issues.

If in addition to basis for the right Krylov subspace (2.33), a basis $\boldsymbol{W}$ for the left Krylov subspace is build up

$$\operatorname{span}\{\boldsymbol{W}\}=\mathcal{K}_k(\boldsymbol{A}^{-T},\boldsymbol{A}^{-T}\boldsymbol{c}^T) \tag{2.40}$$

it is possible to match $2k$ moments. If the basis are constructed to be biorthogonal

$$\boldsymbol{W}^T\boldsymbol{V}=\boldsymbol{I} \tag{2.41}$$

then a reduced order model(r.o.m.) is defined by (2.6).

The advantage with the two-sided version is that it matches $2k$ moments, which is the double number of moments than for the one-sided version, the Arnoldi algorithm. The disadvantage is that it is less stable. The biorthogonalisation process can break down in a way that the orthogonalisation process cannot do. It fails to construct a biorthogonal basis even though the left and right Krylov subspaces are not exhausted. This can be remedied with "look ahead", see [14, 13, 41].

Both one-sided and two-sided methods can be extended to multiple input and multiple output. The Krylov subspace corresponding to one starting vector can be exhausted before the other, this lead to a breakdown that can be cured by a deflation procedure, see [1].

The moment matching methods are local in nature. In frequency domain, the reduced order model is well approximated close to the expansion point, but less well approximated further away. A reduced order model that is well approximated throughout a larger frequency range can be constructed by using several expansion points $s_1,\dots,s_l$, and constructing a reduced order model such that the moments are matched around each expansion point up to a given order, see [18, 37, 16]

2.5 Nonlinear model order reduction

2.5.1 Proper orthogonal decomposition

Consider the nonlinear system

$$\boldsymbol{\Sigma} : \begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{b}u(t) \\ y(t) &= \boldsymbol{c}\boldsymbol{x}(t) \end{cases} \tag{2.42}$$

The basic idea in this subsection is to construct a basis for the projection subspace based on simulation.

Let $u(t)$ be a representative input signal for the simulation one wish to perform. Sample the state vector at n different times

$$\boldsymbol{x}(t_1), \ldots, \boldsymbol{x}(t_n) \tag{2.43}$$

Put the sampled state vectors together into a matrix $\boldsymbol{X}$

$$\boldsymbol{X} = [\boldsymbol{x}(t_1), \boldsymbol{x}(t_2), \ldots, \boldsymbol{x}(t_n)] \tag{2.44}$$

and compute the singular value decomposition (SVD)

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{2.45}$$

where

$$\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m] \in \mathcal{R}^{m,m}, \quad \boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathcal{R}^{m,n}, \quad \boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n] \in \mathcal{R}^{n,n} \tag{2.46}$$

The matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal and the matrix $\boldsymbol{\Sigma}$ is a diagonal matrix with nonnegative values on the diagonal and zeros elsewhere. The values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p$, $p = \min\{m, n\}$ are called the singular values. The vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the ith left and right singular vectors respectively.

As a basis for a projection subspace we take the k left singular vectors $\boldsymbol{U}_k$ that corresponds to the k largest singular values. The reduced order model is defined as follows

$$\hat{\boldsymbol{\Sigma}} : \begin{cases} \dot{\hat{\boldsymbol{x}}}(t) &= \hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}(t)) + \hat{\boldsymbol{b}}u(t) \\ y(t) &= \hat{\boldsymbol{c}}\hat{\boldsymbol{x}}(t) \end{cases} \tag{2.47}$$

where

$$\hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}) = \boldsymbol{U}_k^T \boldsymbol{f}(\boldsymbol{U}_k \hat{\boldsymbol{x}}), \quad \hat{\boldsymbol{b}} = \boldsymbol{U}_k^T \boldsymbol{b}, \quad \hat{\boldsymbol{c}} = \boldsymbol{c}\boldsymbol{U}_k \tag{2.48}$$

One major drawback of P.O.D. is in the representation of the nonlinear function $\boldsymbol{f}$ in the r.o.m. In order to evaluate the function $\hat{\boldsymbol{f}}(\hat{\boldsymbol{x}})$ one needs to construct a vector

$$\boldsymbol{U}_k \hat{\boldsymbol{x}} \tag{2.49}$$

of original dimension m, and then evaluate the original function $\boldsymbol{f}$. The other drawback is that P.O.D. relies on simulation (or measurement) in order to construct a projection subspace. Despite the drawback of P.O.D. it plays an important role in model order reduction of nonlinear systems. For further discussion about P.O.D. see [20, 30].

2.5.2 Extension of Balanced truncation to nonlinear systems

Balanced truncation has been extended to nonlinear systems [34].

2.5.3 Extension of Krylov subspace methods to nonlinear systems

Krylov subspace methods for linear systems have been extended to nonlinear systems in several different ways [8, 31, 27, 28, 24, 6, 7, 4, 5]. Here we will discuss approaches based on multimoment matching and projection of bilinear systems [6, 7, 4, 5], they are inspired by [27, 28]. For the one-sided method, the subspaces build up are similar but the projection framework is different. The two-sided method, which is briefly discussed in [4] ,[5] and more throughly discussed in [7], is much more efficient in terms of multimoment matching than the one-sided method. Differently from P.O.D, these methods construct "true" reduced order models, and the basis for the projection subspace is constructed to incorporate nonlinear terms through moment matching.

A bilinear system is linear in state and linear in input, but not jointly linear in state and input. Consider the single input single output bilinear systems.

$$\boldsymbol{\Sigma} : \begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{N}\boldsymbol{x}(t)u(t) + \boldsymbol{B}u(t) \\ y(t) &= \boldsymbol{c}^T\boldsymbol{x}(t) \end{cases} \tag{2.50}$$

Bilinear systems arise as natural models for many different physical and biomedical processes [25]. They can also be used to approximate more general nonlinear systems, for example through the Carleman bilinearisation [33].

The Volterra-Wiener representation of the bilinear system is given by

$$y(t) = \sum_{k=1}^{\infty} y_k(t), \tag{2.51}$$

where $y_k(t)$ is the degree-k subsystem of the form

$$y_k(t) = \int_0^t h_{reg}(t_1, t_2, \ldots, t_k)u(t-t_1-\cdots-t_k)u(t-t_2-\cdots-t_k)\cdots u(t-t_k)dt_1\cdots dt_k \tag{2.52}$$

with the associated k-th degree regular kernel

$$h_{reg}(t_1, t_2, \ldots, t_k) = \boldsymbol{c}^T e^{\boldsymbol{A}t_k}\boldsymbol{N}\cdots\boldsymbol{N}e^{\boldsymbol{A}t_2}\boldsymbol{N}e^{\boldsymbol{A}t_1}\boldsymbol{b}. \tag{2.53}$$

For further details of the Volterra-Wiener representation of bilinear systems, see [33]. The multi-dimensional Laplace transform of the regular kernel $h_{reg}(t_1, \ldots, t_k)$ yields the transfer function

$$H_k(s_1, s_2, \ldots, s_k) = \boldsymbol{c}^T(s_k\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{N}\cdots\boldsymbol{N}(s_2\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{N}(s_1\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b}. \tag{2.54}$$

By the power series expansion of $(s_j\boldsymbol{I} - \boldsymbol{A})^{-1}$, the transfer function can be expanded in a multivariable series expansion

$$H_k(s_1, s_2, \ldots, s_k) = \sum_{l_k=1}^{\infty}\cdots\sum_{l_1=1}^{\infty} m(l_1, l_2, \ldots, l_k)s_1^{l_1-1}s_2^{l_2-1}\cdots s_k^{l_k-1}, \tag{2.55}$$

where

$$m(l_1, l_2, \ldots, l_k) = (-1)^k\boldsymbol{c}^T\boldsymbol{A}^{-l_k}\boldsymbol{N}\ldots\boldsymbol{N}\boldsymbol{A}^{-l_2}\boldsymbol{N}\boldsymbol{A}^{-l_1}\boldsymbol{b}, \tag{2.56}$$

are called the *multimoments* of the degree-k kernel.

Our approach for model order reduction for bilinear systems is to match the multimoments for the first few kernels of the original bilinear system up to a given order. This is done through a projection framework.

A basis $\boldsymbol{V}$ for a right projection subspace is built up in the following way. First a basis $\boldsymbol{V}^{(1)}$ for a Krylov subspace is constructed by starting with $\boldsymbol{A}^{-1}\boldsymbol{b}$ and operating with $\boldsymbol{A}^{-1}$

$$\text{span}\{\boldsymbol{V}^{(1)}\} = \mathcal{K}_{q_1}(\boldsymbol{A}^{-1}, \boldsymbol{A}^{-1}\boldsymbol{b}) \tag{2.57}$$

the each basis $\boldsymbol{V}^{(k)}$ span a block Krylov subspace by starting on $\boldsymbol{A}^{-1}\boldsymbol{N}$ times the first p_k basis vectors of the previous basis $\boldsymbol{V}^{(k-1)}$ and operating with $\boldsymbol{A}^{-1}$

$$\operatorname{span}\{\boldsymbol{V}^{(k)}\} = \mathcal{K}_{q_k}(\boldsymbol{A}^{-1}, \boldsymbol{A}^{-1}\boldsymbol{N}\boldsymbol{V}^{(k-1)}_{[p_k]}), \tag{2.58}$$

An orthonormal basis $\boldsymbol{V}$ of the projection subspace is chosen as a union of the subspaces $\operatorname{span}\{\boldsymbol{V}^{(k)}\}$:

$$\operatorname{span}\{\boldsymbol{V}\} = \bigcup_{k=1}^{r} \operatorname{span}\{\boldsymbol{V}^{(k)}\}. \tag{2.59}$$

an orthogonal projection approximation, it yields a reduced bilinear system of Σ_{B}:

$$\hat{\Sigma}_{\mathrm{B1}} : \begin{cases} \dot{\hat{\boldsymbol{z}}}(t) = \hat{\boldsymbol{A}}\hat{\boldsymbol{z}}(t) + \hat{\boldsymbol{N}}\,\hat{\boldsymbol{z}}(t)\,u(t) + \hat{\boldsymbol{b}}u(t), \\ y(t) = \hat{\boldsymbol{c}}^T\hat{\boldsymbol{z}}(t), \end{cases} \tag{2.60}$$

where

$$\hat{\boldsymbol{A}}^{-1} = \boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{V},\ \ \hat{\boldsymbol{N}} = \hat{\boldsymbol{A}}\boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{N}\boldsymbol{V},\ \ \hat{\boldsymbol{b}} = \boldsymbol{A}\boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{b},\ \ \hat{\boldsymbol{c}} = \boldsymbol{V}^T\boldsymbol{c}. \tag{2.61}$$

It can be shown that the reduced bilinear system $\hat{\Sigma}_{\mathrm{B1}}$ matches the same number of multimoments as the number of basis vectors. For a precise statement and proof of multimoment-matching, see [6].

A basis $\boldsymbol{W}$ for the left subspace is built up in the following way. First a basis $\boldsymbol{W}^{(1)}$ for a Krylov subspace is constructed by starting with $\boldsymbol{c}$ and operating with $\boldsymbol{A}^{-T}$

$$\operatorname{span}\{\boldsymbol{W}^{(1)}\} = \mathcal{K}_{q_1}(\boldsymbol{A}^{-T}, \boldsymbol{c}). \tag{2.62}$$

then each basis $\boldsymbol{W}^{(k)}$ span a block Krylov subspace by starting on $\boldsymbol{N}^T\boldsymbol{A}^{-T}$ times the first p_k basis vectors of the previous basis $\boldsymbol{W}^{(k-1)}$ and operating with $\boldsymbol{A}^{-T}$

$$\operatorname{span}\{\boldsymbol{W}^{(j)}\} = \mathcal{K}_{q_k}(\boldsymbol{A}^{-T}, \boldsymbol{N}^T\boldsymbol{A}^{-T}\boldsymbol{W}^{(k-1)}_{[p_k]}), \tag{2.63}$$

The basis $\boldsymbol{W}$ for the left projection subspace is then taken as a union of these Krylov subspaces $\operatorname{span}\{\boldsymbol{W}^{(k)}\}$

$$\operatorname{span}\{\boldsymbol{W}\} = \bigcup_{k=1}^{r} \operatorname{span}\{\boldsymbol{W}^{(k)}\} \tag{2.64}$$

Furthermore, the bases $\boldsymbol{V}$ and $\boldsymbol{W}$ are constructed to be biorthogonal. By the principle of an oblique projection approximation, it yields a reduced bilinear system of Σ_{B}:

$$\hat{\Sigma}_{\mathrm{B2}} : \begin{cases} \dot{\hat{\boldsymbol{z}}}(t) = \hat{\boldsymbol{A}}\hat{\boldsymbol{z}}(t) + \hat{\boldsymbol{N}}\,\hat{\boldsymbol{z}}(t)\,u(t) + \hat{\boldsymbol{b}}u(t), \\ \hat{y}(t) = \hat{\boldsymbol{c}}^T\hat{\boldsymbol{z}}(t), \end{cases} \tag{2.65}$$

where

$$\hat{\boldsymbol{A}}^{-1} = \boldsymbol{W}^T\boldsymbol{A}^{-1}\boldsymbol{V},\ \ \hat{\boldsymbol{N}} = \hat{\boldsymbol{A}}\boldsymbol{W}^T\boldsymbol{A}^{-1}\boldsymbol{N}\boldsymbol{V},\ \ \hat{\boldsymbol{b}} = \hat{\boldsymbol{A}}\boldsymbol{W}^T\boldsymbol{A}^{-1}\boldsymbol{b},\ \ \hat{\boldsymbol{c}} = \boldsymbol{V}^T\boldsymbol{c}. \tag{2.66}$$

It can be shown that the reduced bilinear system $\hat{\Sigma}_{\mathrm{B2}}$ matches all multimoments that can be represented through the inner product

$$\boldsymbol{s}^T\boldsymbol{r} = (-1)^k\boldsymbol{c}^T\boldsymbol{A}^{-l_k}\boldsymbol{N}\ldots\boldsymbol{N}\boldsymbol{A}^{-l_2}\boldsymbol{N}\boldsymbol{A}^{-l_1}\boldsymbol{b}, \tag{2.67}$$

where

$$\boldsymbol{r} \in \operatorname{span}\{\boldsymbol{V}\},\ \ \boldsymbol{s} \in \operatorname{span}\{\boldsymbol{W}\} \tag{2.68}$$

For a precise statement and proof of multimoment-matching, see [7].

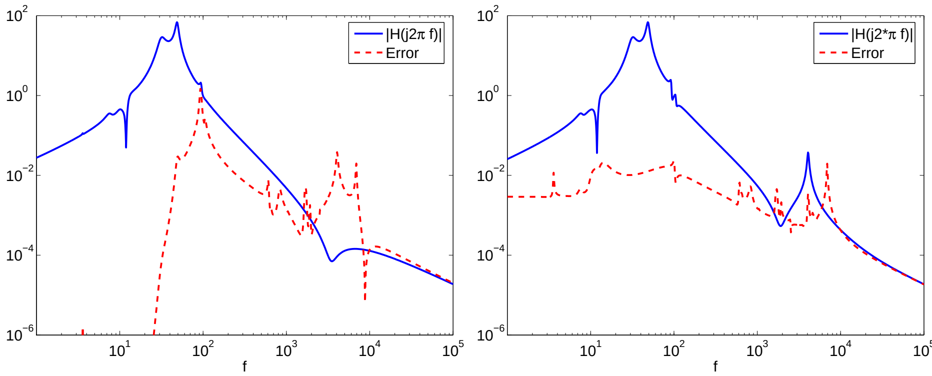


Figure 2.1: The frequency response of the reduced order models constructed by the Arnoldi method (left) and the method of balanced truncation (right) are plotted together with the error. The dimension of the reduced order model is 18.

2.6 Numerical tests

Example 1

Here, the Arnoldi method and the method of balanced truncation for a linear systems are compared. The linear systems used as a test example originate from the the description of the dynamics between the lens actuator and the radial arm position of a portable disc player [18]. The state space dimension is 120. Reduced order models are constructed by the Arnoldi method and method of balanced truncation. Both reduced order models are of dimension 18. The frequency response of the reduced order models together with the error are plotted in figure (2.1). The reduced order model constructed by the Arnoldi method approximate the original model well close to the expansion point $s = 0$, and less well further away. The reduced order model of balanced truncation approximate the original model rather well throughout the frequency range. These results are predicted by the theory.

Example 2

Consider a SISO quadratic system

$$\boldsymbol{\Sigma} : \left\{ \begin{array}{ccl} \dot{\boldsymbol{x}} & = & \boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{A}_2(\boldsymbol{x}\otimes\boldsymbol{x}) + \boldsymbol{b}u(t) \\ y & = & \boldsymbol{c}^T\boldsymbol{x} \end{array} \right. \tag{2.69}$$

where $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are constant matrices of size $\mathcal{R}^{n\times n}$ and $\mathcal{R}^{n\times n^2}$, and $\boldsymbol{b}, \boldsymbol{c}$ are constant vectors. The quadratic system originate from a nonlinear circuit. Two different bilinear systems that approximate the quadratic system are constructed, through Carleman bilinearisation, and an improved bilinear approximation. We will refer to these systems as Quadratic, Bilinear and Improved Bilinear respectively. The improved bilinear approximation is introduced in [5]. For the structure of the system matrices $\boldsymbol{A}$ and $\boldsymbol{N}$ of the bilinear systems, see [5]. The dimension of the quadratic system is 100, and the dimension of the bilinear systems is $n + n^2 = 10100$. In figure 2.2 (left) we plot the response of the Quadratic, Bilinear and Improved Bilinear systems for the input signal $u(t) = \frac{1}{2}(1 - \frac{1}{5}\cos(2t) - \frac{2}{5}\cos(3t))$. The Improved Bilinear system approximates the quadratic system much better than the Bilinear system.

In order for the reduced order model constructed by the two-sided projection method (2.65) to approximate the Improved Bilinear system sufficiently well, the dimension need to be $n = 11$. On the other hand, the dimension of the

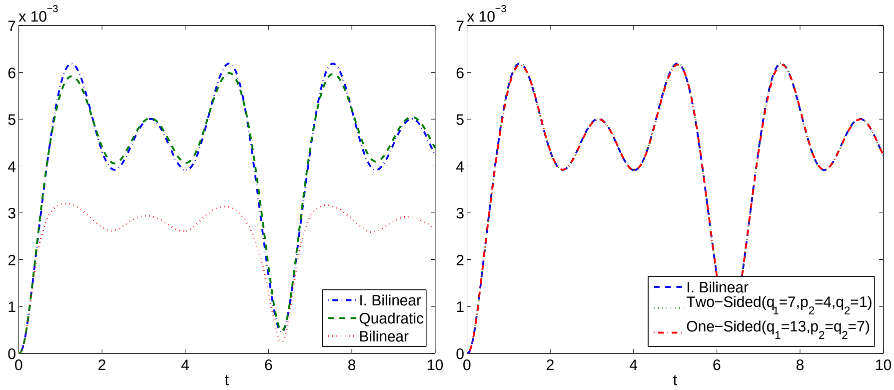


Figure 2.2: The response of the systems Quadratic, Bilinear and Improved Bilinear systems for the input signal $u(t) = \frac{1}{2}(1 - \frac{1}{5}\cos(2t) - \frac{2}{5}\cos(3t))$ are plotted (left). The response of the reduced order models for the input signal $u(t)$ are plotted (right)

reduced order model constructed by the one-sided method (2.60) need to be $n = 62$. The response of the reduced order models for the input signal $u(t)$ is plotted in figure 2.2 (right). For further discussion of this test example, see [5].

3 Backstepping Techniques for Stabilization and Tracking

This chapter gives the ideas behind backstepping up to the point of recent results on tracking properties. As general references for backstepping, see [23] and [35]. For background on the passive normal form, see [40] and [19]. The tracking results are new.

3.1 Introduction and toy example

Backstepping is a general constructive method for control design for nonlinear systems. It was developed from two predecessors, feedback linearization and the method of sliding modes, and shares many features with these. In a sense, backstepping is a generalization covering its parent methods as special cases.

The general idea is as in the following example. Consider the dynamical system

$$\begin{cases} \dot{x} = x^2 + y \\ \dot{y} = x + u \end{cases}$$

where x and y are state variables and u is the control variable. The origin is an unstable equilibrium point, and we want to find a controller $u\,(x, y)$ such that the origin becomes asymptotically stable.

The first equation does not contain u, and may temporarily be considered as a control system in its own right, with x as the state variable and y as the (”virtual”) control. This is a problem of the same kind as the original one, but simpler since it is one-dimensional. We easily obtain a stabilizing controller to this system

$$y = y_{des}(x) = -x^2 - x$$

This is the *desired* value of the virtual control variable y. Introducing a new variable

$$z = y - y_{des}(x) = y + x^2 + x$$

the full dynamics reads

$$\begin{cases} \dot{x} = -x + z \\ \dot{z} = x + 2x(z - x) + u \end{cases}$$

In the second equation we may use the actual control variable u to cancel the terms $x + 2x(z - x)$ and replace them with whatever we want, say $-z$. So by choosing

$$u = -x - 2x(z - x) - z$$

the closed loop dynamics becomes

$$\begin{aligned} \dot{x} &= -x + z \\ \dot{z} &= -z \end{aligned}$$

which is obviously asymptotically stable. In fact, $V = \frac{1}{2}x^2 + \frac{1}{2}z^2$ is a Lyapunov function, since

$$\dot{V} = -x^2 - z^2 + xz \leq -V$$

We have in fact stabilized the system by means of a feedback linearization. The explicit use of a state variable y as a temporary control variable is common to backstepping and the sliding modes method. The "recursive" form (as a sum $\frac{1}{2}x^2 + \frac{1}{2}z^2$) of the Lyapunov function is the hallmark of the backstepping method.

3.2 Sliding Modes and Nonlinear Dynamic Inversion

3.2.1 Sliding Modes

Consider a dynamical system of the form

$$\begin{cases} \dot{x} = f(x, y) \\ \dot{y} = g(x, y, u) \end{cases}$$

where x and y are state variables and u is the control variable. Suppose that $f(0, 0) = 0$ and that we want to construct a controller $u(x, y)$ such that the origin becomes an asymptotically stable equilibrium point of the closed system.

The first equation does not contain u, and may temporarily be considered as a control system in its own right, with x as the state variable and y as the control. This is problem of the same kind as the original one, but in general simpler due to lower state dimension. Assume that we do obtain a stabilizing controller to this system

$$y = y_{des}(x)$$

Introducing a new variable

$$z = y - y_{des}(x)$$

the original system may be written thus

$$\begin{cases} \dot{x} = f_1(x, z) \\ \dot{z} = g_1(x, z, u) \end{cases}$$

where

$$\begin{aligned} f_1(x, z) &= f(x, y_{des}(x) + z) \\ g_1(x, z, u) &= g(x, y_{des}(x) + z, u) - y'_{des}(x) f(x, y_{des}(x) + z) \end{aligned}$$

It now seems plausible that, under suitable conditions, if the control $u(x, y)$ is chosen so that $z \to 0$, by means of the second equation, the first equation will approach $\dot{x} = f_1(x, 0)$, which is stable by construction, and (one may hope), $x \to 0$.

Roughly speaking, the method of sliding modes achieves this by making the $z \to 0$ dynamics fast enough compared to the slower ("sliding") $\dot{x} = f_1(x, 0)$ dynamics. That this actually works is proven by appealing to singular perturbation theory.

In practice though, the fast dynamics seldom has to be any faster than the sliding mode dynamics. The attempts to prove convergence by Lyapunov theory (and without recourse to singular perturbation theory) was historically one of the roads to backstepping.

The sliding mode idea to "separate the timescales" of convergence för different state variables can be combined with Lyapunov techniques to form a variant of backstepping, which we will however not discuss in this report.

3.2.2 Nonlinear Dynamic Inversion

Consider the following dynamical system

$$\begin{aligned}\dot{x}_1 &= v_1\\ \dot{x}_2 &= v_2\\ \dot{v}_1 &= \sum_{j,k=1}^{2} -\Gamma^1_{jk}(x_1,x_2)\,v_j v_k + F_1(x_1,x_2) + u_1\\ \dot{v}_2 &= \sum_{j,k=1}^{2} -\Gamma^2_{jk}(x_1,x_2)\,v_j v_k + F_2(x_1,x_2) + u_2\end{aligned}$$

which might represent a fully actuated two dimensional mechanical system, such as a robot arm with two motorized joints. It is straight forward to apply the idea above of cancelling the nonlinear terms (and replacing them with any desired other expression) by the insignals u_1 and u_2. This method has been used routinely in robotics since the beginning eighties under the name *nonlinear dynamic inversion*, and may be applied whenever there is an independent control variable for each nonlinear equation.

3.3 Feedback Linearization

Feedback linearization may be described as the systematic procedure of first transforming the system equations into a form where nonlinear dynamic inversion may be applied and then applying nonlinear dynamic inversion. In the introductory toy example, the transformation from (x,y) to (x,z) achieved precisely this.

The systematic procedure is as follows. Consider a dynamical system of the form

$$\begin{aligned}\dot{x}_i &= f_i(x) + g_i(x)\,u\\ (i &= 1..n)\end{aligned}$$

where $(x_1,..,x_n)$ are the state variables and u is a scalar control variable. It is convenient to think of the equation coefficients $f_i(x)$ and $g_i(x)$ as coefficient of first order homogeneous partial differential operators

$$\begin{aligned}L_f &= \sum_{i=1}^{n} f_i(x)\frac{\partial}{\partial x_i}\\ L_g &= \sum_{i=1}^{n} g_i(x)\frac{\partial}{\partial x_i}\end{aligned}$$

In differential geometric jargon, L_f and L_g are vector fields. The i:th component $f_i(x)$ may be retrieved from the vector field L_f by letting it act as a differential operator on the i:th coordinate function x_i. The commutator $[L_f, L_g] = L_f\ L_g - L_g\ L_f$ is also a vector field, since all second order terms cancel. The i:th components of $[L_f, L_g]$ are given by

$$\sum_{j=1}^{n}\left(f_j(x)\frac{\partial g_i}{\partial x_j} - g_j(x)\frac{\partial f_i}{\partial x_j}\right)$$

The usefulness of this operator representation is that it is coordinate independent, so if for a certain system it holds that *e.g.* $[L_f, L_g] = 3L_g$, then this relation holds in any coordinate representation, and thereby reveals an intrinsic property of the dynamical system which cannot be changed by any change of coordinates.

3.3.1 Example:

As an example of how simple things become when the coordinate crutches are thrown away, consider the following normal form problem. Find, if possible, a change of coordinates

$$\begin{aligned} \xi &= \xi(x,y) \\ \eta &= \eta(x,y) \end{aligned}$$

which turns our toy example

$$\begin{cases} \dot{x} = x^2 + y \\ \dot{y} = x + u \end{cases}$$

into linear form

$$\dot{\Xi} = A\Xi + Bu$$

where $\Xi = \begin{pmatrix} \xi & \eta \end{pmatrix}^T$.

It is clear that such a linear system would satisfy an identity of the type $[L_f, [L_f, L_g]] = k_1 L_g + k_2 [L_f, L_g]$, for some constants k_1 and k_2 (this is simply the fact that A^2B lies in the span of B and AB, which holds for any two dimensional linear system). But for the toy system

$$\begin{aligned} L_f &= \left(x^2 + y\right)\frac{\partial}{\partial x} + x\frac{\partial}{\partial y} \\ L_g &= \frac{\partial}{\partial y} \\ [L_f, L_g] &= -\frac{\partial}{\partial x} \\ [L_f, [L_f, L_g]] &= 2x\frac{\partial}{\partial x} + \frac{\partial}{\partial y} \end{aligned}$$

so one of the coefficients is not constant. Hence *no* change of coordinates turns the toy system into a linear system.

For notational convenience, we write $[L_f, L_g]$ as $ad_f g$ and $[L_f, [L_f, L_g]] = ad_f^2 g$ and so on.

Now that we have seen how the vector field point of view helps us in showing that a certain system is *not* linearizable by a change of coordinates only, we turn to the problem of showing necessary (and as we shall see also sufficient) conditions for a system to be linearizable by a change of coordinates together with a feedback transformation.

The system has the form

$$\begin{aligned} \dot{x}_i &= f_i(x) + g_i(x)\,u \\ (i &= 1..n) \end{aligned}$$

A *feedback transformation* is a control law of the type

$$u = U(x) + K(x)v$$

with a nowhere vanishing $K(x)$. The resulting dynamics reads

$$\dot{x}_i = \hat{f}_i(x) + \hat{g}_i(x)\,v$$

where

$$\begin{aligned} \hat{f}_i(x) &= f_i(x) + U(x)g_i(x) \\ \hat{g}_i(x) &= K(x)g_i(x) \end{aligned}$$

For the origin to be a possible closed loop equilibrium point, it is necessary that $f_i(0) = g_i(0) u_0$ for some u_0, so we assume that this is the case. An initial feedback transformation $u = -u_0 + v$ transforms the problem into a similar problem but one for which (the new) $f_i(x)$ vanishes at the origin. So by no loss of generality we may assume that this has already been done and that $f_i(0) = 0$. We also assume a controllability property of the system: it is assumed that the linear approximation of the system at the origin is controllable. It is then clear that this will hold also for the system itself, provided that there exists a feedback linearization transformation. Hence, if there is a feedback linearization transformation of the system, the closed loop form of the system may be put (in suitable linear coordinates and possibly after an auxiliary linear feedback transformation) in the controllable standard (Brunovsky) form

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= y_3 \\ &.. \\ \dot{y}_n &= v \end{aligned}$$

where v is the residual control variable. This means that in those coordinates,

$$L_g = g_y(y) \frac{\partial}{\partial y_n}$$

for some function $g_y(y)$ since a feedback transformation leaves the control vector field L_g invariant except for a rescaling by $K(x)$. Likewise, the (original) drift vector field L_f must be of the form

$$L_f = y_2 \frac{\partial}{\partial y_1} + y_3 \frac{\partial}{\partial y_2} + .. + f_y(y) \frac{\partial}{\partial y_n}$$

for some function $f_y(y)$ since the feedback transformation only affects L_f in the L_g direction.

We may collect our findings as statements about the function $h(x) = y_1$ and the vector fields L_f and L_g:

- There is a function $h(x)$ (nonstationary at 0) such that the functions $L_f^k h$, $(k = 0, ..n-1)$ are functionally independent and such that
- $L_{ad_f^k g} h = 0$, $(k = 0, ..n-2)$ and $L_{ad_f^{n-1} g} h$ is nonvanishing at the origin. (It equals $g_y(y)$.)

Conversely, it is also clear that if these conditions are fulfilled, then a change of coordinates to $y_k = L_f^{k-1} h$, $(k = 1, ..n)$ brings the system to the normal form

$$\begin{aligned} L_f &= y_2 \frac{\partial}{\partial y_1} + y_3 \frac{\partial}{\partial y_2} + .. + f_y(y) \frac{\partial}{\partial y_n} \\ L_g &= g_y(y) \frac{\partial}{\partial y_n} \end{aligned}$$

which allows for nonlinear dynamic inversion by means of the control law

$$u = \frac{-f_y(y) + v}{g_y(y)}$$

Now, f and g were given, but the function $h(x)$ has to be found – if it exists. The condition that $L_{ad_f^k g} h = 0$, $(k = 0, ..n-2)$ determines the differential dh up to a scalar factor:

$$dh = \lambda \theta$$

where θ is any differential 1-form representing the common (algebraic) nullspace of the $ad_f^k g$. (Such a θ may be found by linear algebra alone.) The question remains whether or not this 1-form has an integrating factor λ, thereby allowing for a function h. The necessary and sufficient condition for this is that the differential 3-form $\theta \wedge d\theta$ identically vanishes. (This condition, *Frobenius theorem*, may also be expressed in terms of vector field commutators. The present formulation (by Cartan) is often computationally advantageous.) In components, this condition reads

$$\theta_i \left(\frac{\partial \theta_j}{\partial x_k} - \frac{\partial \theta_k}{\partial x_j} \right) + \theta_j \left(\frac{\partial \theta_k}{\partial x_i} - \frac{\partial \theta_i}{\partial x_k} \right) + \theta_k \left(\frac{\partial \theta_i}{\partial x_j} - \frac{\partial \theta_j}{\partial x_i} \right) = 0$$

For two-dimensional systems, the Frobenius condition is automatically satisfied, so generically (namely: assuming controllability and functional independence of the $L_f^k h$) such systems may be feedback transformed to linear form, like our toy example.

In dimensions ≥ 3 the Frobenius condition severely restricts the class that may be fully feedback linearized. In such cases the possibility remains of partial feedback linearization.

Let $h(x)$ be a state space function that we want to control. Assume that $L_{ad_f^k g} h = 0$ for $k = 0, .., r-2$ but $L_{ad_f^{r-1} g} h \neq 0$ for some integer r, known as the relative degree of the system. It is possible to change to coordinates y_k such that $y_k = L_f^{k-1} h$ when $1 \leq k \leq r$. In such coordinates

$$\begin{aligned}
\dot{y}_1 &= y_2 \\
&\ldots \\
\dot{y}_{r-1} &= y_r \\
\dot{y}_r &= f_r(y) + g_r(y) u \\
\dot{y}_{r+1} &= f_{r+1}(y) + g_{r+1}(y) u \\
&.. \\
\dot{y}_n &= f_n(y) + g_n(y) u
\end{aligned}$$

and by a suitable choice of the functions y_k $(r+1 \leq k \leq n)$, it can be arranged that $g_{r+1}(y) = .. = g_n(y) = 0$. The coordinates so obtained put the system into the so-called normal form. We assume that this has been done, and apply the nonlinear dynamic inversion feedback transformation

$$u = \frac{-f_r(y) + v}{g_r(y)}$$

The resulting dynamics is given by

$$\begin{aligned}
\dot{y}_1 &= y_2 \\
&\ldots \\
\dot{y}_{r-1} &= y_r \\
\dot{y}_r &= v \\
\dot{y}_{r+1} &= f_{r+1}(y) \\
&.. \\
\dot{y}_n &= f_n(y)
\end{aligned}$$

which means that the variables $(y_1, .., y_r)$ serve as state variables of a linear system (an integrator chain) independent of the remaining state variables

$y_{r+1}..y_n$, which are related to the so-called the *zero-dynamics* of the system, *i.e.* the remaining dynamics when $y_1 = .. = y_r = 0$ identically in time,

$$\begin{aligned} \dot{y}_{r+1} &= f_{r+1}\left(0, .., 0, y_{r+1}, .., y_n\right) \\ &.. \\ \dot{y}_n &= f_n\left(0, .., 0, y_{r+1}, .., y_n\right) \end{aligned}$$

The zero-dynamics is clearly unaffected by feedback transformations and is an intrinsic property of the original system and the function $h(x)$.

If the system is controlled in such a way that $y_1 = .. = y_r = 0$ becomes an invariant submanifold, (*i.e.* the linear $(y_1, .., y_r)$ system is stabilized as such), it is clear that the closed loop system cannot be stable if the zero-dynamics is unstable, but it is in fact also true that stability of the linear system and of the zero-dynamics is sufficient for stability.

It is however possible that the remaining variables $(y_{r+1}, .., y_n)$ may be stabilized by feedback, even if the zero dynamics is unstable, namely by a feedback law for which $y_1 = .. = y_r = 0$ is not invariant.

3.4 Backstepping

Feedback linearization above was done in two steps: first find the normal form, then perform a nonlinear dynamic inversion cancelling the nonlinearities. The second step may be harmful to the robustness of the system, since it may include the cancellation of stabilizing nonlinearities. It it often wise to sacrifice linearity for robustness in the procedure.

Consider a system in the "triangular form"

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2, x_3) \\ &... \\ \dot{x}_n &= f_n(x_1, x_2, .., x_n) + \\ &\quad g_n(x_1, x_2, .., x_n)u \end{aligned}$$

The algorithms of backstepping may be applied directly to a system of this form, and this is a useful feature in practice. In the first equation, $x_1 \equiv y_1$ is the state variable and x_2 is considered as a virtual control, a Lyapunov function $V_1(y_1)$ is chosen and a corresponding "desired" stabilizing control law $x_2^{des}(y_1)$ is found by inspection. A new coordinate $y_2 = x_2 - x_2^{des}(x_1)$ and a new term $V_2(y_1)$ in the Lyapunov function is chosen, $x_3^{des}(y_1, y_2)$ is found, and so on.

The existence of coordinates so that the system is triangular are equivalent to the conditions for full feedback linearizability, so any such system may be transformed into

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &... \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= f_n(x_1, x_2, .., x_n) + \\ &\quad g_n(x_1, x_2, .., x_n)u \end{aligned}$$

before the backstepping procedure begins. Since the first $n-1$ equations look the same for all systems, one does not have to go through the recursive constructions of Lyapunov terms and controls each time. A systematic procedure is provided by the passive normal form.

First, a linear transformation (see [40])

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_1 + x_2 \\ y_1 &= 2x_1 + 2x_2 + x_3 \\ &\ldots \end{aligned}$$

turns the dynamics into

$$\begin{aligned} \dot{y}_1 &= -y_1 + y_2 \\ \dot{y}_2 &= -y_2 - y_1 + y_3 \\ &\ldots \\ \dot{y}_{n-1} &= -y_{n-1} - y_{n-2} + y_n \\ \dot{y}_n &= -y_n - y_{n-1} + \\ &\tilde{f}(y_1, y_2, .., y_n) + \\ &\tilde{g}(y_1, y_2, .., y_n)u \end{aligned}$$

In this "passive normal form", the system may be considered as the passive interconnection of n passive subsystems, each enjoying an excess passivity margin (a so-called $OFP(1)$ property). This margin is the key to several desirable properties such as robustness and good tracking properties. The robustness properties induced by a passive normal form based design are shown in [40]. The tracking properties will be discussed below.

First, let us give a standard controller design for systems in passive normal form. The function

$$V = \frac{1}{2}\left(y_1^2 + y_2^2 + .. + y_n^2\right)$$

satisfies

$$\dot{V} = -2V + y_n\tilde{f} + y_n\tilde{g}u$$

Now choose u according to Sontag's formula

$$u = \frac{-1}{\tilde{g}}\left(\tilde{f} + sgn(y_n)\sqrt{\tilde{f}^2 + y_n^2\tilde{g}^4}\right)$$

whenever $y_n\tilde{g} \neq 0$ and $u = 0$ otherwise.

It then holds that

$$\dot{V} = -2V - \sqrt{y_n^2\tilde{f}^2 + y_n^4\tilde{g}^4} \leq -2V$$

which implies exponentially asymptotic stability.

Returning to our toy example

$$\begin{cases} \dot{x} = -x + z \\ \dot{z} = x + 2x(z-x) + u \end{cases}$$

we may consider this as the passive normal form if we identify $\tilde{f}$ with $2x + z + 2x(z-x)$ and $\tilde{g}$ with 1.

A passive normal form based design then becomes

$$u = -2x - z - 2x(z-x) - sgn(z)\sqrt{z^2 + (-2x^2 + z + 2x(z-x))^2}$$

The dynamics of the Lyapunov function $V = \frac{1}{2}x^2 + \frac{1}{2}z^2$ is improved and becomes

$$\dot{V} \leq -2V$$

3.5 Tracking Properties

As a first step towards tracking, we construct a set-point controller. Consider again a system in normal form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\ldots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= f + g\ u \end{aligned}$$

Suppose that we want to stabilize the point $x_1 - p = x_2 = x_3 = .. = x_n = 0$, where p is a constant set-point value.

We introduce the passive normal form

$$\begin{aligned} y_1 &= x_1 - p \\ y_2 &= x_1 + x_2 - p \\ y_3 &= 2x_1 + 2x_2 + x_3 - 2p \\ &\ldots \\ y_k &= B_{k,1}(x_1 - p) + B_{k,2}x_2 + .. + B_{k,k}x_k \\ &\ldots \\ y_n &= B_{n,1}(x_1 - p) + B_{n,2}x_2 + .. + B_{n,n}x_n \end{aligned}$$

where the coefficients $B_{i,j}$ are obtained recursively by the formula

$$B_{i,j} = B_{i-1,j} + B_{i-2,j} + B_{i-1,j-1}$$

Thereby the system is transformed into the form

$$\begin{aligned} \dot{y}_1 &= -y_1 + y_2 \\ \dot{y}_2 &= -y_2 - y_1 + y_3 \\ &\ldots \\ \dot{y}_{n-1} &= -y_{n-1} - y_{n-2} + y_n \\ \dot{y}_n &= -y_n - y_{n-1} + \tilde{f} + \tilde{g}\ u \end{aligned}$$

where

$$\begin{aligned} \tilde{f} &= B_{n+1,1}(x_1 - p) + \sum_{j=2}^{n} B_{n+1,j}x_j + f \\ \tilde{g} &= g \end{aligned}$$

Finally, the control u is chosen by means of Sontag's formula, as in the above example. This provides us with a set-point controller with good passivity properties for any fixed p.

This formula may also be used as a *tracking problem control law.* The "parameter" p in now allowed to vary in time. It turns out that the tracking system satisfies a useful differential inequality for the Lyapunov function V. As each variable y_k satisfies

$$\dot{y}_k = Y_k(y) - B_{k,1}\dot{p}$$

where $Y_k(y)$ is the expression from the set-point point dynamics (for $2 \leq k \leq n-1$, it is $-y_k - y_{k-1} + y_{k+1}$) Hence

$$\dot{V} = \sum_{j=1}^{n} y_k Y_k(y) - \dot{p}\sum_{j=1}^{n} y_k B_{k,1} \leq -2V + |\dot{p}|\, b_n\sqrt{V}$$

where

$$b_n = \sqrt{2\sum_{k=1}^{n}(B_{k,1})^2}$$

In terms of $U = \sqrt{V}$, this may also be written as

$$\dot{U} \leq -U + |\dot{p}|\, b_n/2$$

from which integral estimates easily follow by Grönwall's lemma. If the reference signal $p(t)$ is Lipschitz bounded by a constant K, it holds that U (and hence V) must decrease whenever $U > b_n K/2$ and approach the set $U \leq b_n K$. So by using the passive normal form, backstepping and the above set-point regulator design, a tracking controller with known performance properties is obtained.

3.6 Tracking and Zero Dynamics

Consider a system in normal form

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
&\dots \\
\dot{x}_{r-1} &= x_r \\
\dot{x}_r &= f_r(x,\xi) + g_r(x,\xi)\,u \\
\dot{\xi}_{r+1} &= \varphi_{r+1}(x,\xi) \\
&.. \\
\dot{\xi}_n &= \varphi_n(x,\xi)
\end{aligned}$$

where we have split the coordinates into two groups, x_i $(i = 1,,r)$ and ξ_α $(\alpha = r+1,,n)$. The tracking controller described above may be constructed for the x_i -system. Denote by $x_{eq}(p)$ the point $x_1 - p = x_2 = .. = x_r = 0$. Assume that the "zero dynamics"

$$\dot{\xi} = \varphi(x_{eq}(p), \xi)$$

has a stable equilibrium $\xi_{eq}(p)$ for every value p, and that $W(p,\xi)$ is a Lyapunov function for this zero dynamics. It then holds that

$$\dot{W} = \frac{\partial W}{\partial p}\dot{p} + \frac{\partial W}{\partial \xi}\varphi(x,\xi)$$

In this equation each component $\varphi_\alpha(x,\xi)$ may be expressed by the mean value theorem

$$\varphi_\alpha(x,\xi) = \varphi_\alpha(x_{eq},\xi) + \sum_{\alpha=r+1}^{n}\sum_{i=1}^{r}\frac{\partial W}{\partial \xi_\alpha}\frac{\partial \varphi_\alpha}{\partial x_i}\left(x_{(\alpha)},\xi\right)(x - x_{eq})$$

where for each p and α, there is a θ such that $0 < \theta < 1$ and $x_{(\alpha)} = (1-\theta)x_{eq} + \theta x$.

Under the assumptions that $\dot{W} \leq 2\gamma W$ for the zero dynamics, and that

$$\begin{aligned} \left\|\frac{\partial W}{\partial \xi}\right\| &\leq k_1\sqrt{2W} \\ \left\|\frac{\partial \varphi}{\partial x}\right\| &\leq k_2 \\ \left\|\frac{\partial W}{\partial p}\right\| &\leq 2k_3\sqrt{W} \end{aligned}$$

the following system of differential inequalities is obtained in terms of U and $\Upsilon = \sqrt{W}$

$$\begin{aligned} \dot{U} &\leq -U + |\dot{p}|\, b_n/2 \\ \dot{\Upsilon} &\leq -\gamma\Upsilon + k_1k_2U + k_3|\dot{p}| \end{aligned}$$

which gives control of the zero dynamics behavior in the tracking problem. If $|\dot{p}| \leq 2K$, the system will end up in the time dependent set where

$$\begin{aligned} U &\leq b_nK \\ \Upsilon &\leq \frac{K}{\gamma}(k_3 + k_1k_2b_n) \end{aligned}$$

4 Optimization based control of nonlinear systems

The aim of the presentation in this chapter is to provide a short introduction to optimization based control synthesis methods. In particular for application to nonlinear systems.

4.1 Gradient based optimization for open-loop robust control

An efficient method for solving an optimization problem of high dimension, where there are many degrees of freedom of the control is to utilize gradient information of the objective functional. Many standard tools exist for exploiting gradient information in order to find an optima through an iterative procedure, but the computation of the gradient itself can be very computationally demanding in some applications. A popular method in e.g. aerodynamic shape and structure optimization is to use an adjoint equation, defined by an inner product similar to the objective functional, in order to compute gradient information. Consider the nonlinear system,

$$\begin{aligned} \mathcal{N}\dot{\mathbf{x}} &= \mathcal{A}(\mathbf{x}, \mathbf{f}, u, w) && \text{on} \quad 0 < t < T \\ \mathbf{x} &= \mathbf{x_0} && \text{at} \quad t = 0, \end{aligned} \tag{4.1}$$

where $\mathcal{N}$ is a matrix that can be a singular, $\mathbf{x}$ is the state, $\mathbf{f}$ is some known external force and u, w represent controls and unknown disturbances respectively.

The goal is then to compute a control signal u that achieves a desired objective for the worst case disturbance w, as measured by the cost function,

$$\mathcal{J} = \frac{1}{2}\int_0^T (\mathbf{x}^* Q \mathbf{x} + \ell^2 u^* u - \gamma^2 w^* w) \mathrm{d}t. \tag{4.2}$$

4.1.1 Computing the gradient

If $\mathcal{A}$ and $\mathcal{N}$ are such that small perturbations δu to the control u and δw to the disturbance w result in small perturbation $\delta\mathbf{x}$ of the state $\mathbf{x}$ a linearized perturbation equation can be formulated as,

$$\begin{aligned} \mathcal{L}\delta\mathbf{x} &= B_u \delta u + B_w \delta w && \text{on} \quad 0 < t < T \\ \delta\mathbf{x} &= 0 && \text{at} \quad t = 0, \end{aligned} \tag{4.3}$$

where $\mathcal{L} = (\mathcal{N}d/dt - A)$, B_u and B_w are found through linearization of (4.1). Similarly the cost function perturbation resulting from δu and δw can be computed by

$$\delta\mathcal{J} = \int_0^T (\mathbf{x}^* Q \delta\mathbf{x} + \ell^2 u^* \delta u - \gamma^2 w^* \delta w) \mathrm{d}t = \int_0^T \Big((\nabla_u \mathcal{J})^* \delta u + (\nabla_w \mathcal{J})^* \delta w \Big) \mathrm{d}t. \tag{4.4}$$

By introducing the inner product,

$$\langle \mathbf{a}, \mathbf{b} \rangle = \int_0^T \mathbf{a}^* \mathbf{b} \, \mathrm{d}t,$$

we can formulate the relation,

$$\langle \mathbf{p}, \mathcal{L}\delta\mathbf{x} \rangle = \langle \mathcal{L}^* \mathbf{p}, \delta\mathbf{x} \rangle + \mathbf{h},$$

where $\mathcal{L}^*\mathbf{p} = \Big(-\mathcal{N} d/dt - A^* \Big) \mathbf{p}$ and $\mathbf{h} = \mathbf{p}^* \mathcal{N} \delta\mathbf{x}|_{t=T} - \mathbf{p}^* \mathcal{N} \delta\mathbf{x}|_{t=0}$. If we then introduce the $a$djoint equation,

$$\begin{array}{ll} \mathcal{L}^*\mathbf{p} = Q\mathbf{x} \Rightarrow -\mathcal{N}^* \dot{\mathbf{p}} = A^*\mathbf{p} + Q\mathbf{x} & \text{on} \quad 0 < t < T \\ \quad \mathbf{p} = 0 & \text{at} \quad t = T, \end{array} \tag{4.5}$$

that is to be solved "backwards" in time, we can combine the equations to identify the gradients. Utilizing the relation obtained from (4.3) and (4.5),

$$\langle \mathbf{p}, B_u \delta u + B_w \delta w \rangle = \langle Q\mathbf{x}, \delta\mathbf{x} \rangle,$$

we can write (4.4) as

$$\delta \mathcal{J} = \int_0^T \Big((B_u^* \mathbf{p} + \ell^2 u)^* \delta u + (B_w^* \mathbf{p} - \gamma^2 w)^* \delta w \Big) \mathrm{d}t.$$

from which we can identify the gradient expressions,

$$\nabla_u \mathcal{J} = B_u^* \mathbf{p} + \ell^2 u \quad \text{and} \quad \nabla_w \mathcal{J} = B_w^* \mathbf{p} - \gamma^2 w.$$

These gradient expressions can be evaluated by first solving (4.1) and then (4.5). This is then repeated in an iterative algorithm to find the solution to the min max problem. In the linear setting one can identify the optimal solution directly by setting the gradients to zero and substituting the resulting expressions into equations (4.1) and (4.5). Combining these two expression, assuming there is a linear map from $\delta\mathbf{x}$ to $\mathbf{p}$ results in a matrix differential equation, or Riccati equation, that may be solved using standard techniques.

4.2 Dynamic games

In the literature the term "game theory" is used in a number of different meanings encapsulation static, dynamic and differential games. The definition of a dynamic game used here is adopted from [3]:

> ... if the order in which decisions are made is important, ...

or alternatively,

> ... at least one player is allowed to use a strategy that depends on previous actions.

Also adopted from the same source is definition of a non-cooperative game:

> ... if each person involved pursues his or her own interests which are partly conflicting with others.

In typical duel situations there are two players with conflicting interests and with the ability to react to each others actions. One crucial issue is the availability and accuracy of the information regarding the opponent and his actions. A strategy for a player can therefore involve minimizing the information available to the opponent in order to gain a favorable position.

In a zero-sum formulation of a dynamic game the players aim to maximize respectively minimize the same objective or cost. Examples of objectives for a missile-target duel are miss-distance and time-to-intercept. A zero-sum formulation is preferred since it can be formulated as a min-max problem, though it is not unreasonable that the players objectives are different and even dynamic. It is also questionable if the assumption that each player is aware of the other players objectives, and thereby also their optimal strategy, is valid.

Making the assumption that there is no way of planning ahead for all possible events in a game it is reasonable to have some dynamics in the strategy of play. Say that given the currently available information it is possible to device a strategy for the remainder of the game. Then if there is new information available at some future time, it should be incorporated and the strategy modified accordingly. Take as an example the game of chess where a player has the opportunity to change his strategy based on the information obtained every time the opponent has made a move. If the move is unexpected a reexamination of the strategy is possible before the next step. This kind of discrete points of decision making is not always present in a game but in a situation where events take place at unknown times it is certainly applicable. Utilizing controllers based on the described framework is referred to as receding horizon control or sometimes model predictive control (MPC) or nonlinear model predictive control (NMPC).

Historically game theory has had most success through applications in economics. As a method to develop control strategies it is today most widely used in academical applications due to the need for computational resources. If the processors become increasingly efficient and modern methods for model order reduction, as described in chapter 2, are employed to get simplified models of nonlinear systems they can become very beneficial for controlling complex systems also in practice.

Characteristic features of (N)MPC are that it allows a nonlinear model to be used for prediction, explicit consideration of constraints, optimization of performance criteria id performed online. Drawbacks are that an open-loop optimal control problem must be solved online and that system states must be measured or estimated.

4.2.1 Missile target duel example

As an illustrative scenario we consider a missile-target duel example one can imagine that the target suddenly changes his strategy. Say that the target suddenly drops his payload and increases performance. The missile can thus, once aware of the event, change its strategy to adapt the the new behavior of the target. This example admittedly requires a very advanced missile logic but is still not unrealistic in systems with high performance target estimators. An alternative to this event-driven strategy change is to incorporate all possible events into a robust or perhaps adaptive guidance law design. This would result in a guidance law that could handle all such events, but that would perform far from optimal in most situations since it will have to be conservative.

If we at a given time t (see Figure 4.1) can assume that we have a reliable model of both missile and target dynamics as well as a qualified guess of the current objective we can compute the optimal missile trajectory, given a suitable formulation of the objective, for the worst case target maneuver until the

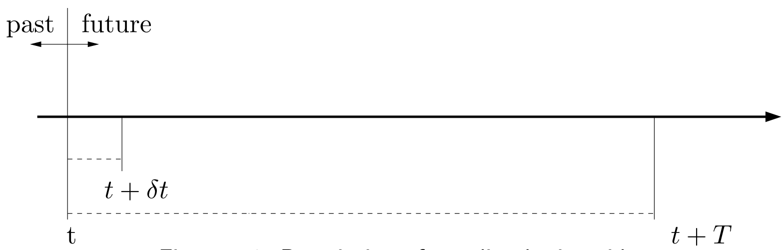


Figure 4.1: Description of receding horizon idea.

time $t+T$. We then let the missile follow this trajectory for a period of time δt while gathering more information about the target. Then at a new point in time $t+\delta t$ we repeat the procedure given the current situation. In doing so the missile will periodically adapt its strategy to the actual target maneuvers and thus there will be some adaptation due to unexpected/unpredictable events given that these are captured by the missile sensors and logic. The procedure described can at each instant be formulated as a (nonlinear) robust control problem.

One difficult issue is the formulation of a suitable objective that will actually result in a good trajectory as well as a solvable saddle point optimization problem. To be able to prove stability of a receding horizon controller in closed-loop, a penalty on the terminal state (at the end of the optimization horizon) is often required as a part of the objective.

4.2.2 Mathematical formulation

The formulation of the general problem can be described as follows. Considering a class of systems described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0$$

where,

$$\mathbf{u}(t) \in \mathcal{U}, \forall t > 0 \quad \mathbf{x}(t) \in \mathcal{X}, \forall t > 0,$$

with $\mathcal{X} \subseteq R^n$ and $\mathcal{U} \subseteq R^m$. Under the assumptions that $\mathcal{U} \subset R^p$ is compact, $\mathcal{X}$ is connected and $(0,0) \in \mathcal{X} \times \mathcal{U}$ with continuity of the vector field $\mathbf{f}$, which is locally Lipschitz continuous in $\mathbf{x}$, and finally that $\mathbf{u}$ is piecewise continuous, the problem is to find

$$\min_{\bar{\mathbf{u}}(\,\cdot\,)} J(\mathbf{x}(t), \bar{\mathbf{u}}(\,\cdot\,); T_c, T) \quad \text{with the control horizon } T_c \leq T,$$

where,

$$J(\mathbf{x}(t), \bar{\mathbf{u}}(\,\cdot\,); T_c, T) = \int_t^{t+T} F(\bar{\mathbf{x}}(s), \bar{\mathbf{u}}(s))\,\mathrm{d}s$$

subject to,

$$\begin{aligned}
&\dot{\bar{\mathbf{x}}}(s) = \mathbf{f}(\bar{\mathbf{x}}(s), \bar{\mathbf{u}}(s)), && \bar{\mathbf{x}}(t) = \mathbf{x}(t) \\
&\bar{\mathbf{u}}(s) \in \mathcal{U}, && \forall s \in [t, t+T] \\
&\bar{\mathbf{u}}(s) = \bar{\mathbf{u}}(s+T_c), && \forall s \in [t+T_c, t+T] \\
&\bar{\mathbf{x}}(s) \in \mathcal{X}, && \forall s \in [t, t+T].
\end{aligned}$$

This is a classical optimization problem for which many efficient solution methods exist, and in the linear setting using a quadratic objective function J, the solution can be found by solving a matrix Riccati equation.

In the case of a problem formulated as a non-cooperative dynamic game the problem is to find a saddle point rather than a minima for a class of systems described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0$$

where,

$$\mathbf{u}(t) \in \mathcal{U}, \forall t > 0 \quad \mathbf{w}(t) \in \mathcal{W}, \forall t > 0 \quad \mathbf{x}(t) \in \mathcal{X}, \forall t > 0,$$

with $\mathcal{X} \subseteq R^n, \mathcal{U} \subseteq R^m$ and $\mathcal{W} \subseteq R^p$. The problem is then to find

$$\max_{\bar{\mathbf{w}}(\,\cdot\,)} \min_{\bar{\mathbf{u}}(\,\cdot\,)} J(\mathbf{x}(t), \bar{\mathbf{u}}(\,\cdot\,), \bar{\mathbf{w}}(\,\cdot\,); T_c, T) \quad \text{with the control horizon } T_c \leq T.$$

The problem of finding a saddle point is generally more difficult that finding an optima, but in the linear setting there are well established methods based on the solution of Riccati equations within the framework of $\mathcal{H}_\infty$ control.

5 Applications of nonlinear control theory

5.1 Design of a Nonlinear Autopilot for Velocity and Attitude Control Using Block Backstepping

Using the powerful ideas from multi-input, or block (or vector), backstepping we construct a controller for simultaneous control of attitude and velocity in three dimensions for an aircraft described by the full nonlinear six degrees-of-freedom equations for rigid body motion. The proposed controller has a very simple structure and has global stabilizing properties in attitude and local stabilizing properties in velocity [32]. A key (and novel) element of the controller is the use of a spherical linear interpolation (slerp) to determine a geodesic on the manifold of unit norm quaternions, which represents the minimal rotation required to control the attitude. The attitude control is combined with a velocity control employing rotation of the airspeed vector and thrust magnitude control to stabilize angle of attack, the sideslip angle and the absolute velocity to desired values. Only weak assumptions about the aerodynamic forces (i.e. the aerodynamic configuration of the aircraft) are necessary for application of the controller. We illustrate the behavior of the controller with simulations using a implementation in the Modelica language of the ADMIRE model which represents a small single engine fighter aircraft (similar to the JAS39).

Introduction

The motivation for the work reported here is twofold: First, the backstepping design methodology has recently emerged as a powerful alternative to existing nonlinear design methods and has shown great promise in various vehicle control problems [10],[22],[9]. A natural design challenge in aircraft control is therefore to apply backstepping to the design of a full three-dimensional autopilot for an aircraft described by a realistic model. This is a key problem in the control of Unmanned Aerial Vehicles (UAVs). Second, the choices in software tools for a modeler/designer/analyst in control is by now quite large and it is possible to perform the whole chain (or rather, iterative loop) of modeling, analysis/design, simulation and code generation for e.g a target embedded system using a single tool, or a small number of tools, running on a personal computer. In particular, the emergence of integrated development environments for object oriented high level modeling/simulation languages have made it possible to very quickly design, analyze and evaluate (nonlinear) controllers for various aerospace applications with subsequent code generation for tests in e.g. a flight simulator. One such environment is Dymola of Dynasim AB featuring the modeling language Modelica [15],[42] which has been evaluated in this study and in which our aircraft model, trimming routine and controller have been implemented.

The problem of stabilizing the motion of a rigid body is central to the aerospace control literature but the vast majority of the works on the subject treat the linearized case [38]. Solutions to the nonlinear problem have been given in various settings in the control theory literature, see e.g. [2] and the references therein, but they focus mostly on the particular case encountered in satelite control where usually no external forces are present except gravity and the possibility for control action may be limited (e.g. due to malfunctioning actuators). Recently, however, Glad and Härkegård [17] have given a solution for a variant of the velocity and angular velocity control problem for a rigid body in the aircraft control setting using multi-input, or block, backstepping. They assume, however, that the thrust force of the engine is always aligned along the velocity vector and this is clearly not the case in general (unless thrust vectoring is employed). We present a new solution to the simultaneous attitude and velocity control problem that employs a velocity control similar to that of Glad and Härkegård but which is capable of handling a thrust vector that is not necessarily aligned with the velocity vector and which is complemented by a quaternion based attitude control. The attitude control utilizes spherical linear interpolation (slerp) on the sphere $\mathcal{S}^3 \subseteq \mathbb{R}^4$ to compute a geodesic representing the minimal rotation of the body needed to control the attitude to the desired value. The velocity control employs rotation of the airspeed vector and thrust control to stabilize angle of attack, the sideslip angle and the absolute velocity.

Our presentation is organized as follows. In the next section we introduce the equations that we are going to use to describe the motion of an aircraft around a reference flight condition, such as a straight flight path in some direction in space or a constant angular velocity turn. After this we introduce in Section 5.1.2 the mathematical formulation of the nonlinear control problem and we present our solution. This is done in a series of steps, beginning with a short review of the necessary tools from backstepping theory, proceeding with a description on how the autopilot control problem can be cast in the standard form for integrator backstepping, and ending with a presentation of the controller. (A proof of stability of the controller can be found in the paper [32].) Finally, in Section 5.1.3 we give a short introduction to the Modelica implementation of the aircraft model and we show some simulation results illustrating the behavior of the controller.

5.1.1 Equations of Motion

In the equations of motion for the aircraft two reference frames are present, one frame F_e fixed in the earth which we assume to be an inertial frame, and one frame f_b fixed in the center of gravity [1] (CoG) of the aircraft. The translational and rotational velocities for the aircraft expressed in F_e and f_b are related by

$$\begin{aligned} \mathbf{V} &= \mathbf{R}\mathbf{v}, \\ \mathbf{\Omega} &= \mathbf{R}\boldsymbol{\omega}, \end{aligned}$$

where $\mathbf{V}, \mathbf{\Omega}$ are the translational and angular velocities, respectively, in F_e and $\mathbf{v}, \boldsymbol{\omega}$ are their counterparts in f_b. The matrix $\mathbf{R}$ is the rotation matrix involved in the transformation and $\mathbf{Q}$ is the corresponding quaternion.

Kinematics and Dynamics

The equations of motion for the aircraft in F_e and f_b are given by the Newton and Euler equations for rigid body motion, formulated around the center of gravity, and combined with the standard quaternion differential equation for

[1] We assume standard vehicle axis configuration [38] and that the CoG is not moving relative to the aircraft body.

orientation [38], viz.

$$\begin{aligned} \mathbf{V} &= \mathbf{R}\mathbf{v}, && (5.1) \\ \dot{\mathbf{Q}} &= \frac{1}{2}\mathbf{Q} \circ \tilde{\boldsymbol{\omega}}, && (5.2) \\ \mathbf{f} + \mathbf{t} &= m(\dot{\mathbf{v}} + \boldsymbol{\omega} \times \mathbf{v} - \mathbf{g}), && (5.3) \\ \mathbf{m} &= \mathbf{j}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{j}\boldsymbol{\omega}, && (5.4) \end{aligned}$$

where $\tilde{\boldsymbol{\omega}} = (0, \boldsymbol{\omega})$ denotes the pure quaternion formed from the angular velocity vector $\boldsymbol{\omega}$ in f_b and $\circ$ denotes quaternion multiplication. The aerodynamic force $\mathbf{f} = [f_1, f_2, f_3]^T$ and moment vectors [2] $\mathbf{m} = [m_1, m_2, m_3]^T$ in f_b are defined in terms of standard aerodynamic coefficients C_x, C_y, C_z, C_l, C_m, C_n as

$$\mathbf{f} = qS_{ref}[C_x, C_y, C_z]^T, \quad \mathbf{m} = qS_{ref}[b_{ref}C_l, c_{ref}C_m, b_{ref}C_n]^T,$$

where q is the dynamic pressure and b_{ref}, c_{ref}, b_{ref}, S_{ref} are the standard reference lengths and areas occurring in the formulation of the aerodynamic coefficients [38]. The total mass of the aircraft is m, the moment of inertia matrix in f_b is denoted $\mathbf{j}$ and $\mathbf{g}$ is the gravitational acceleration vector in f_b.

The engine dynamics are modeled using a simple first order linear system as

$$\dot{\tau} = b(\tau - u_\tau), \tag{5.5}$$

where τ is the thrust force [3] along the x-axis in f_b, the thrust command is u_τ and b is the value of the time constant (which is set to 0.5 in the simulations below).

Deviations from an equilibrium

In order for (5.2)–(5.4) and (5.5) to be useful for our further developments we must make a change of variables and rewrite these equations in terms of deviations from a reference point. Let

$$[\mathbf{v}_0, \mathbf{Q}_0, \boldsymbol{\omega}_0, \tau_0]^T \tag{5.6}$$

be a vector of reference values for the state variables in (5.2)–(5.4) and (5.5), and let

$$[\mathbf{v}, \mathbf{Q}, \boldsymbol{\omega}, \tau]^T, \tag{5.7}$$

be the vector of deviations from these reference values. We shall consider two types of reference vectors (5.6). The first type is a reference vector corresponding to an equilibrium to the state vector in (5.2)–(5.4) and (5.5). In this case we must necessarily have $\boldsymbol{\omega}_0 = \mathbf{0}$, which corresponds to straight path flight. The second type of reference vector is the one obtained when we have an equilibrium only for the states in (5.3)–(5.4) and (5.5) (i.e. $[\mathbf{v}_0, \boldsymbol{\omega}_0, \tau_0]^T$ is constant), and $\boldsymbol{\omega}_0$ is constant nonzero (so that $\mathbf{Q}_0$ is time varying [4]). This corresponds to a constant-g turn. In this case we shall assume that the reference trajectory $\mathbf{Q}_0$ is the solution to (5.2) corresponding to $\boldsymbol{\omega}_0$, i.e.

$$\dot{\mathbf{Q}}_0 = \frac{1}{2}\mathbf{Q}_0 \circ \tilde{\boldsymbol{\omega}}_0. \tag{5.8}$$

[2] We assume that the engine is aligned so that the thrust does not contribute to the torque around the CoG and we neglect the gyroscopic effects of the engine rotational inertia.

[3] We assume, without loss of generality, that the thrust force acts in the x-axis in f_b only.

[4] The proof of stability in [32] of our controller is applicable, as it stands, only to the case of a constant $\mathbf{Q}_0$. However, as is evident from the simulations, it is also applicable to the case of a slowly time varying $\mathbf{Q}_0$, and it is moreover not hard to extend the proof to the time varying case, with a slight modification of the resulting control law.

Now, a change of variables

$$\begin{bmatrix} \mathbf{v} \\ \mathbf{Q} \\ \boldsymbol{\omega} \\ \tau \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{v}+\mathbf{v}_0 \\ \mathbf{Q}+\mathbf{Q}_0 \\ \boldsymbol{\omega}+\boldsymbol{\omega}_0 \\ \tau+\tau_0 \end{bmatrix}$$

brings the system (5.2)–(5.4) and (5.5) onto the form

$$\begin{aligned} \dot{\mathbf{v}} &= \frac{1}{m}\mathbf{f}^{(a)} + \frac{1}{m}\mathbf{t} + \mathbf{g} + (\mathbf{v}+\mathbf{v}_0) \times (\boldsymbol{\omega}+\boldsymbol{\omega}_0), && (5.9) \\ \dot{\mathbf{Q}} + \dot{\mathbf{Q}}_0 &= \frac{1}{2}(\mathbf{Q}+\mathbf{Q}_0) \circ (\tilde{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}_0), && (5.10) \\ \dot{\boldsymbol{\omega}} &= \mathbf{j}^{-1}(\mathbf{m} - (\boldsymbol{\omega}+\boldsymbol{\omega}_0) \times \mathbf{j}(\boldsymbol{\omega}+\boldsymbol{\omega}_0)), && (5.11) \\ \dot{\tau} &= b(\tau + \tau_0 - u_\tau), && (5.12) \end{aligned}$$

where we have written out the derivative $\dot{\mathbf{Q}}_0$ since it may not be zero. In (5.9)–(5.12) and henceforth the variables $\mathbf{v}, \mathbf{Q}, \boldsymbol{\omega}, \tau$ thus represent deviations from the reference values in (5.6).

We assume that the aerodynamic forces $\mathbf{f}$ are mainly dependent on $\mathbf{v}+\mathbf{v}_0$, the aerodynamic moments $\mathbf{m}$ are mainly dependent on $\mathbf{v}+\mathbf{v}_0, \boldsymbol{\omega}+\boldsymbol{\omega}_0$, and the thrust $\mathbf{t}$ acts only along the aircraft x-axis (i.e. the x-axis in f_b). We can make this dependence explicit by writing

$$\mathbf{f} = \mathbf{f}(\mathbf{v}+\mathbf{v}_0), \quad \mathbf{m} = \mathbf{m}(\mathbf{v}+\mathbf{v}_0, \boldsymbol{\omega}+\boldsymbol{\omega}_0), \quad \mathbf{t} = (\tau+\tau_0)\mathbf{e}_x.$$

Likewise, the gravity vector $\mathbf{g}$ is only dependent on $\mathbf{Q}+\mathbf{Q}_0$ and therefore we can write

$$\mathbf{g} = \mathbf{g}(\mathbf{Q}+\mathbf{Q}_0).$$

It will be convenient to introduce the functions $\tilde{\mathbf{f}}(\mathbf{v}, \mathbf{v}_0)$, $\tilde{\mathbf{m}}(\mathbf{v}, \mathbf{v}_0, \boldsymbol{\omega}, \boldsymbol{\omega}_0)$, and $\tilde{\mathbf{g}}(\mathbf{Q}, \mathbf{Q}_0)$ by

$$\begin{aligned} \tilde{\mathbf{f}}(\mathbf{v}, \mathbf{v}_0) &= \mathbf{f}(\mathbf{v}+\mathbf{v}_0) - \mathbf{f}(\mathbf{v}_0), && (5.13) \\ \tilde{\mathbf{m}}(\mathbf{v}, \mathbf{v}_0, \boldsymbol{\omega}, \boldsymbol{\omega}_0) &= \mathbf{m}(\mathbf{v}+\mathbf{v}_0, \boldsymbol{\omega}+\boldsymbol{\omega}_0) - \mathbf{m}(\mathbf{v}_0, \boldsymbol{\omega}_0), && (5.14) \\ \tilde{\mathbf{g}}(\mathbf{Q}, \mathbf{Q}_0) &= \mathbf{g}(\mathbf{Q}+\mathbf{Q}_0) - \mathbf{g}(\mathbf{Q}_0), && (5.15) \end{aligned}$$

and the matrix function $\mathbf{C}$ taking values in $\mathbb{R}^{3\times 3}$ representing the cross product, so that e.g.

$$\mathbf{C}(\mathbf{v})\boldsymbol{\omega} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_1 \\ \omega_3 \end{bmatrix} = \mathbf{v} \times \boldsymbol{\omega}. \tag{5.16}$$

Moreover, it will be convenient to introduce also a matrix-vector representation for the product of two quaternions. Let $\mathcal{Q}_1 = (a_1, \mathbf{b}_1)$, $\mathcal{Q}_2 = (a_2, \mathbf{b}_2)$ be two quaternions with real parts $a_1, a_2 \in \mathbb{R}$ and imaginary parts $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^3$, respectively. Then the quaternion product $\mathcal{Q}_1 \circ \mathcal{Q}_2$ can be written in terms of an ordinary matrix-vector product as

$$\mathcal{Q}_1 \circ \mathcal{Q}_2 = \mathbf{T}(\mathcal{Q}_1)\mathcal{Q}_2,$$

where the matrix $\mathbf{T}(\mathcal{Q}_1)$ is given by

$$\mathbf{T}(\mathcal{Q}_1) = \begin{bmatrix} a_1 & -\mathbf{b}_1^T \\ \mathbf{b}_1 & \mathbf{C}(\mathbf{b}_1) + a_1\mathbf{I}_{3\times 3} \end{bmatrix}, \tag{5.17}$$

and $\mathbf{C}$ is the skew-symmetric matrix in (5.16) giving the vector product. From $\mathbf{T}(\mathcal{Q}_1)$ we can define a related matrix function $\mathbf{B}(\mathcal{Q}_1)$ as

$$\mathbf{B}(\mathcal{Q}_1) = \begin{bmatrix} -\mathbf{b}_1^T \\ \mathbf{C}(\mathbf{b}_1) + a_1\mathbf{I}_{3\times 3} \end{bmatrix} \tag{5.18}$$

Using (5.13)–(5.15), (5.16) and (5.18) we can rewrite the state equations (5.9)–(5.12) equivalently as

$$\begin{aligned}
\dot{\mathbf{v}} &= \frac{1}{m}\tilde{\mathbf{f}}(\mathbf{v}, \mathbf{v}_0) + \tilde{\mathbf{g}}(\mathbf{Q}, \mathbf{Q}_0) + \mathbf{v} \times \boldsymbol{\omega}_0 + \mathbf{C}(\mathbf{v} + \mathbf{v}_0)\boldsymbol{\omega} + \frac{1}{m}\tau\mathbf{e}_x, && (5.19)\\
\dot{\mathbf{Q}} &= \frac{1}{2}\mathbf{B}(\mathbf{Q})\boldsymbol{\omega}_0 + \frac{1}{2}\mathbf{B}(\mathbf{Q} + \mathbf{Q}_0)\boldsymbol{\omega}, && (5.20)\\
\dot{\boldsymbol{\omega}} &= \mathbf{j}^{-1}(\mathbf{j}\boldsymbol{\omega} \times (\boldsymbol{\omega} + \boldsymbol{\omega}_0) + \mathbf{j}\boldsymbol{\omega}_0 \times \boldsymbol{\omega}) + \mathbf{j}^{-1}\tilde{\mathbf{m}}(\mathbf{v}, \mathbf{v}_0, \boldsymbol{\omega}, \boldsymbol{\omega}_0), && (5.21)\\
\dot{\tau} &= b\tau - b\tilde{u}_\tau, && (5.22)
\end{aligned}$$

where we have used the fact that many terms cancel or vanish at the equilibrium or reference point (in particular we have (5.8)) and also introduced $\tilde{u}_\tau$ as

$$\tilde{u}_\tau = u_\tau - \tau_0.$$

(Note that there is no linearization involved in (5.19)–(5.22), it is still the full nonlinear equations but rewritten omitting terms that sum to zero.) We are going to assume that we can control the three moment vector components m_1, m_3, m_3 directly and that it is the responsibility of the controller to perform the "inversion" from moments to control surface commands. Furthermore, we are going to neglect actuator dynamics and assume that all states in the state vector (5.7) are measurable so that we can employ state feedback control on (5.19)–(5.22).

5.1.2 A Nonlinear Autopilot

The three-dimensional attitude-velocity control problem can be cast as the problem of controlling all three components of the aircraft velocity vector $\mathbf{V}$ in the frame F_e. If this is done with regard only to the relations between the velocity vector components V_1, V_2, V_3 we obtain a natural three-dimensional generalization of the standard flight path angle control problem. However, we are here going to also consider simultaneous control of the magnitude $\|\mathbf{V}\|$ of the velocity vector in F_e. If we recall the relation (5.1) between the velocity in the frames F_e and f_b, respectively, it is clear that the attitude-velocity control problem can (neglecting wind) be split into two separate problems; (*i*) the problem of controlling the body components (i.e. in f_b) of the airspeed vector $\mathbf{v}$ and (*ii*) simultaneously controlling the orientation in terms of $\mathbf{R}$ of the aircraft (in F_e). Since we only have to our disposal as control inputs the three moment vector components m_1, m_2, m_3 and the thrust command τ it is clear that this is an underactuated control problem. [5]

Integrator Block Backstepping

The simplest form of multi-input, or block, backstepping deals with controlled dynamical systems of the generic affine form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t), \tag{5.23}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m$ are Lipschitz continuous and $\mathbf{u}(t)$ is a vector in $\mathbb{R}^m$ of continuous control functions. To make things simple we

[5] Controllability is however, as shown by extensive human experience, not a problem.

are going to assume that both f, g are smooth. The object is then to find a smooth $\ell : \mathbb{R}^n \to \mathbb{R}^m$ such that if $\mathbf{u}$ in (5.23) is given by

$$\mathbf{u} = \ell(\mathbf{x}) \tag{5.24}$$

the resulting system

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\ell(\mathbf{x}(t)) \tag{5.25}$$

is stable, in some suitable sense. In integrator backstepping this is accomplished by first augmenting the system with integrators on the inputs so that the system (5.23) transforms into

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\boldsymbol{\xi}(t), \tag{5.26}$$
$$\dot{\boldsymbol{\xi}}(t) = \mathbf{u}(t). \tag{5.27}$$

The following step can intuitively be described as finding a control function $\mathbf{u}$ to the augmented system (5.26),(5.27) such that when it is applied the state $\boldsymbol{\xi}$ behaves as "if it were" a suitable feedback function ℓ as in (5.24). This desired behavior is called a *virtual control* and closely associated to it is the concept of control Lyapunov function. A smooth positive definite function $\mathcal{V} : \mathbb{R}^n \to [0, \infty)$ is called a *control Lyapunov function* (CLF) for the system (5.23) if it holds that

$$\inf_{\mathbf{u}\in\mathbb{R}^m} \frac{\partial \mathcal{V}}{\partial \mathbf{x}}(\mathbf{x})\big(f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}\big) < 0, \quad \forall \mathbf{x} \neq 0.$$

The basic assumption about the system (5.27) that we need in order to actually apply integrator backstepping is one about stabilizability.

Assumption 5.1.1. Consider the system in (5.23) and assume that there exist a smooth feedback law ℓ as in (5.24) and a smooth, positive definite function $\mathcal{V} : \mathbb{R}^n \to [0, \infty)$ such that

$$\frac{\partial \mathcal{V}}{\partial \mathbf{x}}(\mathbf{x})\big(f(\mathbf{x}) + g(\mathbf{x})\ell(\mathbf{x})\big) \leq -W(\mathbf{x}) \leq 0, \quad \forall \mathbf{x} \in \mathbb{R}^n, \tag{5.28}$$

for some continuous $W : \mathbb{R}^n \to \mathbb{R}$ which is positive definite.

Under this assumption one can formulate the basic result for integrator backstepping as follows.

Theorem 5.1.2. *Consider the system (5.26), (5.27) and suppose that (5.26) satisfies Assumption 5.1.1 with $\boldsymbol{\xi}$ replaced by the control $\mathbf{u}$ in (5.24). If the function W is positive definite, then*

$$\mathcal{V}_a(\mathbf{x}, \boldsymbol{\xi}) = \mathcal{V}(\mathbf{x}) + \frac{1}{2}\|\boldsymbol{\xi} - \ell(\mathbf{x})\|^2 \tag{5.29}$$

is a CLF for the full system (5.26), (5.27) (i.e. $\boldsymbol{\xi}$ plays the role of a control in (5.29)) and there exists a feedback law $\mathbf{u} = \ell_a(\mathbf{x}, \boldsymbol{\xi})$ that makes the full system (5.26), (5.27) asymptotically stable around $\mathbf{x} = 0, \boldsymbol{\xi} = 0$. One such control law is

$$\mathbf{u} = -c(\boldsymbol{\xi} - \ell(\mathbf{x})) + \frac{\partial \ell(\mathbf{x})}{\partial \mathbf{x}}\big(f(\mathbf{x}) + g(\mathbf{x})\boldsymbol{\xi}\big) - \big(\frac{\partial \mathcal{V}}{\partial \mathbf{x}}(\mathbf{x})g(\mathbf{x})\big)^T, \quad c > 0. \tag{5.30}$$

A proof of this theorem can be found in e.g. [21]. In what follows we are going to apply the method of backstepping as outlined in Thm. 5.1.2 to a slightly augmented version of the system (5.26),(5.27) namely

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t)) + g(\mathbf{x}(t))\boldsymbol{\xi}(t), && (5.31)\\ \dot{\boldsymbol{\xi}}(t) &= h(\boldsymbol{\xi}(t)) + k\mathbf{u}(t), && (5.32) \end{aligned}$$

where $\boldsymbol{\xi}$ takes values in $\mathbb{R}^m$, the function $h : \mathbb{R}^m \to \mathbb{R}^m$ is smooth and k is a constant invertible matrix in $\mathbb{R}^{m\times m}$. Since k in (5.32) is nonsingular, and we can choose the control $\mathbf{u}$ freely, the system in (5.31), (5.32) can via a simple change of variables be brought to the form (5.26),(5.27) and the backstepping control problem for the two systems is one and the same.

Standard Form of the Equations of Motion

To be able to apply the theory for integrator block backstepping as it stands the system in question has to be on the standard from (5.26),(5.27) or (5.31),(5.32). However, a glance at (5.19)–(5.22) reveals that this system is already on the required standard form. Indeed, if we make the following associations (here $\sim$ means "corresponds to")

$$\mathbf{x} \sim \begin{bmatrix} \mathbf{v} \\ \mathbf{Q} \end{bmatrix}, \quad \boldsymbol{\xi} \sim \begin{bmatrix} \boldsymbol{\omega} \\ \tau \end{bmatrix}, \quad \mathbf{u} \sim \begin{bmatrix} \tilde{\mathbf{m}}(\mathbf{v}, \mathbf{v}_0, \boldsymbol{\omega}, \boldsymbol{\omega}_0) \\ \tilde{u}_\tau \end{bmatrix} \qquad (5.33)$$

and

$$f \sim \begin{bmatrix} \frac{1}{m}\tilde{\mathbf{f}}(\mathbf{v}, \mathbf{v}_0) + \tilde{\mathbf{g}}(\mathbf{Q}, \mathbf{Q}_0) + \mathbf{v} \times \boldsymbol{\omega}_0 \\ \frac{1}{2}\mathbf{B}(\mathbf{Q})\boldsymbol{\omega}_0 \end{bmatrix}, \qquad (5.34)$$

$$g \sim \begin{bmatrix} \mathbf{C}(\mathbf{v} + \mathbf{v}_0) & \frac{1}{m}\mathbf{e}_x \\ \frac{1}{2}\mathbf{B}(\mathbf{Q} + \mathbf{Q}_0) & \mathbf{0}_{4\times 1} \end{bmatrix}, \quad h \sim \begin{bmatrix} \mathbf{j}^{-1}(\mathbf{j}\boldsymbol{\omega} \times (\boldsymbol{\omega} + \boldsymbol{\omega}_0) + \mathbf{j}\boldsymbol{\omega}_0 \times \boldsymbol{\omega}) \\ b\tau \end{bmatrix}, \qquad (5.35)$$

$$k \sim \begin{bmatrix} \mathbf{j}^{-1} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & -b \end{bmatrix} \qquad (5.36)$$

we see that (5.19)–(5.22) is on standard form for integrator block backstepping.

The Proposed Controller

The first task when developing a backstepping controller for the system on standard form is to find a suitable Lyapunov function $\mathcal{V}$ as in Thm. 5.1.2 and an accompanying feedback law ℓ such that the feedback connected first part of the system, as in (5.25), is stable with suitable dynamics. When determining what is "suitable" dynamics for the system (5.19), (5.20) we must take into account at least two obvious requirements; (i) the need to aerodynamically stabilize the aircraft and (ii) the desire to solve the attitude and velocity control problem outlined in the beginning of Sec. 5.1.2. It is intuitively clear that these two requirements can not be dealt with independently since rotating the aircraft body so that the body velocity error vector $\mathbf{v}$ becomes zero does not necessarily mean that the aircraft has desired orientation $\mathbf{Q}_0$. We are going to solve this problem by combining two controllers, one for the velocity error and one for the orientation error.

Controlling the velocity.

The problem of aerodynamically stabilizing the aircraft, without regard to its orientation, is not too hard once the system has been brought onto the form (5.19)–(5.22). For instance, one can control $\mathbf{m}(\mathbf{v}, \mathbf{v}_0, \boldsymbol{\omega}, \boldsymbol{\omega}_0)$ such that the velocity vector $\mathbf{v} + \mathbf{v}_0$ is rotated into a position aligned with $\mathbf{v}_0$, while simultaneously controlling the thrust setting $\tilde{u}_\tau$ so that the magnitude becomes right.

A simple way of achieving a rotation of the velocity vector $\mathbf{v} + \mathbf{v}_0$ in the right direction is to use a (virtual) control of the form

$$\boldsymbol{\omega}_{\mathbf{v}}^{des}(\mathbf{v}, \mathbf{v}_0) = -\frac{c_{\mathbf{v}}}{\|\mathbf{v}_0\|^2}\mathbf{v} \times \mathbf{v}_0, \tag{5.37}$$

where $c_{\mathbf{v}}$ is some positive constant. To give some motivation at this point for the choice (5.37) of a (virtual) control $\boldsymbol{\omega}_{\mathbf{v}}^{des}(\mathbf{v}, \mathbf{v}_0)$ one can note that

$$\begin{aligned}
-(\mathbf{v} + \mathbf{v}_0) \times \frac{1}{\|\mathbf{v}_0\|^2}(\mathbf{v} \times \mathbf{v}_0) = \\
-\mathbf{v} \times \frac{1}{\|\mathbf{v}_0\|^2}(\mathbf{v} \times \mathbf{v}_0) - \mathbf{v}_0 \times \frac{1}{\|\mathbf{v}_0\|^2}(\mathbf{v} \times \mathbf{v}_0) \\
= -\mathbf{v} \times \frac{1}{\|\mathbf{v}_0\|^2}(\mathbf{v} \times \mathbf{v}_0) - P_{[\mathbf{v}_0]^\perp}(\mathbf{v}),
\end{aligned} \tag{5.38}$$

where the first term on the right is perpendicular to $\mathbf{v}$ and $P_{[\mathbf{v}_0]^\perp}(\mathbf{v})$ is the projection of $\mathbf{v}$ onto the subspace of vectors in $\mathbb{R}^3$ that are orthogonal to $\mathbf{v}_0$. The first term is in general much smaller in magnitude than the second and therefore, when inserted instead of $\boldsymbol{\omega}$ in (5.19), the (virtual) control $\boldsymbol{\omega}_{\mathbf{v}}^{des}(\mathbf{v}, \mathbf{v}_0)$ in (5.37) can act to reduce the error $\mathbf{v}$. However, since the main reduction of the velocity error $\mathbf{v}$ is in the component of it that is orthogonal to $\mathbf{v}_0$ there is a need to complement the control with some action also in the direction of $\mathbf{v}_0$.

The direction of $\mathbf{v}_0$ is normally almost the same as the direction in which the thrust acts (here, in the body x-direction) and therefore it is natural to try to achieve control action in the $\mathbf{v}_0$ direction by (virtual) thrust control. A simple way to achieve this is to employ a negative velocity feedback along the direction of $\mathbf{v}_0$, for example using the (virtual) thrust control $\tau^{des}(\mathbf{v}, \mathbf{v}_0, \mathbf{Q}, \mathbf{Q}_0)$ as

$$\tau^{des}(\mathbf{v}, \mathbf{v}_0, \mathbf{Q}, \mathbf{Q}_0) = -\frac{c_\tau m v_1}{c_\ell + v_1^2}\Big(\frac{\mathbf{v}^T\mathbf{v}_0}{\|\mathbf{v}_0\|}\Big)^2 = -\frac{c_\tau m v_1}{c_\ell + v_1^2}\big\|P_{[\mathbf{v}_0]}(\mathbf{v})\big\|^2, \tag{5.39}$$

where c_τ, c_ℓ are positive constants and $P_{[\mathbf{v}_0]}(\mathbf{v})$ is the projection of $\mathbf{v}$ onto the one-dimensional subspace spanned by $\mathbf{v}_0$. This type of virtual thrust control would, if $\mathbf{v}$ is large and mostly aligned with $\mathbf{v}_0$, approximately give a stable linear first order contribution to the dynamics in the x-axis component of the force equation (5.19). (For small $\mathbf{v}$ this control would under the same conditions do essentially nothing.) Therefore, when c_τ is close to $c_{\mathbf{v}}$ it is clear from (5.37) and (5.38) that the combined effect of the virtual controls $\boldsymbol{\omega}_{\mathbf{v}}^{des}(\mathbf{v}, \mathbf{v}_0)$ and $\tau^{des}(\mathbf{v}, \mathbf{v}_0, \mathbf{Q}, \mathbf{Q}_0)$ for large $\mathbf{v}$ is to give the overall system roughly first order stable (virtual) dynamics for the error $\mathbf{v}$.

Controlling the attitude.

We now turn to the problem of controlling the attitude of the aircraft. When conceiving a solution to this problem we will, in analogy with the approach above, neglect the other part of the control problem, viz. the problem of aerodynamically stabilizing the aircraft. The attitude control can be archived by rotating the body along the shortest path from the current orientation to the desired orientation on the set of unit norm quaternions, which we here identify with (one "half" of) $\mathcal{S}^3$, the unit sphere in ordinary four dimensional space. Such a shortest path is the same as a *geodesic* (in the ordinary Euclidean metric on the tangent space of $\mathcal{S}^3$). A simple parameterization for this type of geodesic called *slerp* (for Spherical Linear Interpolation) was introduced in 1985 by Shoemake [36].

The slerp $\mathcal{Q}$ that describes the path from unit norm quaternion $\mathcal{Q}_1$ to unit norm quaternion $\mathcal{Q}_0$ is given by

$$\mathcal{Q}(t) = \frac{\sin((1-t)\theta)\mathcal{Q}_0}{\sin(\theta)} + \frac{\sin(t\theta)\mathcal{Q}_1}{\sin(\theta)}, \quad t \in [0,1], \tag{5.40}$$

where θ is defined by

$$\cos(\theta) = \mathcal{Q}_0^T \mathcal{Q}_1$$

and the inner product on the right hand side is calculated as for ordinary vectors in four-space. The time-derivative of the slerp $\mathcal{Q}$ is easily calculated as

$$\dot{\mathcal{Q}}(t) = \frac{\theta}{\sin(\theta)}(\cos(t\theta)\mathcal{Q}_1 - \cos((1-t)\theta)\mathcal{Q}_0), \quad t \in [0,1], \tag{5.41}$$

and this shows that the motion along the slerp path takes place at constant speed i.e. $\|\dot{\mathcal{Q}}(t)\| \equiv const.$

In our application, where we want to design a feedback law based on the slerp formula (5.40), the start quaternion $\mathcal{Q}_1$ will be constantly changing and so we really only use the expression (5.41) for the slerp velocity vector, and evaluate it at the (changing) starting point. Indeed, at least in the case that $\mathbf{Q}_0$ is constant (i.e. $\boldsymbol{\omega}_0 = 0$) it is clear that what we want to achieve with the (virtual) attitude control is

$$\dot{\mathbf{Q}} = c_{\mathbf{Q}} \dot{\mathcal{Q}}(0), \tag{5.42}$$

where $\mathbf{Q}$ is the state quaternion in (5.20) and $\mathbf{Q} + \mathbf{Q}_0, \mathbf{Q}_0$ in (5.20) are used instead of $\mathcal{Q}_0, \mathcal{Q}_1$ in (5.41), for some positive constant $c_{\mathbf{Q}}$. This gives a condition for the sought virtual angular velocity $\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0)$ for attitude control as

$$\frac{1}{2}\mathbf{B}(\mathbf{Q}+\mathbf{Q}_0)\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0) = c_{\mathbf{Q}} \frac{\theta}{\sin(\theta)}\big(\mathbf{Q}_0 - \cos(\theta)(\mathbf{Q}+\mathbf{Q}_0)\big), \tag{5.43}$$

where now θ is given by

$$\cos(\theta) = (\mathbf{Q}+\mathbf{Q}_0)^T \mathbf{Q}_0. \tag{5.44}$$

From (5.43) it might appear impossible to solve (uniquely) for $\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0)$ in (5.43) since the matrix $\mathbf{B}$ is not square, but if we remember that the left hand side of (5.43) is really just another way of writing the product of two quaternions, one unit norm and one pure, we can determine $\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0)$ explicitly. Working through the algebra we get

$$\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0) = c_{\mathbf{Q}} \frac{2\theta}{\sin(\theta)} \Im(\mathbf{Q}^c \circ \mathbf{Q}_0) \tag{5.45}$$

(with θ as in (5.44)) where $\Im(\,\cdot\,)$ denotes quaternion imaginary part and $(\,\cdot\,)^c$ denotes quaternion conjugation. In case $\boldsymbol{\omega}_0$ is constant but nonzero it is clear, after a moments contemplation, that the same principle for selecting $\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0)$ ought to apply, and that the resulting dynamics in this "moving" scenario on $\mathcal{S}^3$ then becomes the same as in (5.20), if we replace $\boldsymbol{\omega}$ there by $\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0)$. We then have

$$\frac{d}{dt}(\mathbf{Q}+\mathbf{Q}_0) = \dot{\mathbf{Q}} + \dot{\mathbf{Q}}_0 = \frac{1}{2}\mathbf{B}(\mathbf{Q}+\mathbf{Q}_0)(\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0) + \boldsymbol{\omega}_0), \tag{5.46}$$

with $\dot{\mathbf{Q}}$ as in (5.42) and $\boldsymbol{\omega}_{\mathbf{Q}}^{des}(\mathbf{Q}, \mathbf{Q}_0)$ as in (5.45).

Total controller.

The complete (virtual) controller corresponding to ℓ in Thm. 5.1.2 is now given by the vector

$$\begin{pmatrix} \boldsymbol{\omega}^{des}(\mathbf{v},\mathbf{v}_0,\mathbf{Q},\mathbf{Q}_0) \\ \tau^{des}(\mathbf{v},\mathbf{v}_0,\mathbf{Q},\mathbf{Q}_0) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\omega}^{des}_{\mathbf{v}}(\mathbf{v},\mathbf{v}_0) + \boldsymbol{\omega}^{des}_{\mathbf{Q}}(\mathbf{Q},\mathbf{Q}_0) \\ \tau^{des}(\mathbf{v},\mathbf{v}_0,\mathbf{Q},\mathbf{Q}_0), \end{pmatrix}, \tag{5.47}$$

with the components on the right hand side given by (5.37),(5.45) and (5.39). As a candidate for the "inner" Lyapunov function $\mathcal{V}$ as in Thm. 5.1.2 we shall take

$$\mathcal{V}(\mathbf{v},\mathbf{Q}) = \frac{\gamma_{\mathbf{v}}}{2}\|\mathbf{v}\|^2 + \frac{\gamma_{\mathbf{Q}}}{2}\|\mathbf{Q}\|^2, \tag{5.48}$$

where $\gamma_{\mathbf{v}}, \gamma_{\mathbf{Q}}$ are two positive constants [6] (to be determined later) and the norms are ordinary 2-norms in $\mathbb{R}^3$ and $\mathbb{R}^4$, respectively.

A proof of stability for this type of controller is given in [32] for suitable choices of the constants $\gamma_{\mathbf{v}}$,$\gamma_{\mathbf{Q}}$, $c_{\mathbf{v}}$ and $c_{\mathbf{Q}}$, using only weak assumptions [7] about the aerodynamic configuration of the aircraft. It should be pointed out, however, that many other controller solutions are possible. Once the model has been put on the standard form as in (5.19)–(5.22) there are many ways of constructing stabilizing controllers using backstepping.

5.1.3 Simulations

The ADMIRE model is a full nonlinear six-degree-of-freedom model of a small single engine fighter aircraft with delta canard configuration (similar to the JAS39 Gripen). It is implemented in Simulink and is freely available on the Internet at http://www.foi.se/admire. It has been used as classroom model at universities and as benchmark model in research collaboration projects between industry and universities. A simplified version of the model has also been implemented in Modelica, and this version is the one used in the present work. A block diagram overview of the Modelica version of the model with controller is shown in Figure 5.1. The simulations have been carried out using the Dymola integrated development environment on a personal computer running GNU/Linux. A flight trajectory lasting for 200 seconds was programmed which included maneuvering as well as straight path flight. The various constants (e.g. the "feedback gains") occurring in the controller description were chosen to give a reasonable fast system with good performance. [8]

The chosen flight trajectory is shown in Figure 5.2 and starts at wings level flight at an altitude of 3000 meters. The following right and left turns require bank angles of about 30°. As the left turn is completed, the aircraft begins an 11°-12° climb. During this climb, the Mach number is gradually increased from 0.6 to 0.7 shown in Figure 5.3(a). The trajectory ends at an altitude of about 8000 meters. It should be noted that the programmed trajectory contains short segments where the reference values in (5.6) for $[\mathbf{v}_0, \mathbf{Q}_0, \boldsymbol{\omega}_0, \tau_0]^T$ are obtained by linear interpolation and therefore do not satisfy the requirements imposed by the theory.

---

[6] The reason we introduce two independent weighting constants here is that we want to be able to control the relative magnitude of all three terms of the resulting total control law in (5.30).

[7] A local force stability around a trimmed flight condition is assumed [32]. This is often satisfied for all but very low airspeeds.

[8] The feedback gains are outside the somewhat conservative region of values given by the sufficient condition for stability in [32]. The main difference between the performance here and in [32], where the constants are in the guaranteed stability region, is a slower velocity error convergence in [32].

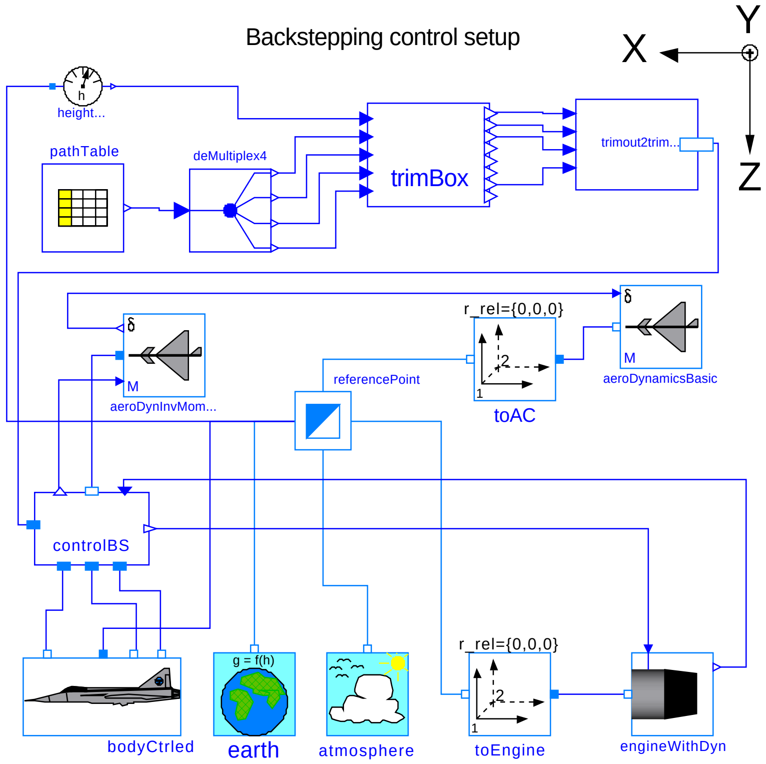


Figure 5.1: Block diagram representation of the Modelica version of the ADMIRE model with trimming routine and controller.

The velocity tracking yields quite small velocity deviations from the trimmed values, as seen in Figure 5.3(b). There seem to be a steady state velocity error in the last part of the trajectory. The reason is that we require a constant Mach number during the climb, but the controller tracks the velocity. Since the velocity of sound decreases with altitude, the velocity of the aircraft will be decreased as well. Hence the velocity deviation.

Figure 5.4(a) shows the actual angle of attack as well as the trimmed angle of attack at each time instant. The discrepancies in these values seen between 75 seconds and 120 seconds can be explained by the pull-up maneuver for the climb and by the increase in velocity. The increasing angle of attack after 120 seconds is due to the climb since the air gets thinner. The small mismatch during the climb originates from the velocity deviation discussed above. As desired, the sideslip angle is small during the whole simulation, see Figure 5.4(b).

The orientation of the aircraft stays very close to the trimmed orientation, shown in Figure 5.5. The somewhat larger discrepancy in these values occurring between 75 and 100 seconds comes from the maneuver when the aircraft finishes the turn, by banking to wings level flight, and simultaneously pulls up for the climb. In the last half of the simulation, the slowly increasing angle of attack can be noted in the aircraft orientation.

Finally, in Figure 5.6, the value of the time derivative of the Lyapunov function $\mathcal{V}$ in (5.48) along the solutions to the feedback connected system (corresponding to (5.25)) is displayed. It can be seen from Fig. 5.6(a) that the overall behavior of the time derivative is as expected with the most negative values occurring shortly after the start when the system moves towards an equilibrium corresponding to straight path flight. The derivative becomes (slightly) positive on a few occasions when the set point values change linearly

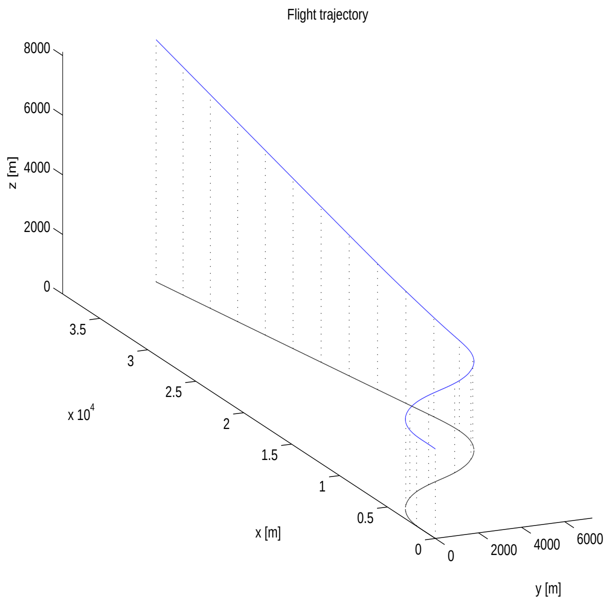
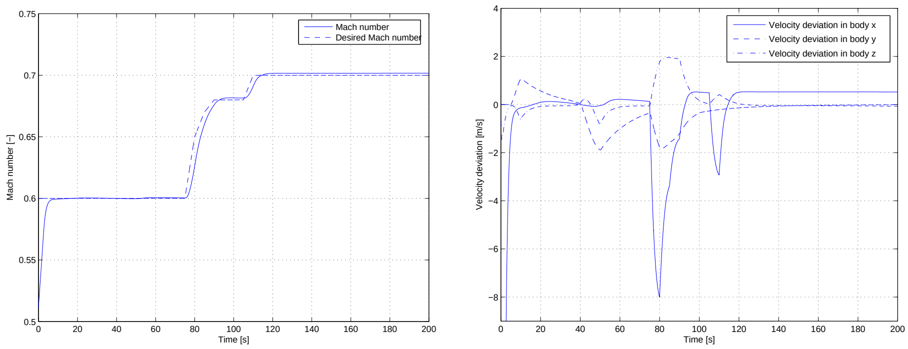


Figure 5.2: Flight trajectory.



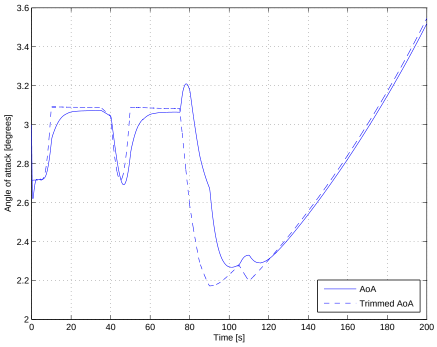
(a) Mach number

(b) Velocity deviation

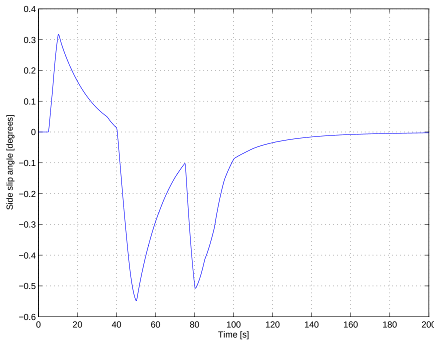
Figure 5.3: Mach number and velocity deviation.

between trimmed values by interpolation, a situation which is not covered by the stability result in [32]. [9]

[9] These periods of time moreover correspond to the most significant changes in reference angular velocity $\boldsymbol{\omega}_0$, as can be inferred from e.g. the trimmed roll angle, cf. Fig. 5.5(b). The theory in [32] guarantees negativity of the Lyapunov function time derivative in the periods between the changes, if the changes are step-wise changes between straight path flight segments, provided the various constants used in the controller are within the region prescribed by the stability result in [32].

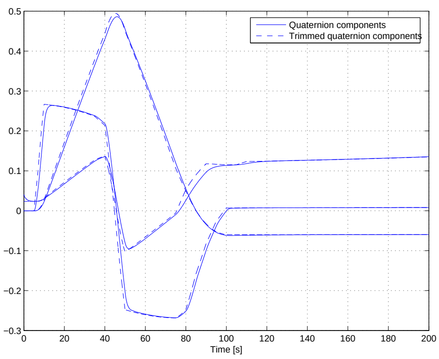


(a) Angle of attack

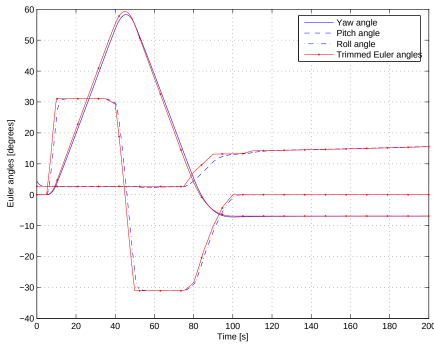


(b) Sideslip angle

Figure 5.4: Angle of attack and sideslip angle.

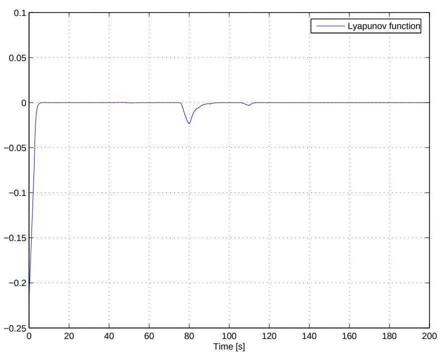


(a) Quaternion components

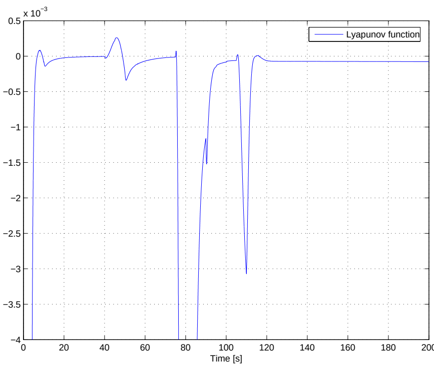


(b) Euler angles

Figure 5.5: Components of the quaternion and corresponding Euler angles.



(a) Lyapunov function rate of change



(b) Detail of (a)

Figure 5.6: Lyapunov function rate of change.

Bibliography

[1] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández. A Lanczos-type method for multiple starting vectors. *Mathematics of Computation*, 69:1577–1601, 2000.

[2] A. Astolfi and A. Rapaport. Robust stabilization of the angular velocity of a rigid body. *Systems & Control Letters*, 1998.

[3] Tamer Başar and Geert Jan Olsder. Dynamic Noncooperative Game Theory. SIAM, 1998.

[4] Zhaojun Bai and Daniel Skoogh. Krylov subspace techniques for reduced-order modeling of nonlinear dynamical systems. In *Electronic Proceedings of 15th International Symposium of the Mathematical Theory of Networks and Systems*, University of Notre Dame, South Bend, Indiana, USA, 2002.

[5] Zhaojun Bai and Daniel Skoogh. Model reduction of weakly nonlinear systems using modified carleman bilinearization and projection formulation. In *Electronic Proceedings of 16th International Symposium of the Mathematical Theory of Networks and Systems*, Katholieke Universiteit Leuven, Belgium, 2004. Link error on CD, correct version obtained in the online proceedings at `www.mtns2004.be`.

[6] Zhaojun Bai and Daniel Skoogh. A projection method for model reduction of bilinear dynamical systems. *To appear in: Linear Algebra and its Application*, 2005.

[7] Zhaojun Bai and Daniel Skoogh. A two-sided projection method for model reduction of bilinear dynamical systems. *Work in progress*, 2005.

[8] Y. Chen and J. White. A quadratic method for nonlinear model order reduction. In *Inter. Conf. on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators*, pages 477–480, San Diego, 2000.

[9] J. Dahlgren. Robust nonlinear control design for a missile using backstepping. Technical report, Linköping Institute of Technology, 2002.

[10] K. Pettersen F. Mazenc and H. Nijmeijer. Global uniform asymptotic stabilization of an underactuated surface vessel. *IEEE Trans. Autom. Control*, 47(10):1759–1762, October 2002.

[11] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14:639–649, 1995.

[12] R. W. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. In *Applied and Computational Control, Signals, and Circuits*, chapter 9. Birkhauser, 1999.

[13] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM J. Sci. Comp.*, 14:137–158, 1993.

[14] R. W. Freund and H. Zha. A look-ahead algorithm for the solution of general Hankel systems. *Numerische Mathematik*, 64:295–321, 1993.

[15] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1.* IEEE Press, Piscataway NJ, 2004.

[16] K. Gallivan, E. J. Grimme, and P. Van Dooren. A rational Lanczos algorithm for model reduction. *Numerical Algorithms*, 12:33–63, 1996.

[17] Torkel Glad and Ola Härkegård. Backstepping control of a rigid body. In *Proc. CDC '02*, Las Vegas NV, 2002.

[18] E. J. Grimme. *Krylov Projection Methods For Model Reduction.* PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[19] Johan Hamberg and Fredrik Svahn. Flight path control of a missile using the passive normal form. *Accepted for AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[20] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge Univ. Press, 1996.

[21] H.K. Khalil. *Nonlinear Systems.* Prentice Hall, NJ, 3 edition, 2002.

[22] K-S. Kim and Y.K. Kim. Robust backstepping control for slew maneuver using nonlinear tracking function. *IEEE Trans. Control Syst. Technol.*, 11(6):822–829, 2003.

[23] Miroslav Krstić, Ioannis Kanellakopoulos, and Petar Kokotović. *Nonlinear & Adaptive Control Design.* Wiley Interscience, 1995.

[24] Peng Li and Lawrence T. Pileggi. NORM: Compact model order reduction of weakly nonlinear systems. In *DEC*, 2003.

[25] Ronald R. Mohler. *Nonlinear systems, volume II applications to bilinear control.* Prentice Hall, 1991.

[26] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. In *Technical Digest of the 1997 IEEE/ACM International Conference on Computer-Aided Design*, pages 58–65. IEEE Computer Society Press, Los Alamitos, California, 1997.

[27] Joel R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Proceedings Design Automation Conference*, pages 184–189, 2000.

[28] Joel R. Phillips. Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 22(2), February 2003.

[29] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Transactioins on Computer-Aided Design*, 9:352–366, 1990.

[30] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. to appear.

[31] M. Rewieński and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuit and micromachined devices. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 22:155–170, 2003.

[32] J.W.C. Robinson and U. Nilsson. Design of a nonlinear autopilot for velocity and attitude control using block backstepping. In *Proc. AIAA Guidance, Navigation & Control Conf. & Exhibit '05*, San Francisco CA, 2005. AIAA paper 2005-6266.

[33] Wilson J. Rugh. *Nonlinear System Theory, The Volterra/Wiener Approach.* The Johns Hopkins University Press, Baltimore and London, 1981.

[34] J.M.A. Scherpen. Balancing for nonlinear systems. *Systems and Control Letters*, 21:143–153, 1993.

[35] Rodolphe Sepulchre and Mrdjan Janković Petar Kokotović. *Constructive Nonlinear Control.* Springer, 1997.

[36] K. Shoemake. Animating rotation with quaternion curves. In *Proc. SIGGRAPH '85*, pages 245 – 254, San Francisco CA, 1985. ACM.

[37] Daniel Skoogh. *Krylov Subspace Methods for Linear Systems, Eigenvalues and Model Order Reduction.* PhD thesis, Department of Mathematics, Chalmers University of Technology, Sweden, December 1998.

[38] B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation.* Wiley, Hoboken NJ, 2003.

[39] G. W. Stewart. *Matrix Algorithms, Vol I: Basic decompositions.* SIAM, 1998.

[40] Fredrik Svahn. Backstepping and its application to a missile. Technical report, FOI-R–1440, 2004.

[41] D. R. Taylor. *Analysis of the Look Ahead Lanczos Algorithm.* PhD thesis, University of California at Berkeley, 1982.

[42] Michael A. Tiller. *Introduction to Physical Modeling with Modelica.* Kluwer, Boston MA, 2001.

[43] Kemin Zhou and John C. Doyle. *Essentials of Robust Control.* Prentice Hall, 1998.

FOI
Swedish Defence Research Agency
SE-164 90 STOCKHOLM

Tel: +46 8 5550 3000
Fax: +46 8 5550 3100

www.foi.se