On Modeling and Nonlinear Model Reduction in Automotive Systems

Nilsson, Oskar

2009

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Nilsson, O. (2009). *On Modeling and Nonlinear Model Reduction in Automotive Systems*. [Doctoral Thesis (monograph), Department of Automatic Control]. Department of Automatic Control, Lund Institute of Technology, Lund University.

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

On Modeling and Nonlinear Model Reduction in Automotive Systems

On Modeling and Nonlinear Model Reduction in Automotive Systems

Oskar Nilsson

Department of Automatic Control
Lund University
Lund, March 2009

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

ISSN 0280–5316
ISRN LUTFD2/TFRT--1085--SE

Printed in Sweden.
Lund 2009

Abstract

The current control design development process in automotive industry and elsewhere involves many expensive experiments and hand-tuning of control parameters. Model based control design is a promising approach to reduce costs and development time. In this process low complexity models are essential and model reduction methods are very useful tools.

This thesis combines the areas of modeling and model reduction with applications in automotive systems. A model reduction case study is performed on an engine air path. The heuristic method commonly used when modeling engine dynamics is compared with a more systematic approach based on the balanced truncation method.

The main contribution of this thesis is a method for model reduction of nonlinear systems. The procedure is focused on reducing the number of states using information obtained by linearization around trajectories. The methodology is closely tied to existing theory on error bounds and good results are shown in form of examples such as a controller used in real-world cars.

Also, a model of the exhaust gas oxygen sensor, used for air-fuel ratio control in automotive spark-ignition engines, is developed and successfully validated.

Acknowledgments

First of all, I would like to thank my supervisor Anders Rantzer for his support, encouragement and fruitful discussions throughout my work.

My thanks goes to Klaus Papadakis and Ingemar Odenbrand at the Department of Chemical Engineering, Lund University, for their helpful suggestions concerning the Lambda sensor chemistry. Also, Per Tunestål, Jan-Erik Everitt with others deserve acknowledgments for their assistance in the Combustion Engines Lab.

I owe gratitude to Jonathan Chauvin at École Nationale Supérieure des Mines de Paris, for contributing with the heuristic model reduction part in Chapter 2.

Our industrial contact Akira Ohata from Toyota Motor Corporation is acknowledged for providing industrial problem settings and for enabling a six-month visit at their facilities in 2007. More people than can be listed here made this visit fruitful in many aspects. Ito-san, Okuda-san, Katayama-san, Kaneko-san, Katsumata-san, Nakada-san and Kato-san all made this an unforgettable experience together with the rest of the 42Y group. Without your support I would have been helplessly lost in Japan.

Pedro García from Universidad Politécnica de Valencia is acknowledged for his interesting viewpoints and excellent management of my five week visit in Valencia.

I am also grateful for the proofreading performed by Anders Rantzer, Kin Cheong Sou, Rolf Johansson, Henrik Sandberg, Martin Ohlin and Jonas Sjögren. Others who I would like to thank for their technical and, in particular, moral support are Brad Schofield, Mathias Persson, Martin Kjær and Martin Ohlin.

Leif Andersson and Anders Blomdell have greatly facilitated all contact with computers. Leif is also acknowledged for typesetting support of this thesis. The secretaries Agneta Tuszynski, Britt-Marie Mårtensson, Eva Schildt and Eva Westin have solved all my administrative problems and they are important cogs in the social machinery of the department.

Financial support was received from Toyota Motor Corporation and this research was partially done in the framework of the HYCON Network of Excellence, contract number FP6-IST-511368.

Finally I would like to thank my friends and family for your encouragement and support.

Oskar

Contents

Preface

Motivation

Automotive industry development

The current control design development process in automotive industry involves many expensive experiments and hand-tuning by experienced personnel. This process is time-consuming and even if only small changes have been done between two models, many tuning tasks have to be made over and over again.

Model-based development is a promising approach to reduce costs, development time and dependency of the undocumented knowledge possessed by experienced personnel. The idea is to replace expensive experiments with simulation of mathematical models.

Complexity versus fidelity

The modeling process is highly dependent of the model purpose. Depending on the model usage different effects in the physical plant should either be taken into consideration or be neglected. However, what effects to be included can be very hard to know and requires experience and understanding of the real process.

A model is always approximate and the level of accuracy is typically a function of its complexity. A very detailed model is more likely to cover the most important dynamics but large complexity has many downsides. In general, simulation time and memory requirements scale badly with complexity, also model analysis is made harder. Yet another inconvenience is that larger models generally contain more model parameters. In control design, this could yield that the hand-tuning saved by model-based control design is replaced by time consuming calibration of model parameters.

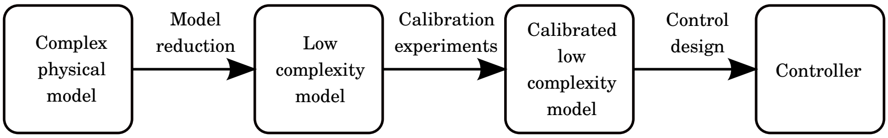

Figure 0.1 Example of alternative controller development process

Why model reduction?

A systematic method to reduce model complexity would be very useful in many situations. If a detailed component model has been developed, the modeling effort could later be reused for other purposes. For example, component models could be combined to model a more overall behaviour. The model complexity could then be reduced with a model reduction method to match the required level of detail for the actual purpose.

In model-based control design, simple models are highly preferred. Some methods, e.g. Linear Quadratic Gaussian control or Model Predictive Control, yield controllers with complexity comparable to the model.

A control design approach where model reduction plays a central role is illustrated in Figure 0.1. A complex physical model with a large number of uncertain parameters could be reduced by a model reduction method. Ideally, the resulting model should not only be of low complexity but should also contain few parameters, facilitating calibration. The small model is then calibrated with experiment data and used for model-based control design.

In some cases fast simulation models for real-time purposes are essential, e.g. in on-board fault diagnostics where computing power is not abundant.

Outline and contributions

The thesis combines the areas of modeling and model reduction of automotive systems. Here is an outline together with related publications.

Chapter 1: Background

The first chapter gives an introduction to model reduction methods of linear and nonlinear systems.

Chapter 2: A model reduction case study

Two model reduction methodologies are applied on a detailed engine air path model. One of the methodologies is systematic and mathematically motivated, while the other is heuristic and based on intuition and experience.

Related publications

Nilsson, O., A. Rantzer, and J. Chauvin (2006): "A model reduction case study: Automotive engine air path." In *Proceedings of the IEEE International Conference on Control Applications*. Munich, Germany.

Nilsson, O. (2006): "Modeling and model reduction in automotive systems." Licentiate Thesis ISRN LUTFD2/TFRT--3242--SE. Department of Automatic Control, Lund University, Sweden.

Chapter 3: The average Gramian approach to model reduction

This chapter presents a general model reduction method for nonlinear systems. The method is numerically attractive and is developed both for the continuous and discrete-time case. Its applicability is demonstrated through numerical examples such as a controller used in real-world cars.

Related publications

Nilsson, O. and A. Rantzer (2009a): "The average Gramian approach to nonlinear model reduction." *IEEE Transactions on Control Systems Technology*. Preprint, submitted.

Nilsson, O. (2006): "Modeling and model reduction in automotive systems." Licentiate Thesis ISRN LUTFD2/TFRT--3242--SE. Department of Automatic Control, Lund University, Sweden.

Nilsson, O. and A. Rantzer (2009a): "A novel approach to balanced truncation of nonlinear systems." In *European Control Conference*. Preprint, submitted.

Nilsson, O. and A. Rantzer (2009b): "A novel nonlinear model reduction method applied to automotive controller software." In *American Control Conference*. Preprint, accepted.

Chapter 4: Modeling the exhaust gas oxygen sensor

A model of an exhaust gas oxygen sensor, also called Lambda sensor, is derived. This sensor is a core component in the emission control in modern spark ignition combustion engines. The model is calibrated using measurement data.

Related publications

Nilsson, O. (2006): "Modeling and model reduction in automotive systems." Licentiate Thesis ISRN LUTFD2/TFRT--3242--SE. Department of Automatic Control, Lund University, Sweden.

Chapter 5: Conclusions

Finally, this chapter contains concluding remarks together with possible directions of further research.

1

Background

This chapter presents an overview of model reduction methods relevant for this thesis. For further reading, a broad overview of model reduction methods is presented in [Antoulas and Sorensen, 2001] and [Obinata and Anderson, 2001].

1.1 Model reduction of linear systems

A linear time-invariant system can be represented in many different ways. A common description is the state-space form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.1}$$

where $x(t) \in \mathbf{R}^n$, $u(t) \in \mathbf{R}^l$ and $y(t) \in \mathbf{R}^m$. If the model order n is much larger than the number of inputs and outputs($n \gg l$, $n \gg m$) it can be suspected that the model contains redundant states. The model reduction problem is how to find and remove such redundancy.

Gramians

The notion of Gramians is a central concept in many model reduction methods. They give a measure of how strongly states are connected to the input and output signals.

The controllability function, as defined in [Scherpen, 1993], is the minimum amount of input energy required to drive the system from the zero state to x_0.

$$L_c(x_0) = \min_{\substack{u \in L_2(-\infty,0) \\ x(-\infty)=0 \\ x(0)=x_0}} \frac{1}{2} \int_{-\infty}^{0} ||u(t)||^2 dt \tag{1.2}$$

Further, the observability function is the amount of energy the initial state x_0 generates in the output signal while the input signal is zero.

$$L_o(x_0) = \frac{1}{2}\int_0^\infty ||y(t)||^2 dt, \quad x(0) = x_0, \quad u \equiv 0 \tag{1.3}$$

The usefulness of these functions for model reduction is clear. If a large amount of energy is required to reach a certain state and if the same state yields a small output energy, this state is unimportant for the input-output behaviour of the system. For linear systems, as defined in (1.1), these functions become the quadratic expressions

$$L_c(x_0) = \frac{1}{2}x_0^T P^{-1} x_0 \qquad\qquad L_o(x_0) = \frac{1}{2}x_0^T Q x_0$$

where P and Q are called the controllability Gramian resp. the observability Gramian. It can be shown, see [Moore, 1981], that these Gramians are

$$P = \int_0^\infty e^{At}BB^Te^{A^Tt}dt \qquad\qquad Q = \int_0^\infty e^{A^Tt}C^TCe^{At}dt$$

Usually stability is assumed and these integrals are well-defined. A more numerically feasible way to compute the Gramians is to determine the unique solutions to the Lyapunov equations

$$\begin{aligned} AP + PA^T + BB^T &= 0 \\ A^TQ + QA + C^TC &= 0 \end{aligned} \tag{1.4}$$

Moreover, the column vectors of P span the controllable subspace in $\mathbf{R}^n$ and correspondingly the null space of Q is the unobservable subspace. These Gramians P and Q, and their analogues for other system classes, are central to many model reduction methods. They show how strongly states are connected to the inputs and outputs and thereby supply essential information of which state subspace is of most significance.

Balanced truncation

Balanced truncation is a popular model reduction technique introduced in [Moore, 1981]. The method guarantees preserved stability and comes with an a priori error bound.

The idea of the method is to apply a coordinate change so that each state is equally controllable and observable. The model is then reduced by truncating states with relatively weak input-output dependency. Applying a linear coordinate change, $T\tilde{x} = x$, to the state-space form given in (1.1) yields the system

$$\begin{aligned}\dot{\tilde{x}}(t) &= TAT^{-1}\tilde{x}(t) + TBu(t)\\ y(t) &= CT^{-1}\tilde{x}(t) + Du(t)\end{aligned}$$

The previously mentioned Gramians determine how controllable and observable these new states are. From (1.4) it can be derived that the new Gramians become $\tilde{P} = TPT^T$ and $\tilde{Q} = T^{-T}QT^{-1}$. A balanced realization is achieved if the coordinate change makes the Gramians diagonal and equal.

$$\tilde{P} = \tilde{Q} = \tilde{\Sigma} = \begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_n\end{bmatrix} \tag{1.5}$$

Methods for computing this coordinate change T can be found in [Zhou and Doyle, 1998; Li, 2000]. The diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ are the Hankel singular values that indicate how important a state is for the input-output relationship. Consequently, the reduced model is derived by truncating states in the balanced realization corresponding to small singular values. Now, letting $\sigma_1^* > \sigma_2^* > \ldots > \sigma_r^*$ denote the *distinct* singular values of the truncated dimensions, it can be shown that the approximation error is bounded as

$$\max_u \frac{\|\tilde{y}(t) - y(t)\|_2}{\|u(t)\|_2} \leq 2\sum_{k=1}^{r} \sigma_k^* \tag{1.6}$$

Balanced truncation of linear discrete-time systems

In the discrete-time case a common description is the state-space form

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k\\ y_k &= Cx_k + Du_k\end{aligned}$$

where x_k is the state vector, u_k the input signal and y_k the output signal at time k. Further, A, B, C and D are matrices of appropriate dimensions. Here the sub-index denotes time. The Gramians for this model class are given by the Lyapunov equations

$$\begin{aligned}APA^T - P + BB^T &= 0\\ A^TQA - Q + C^TC &= 0\end{aligned} \tag{1.7}$$

These Gramians can be balanced in the same way as in the continuous case. The error bound in 1.6 is also valid for this model class, as proven in [Al-Saggaf and Franklin, 1987].

This model reduction method is applied on a combustion engine model in Chapter 2.

Balanced truncation of linear time-varying systems

Balanced truncation has been extended to also cover the linear time-varying case, see [Verriest and Kailath, 1983; Shokoohi *et al.*, 1983]. For this class of linear systems, the matrices A, B, C and D are time varying.

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \tag{1.8}$$

The method follows the ideas of balanced truncation of time-invariant systems. For time-varying systems one can use the notion of controllability or observability over a time interval, say $t \in [0, T]$. The energy functions in (1.2) and (1.3) are then slightly modified. The controllability function is here the minimum required input energy to reach x_0 at time t starting from the zero state at $t = 0$.

$$L_c(x_0, t) = \min_{\substack{u \in L_2(0,t) \\ x(0)=0 \\ x(t)=x_0}} \frac{1}{2} \int_0^t ||u(\tau)||^2 d\tau$$

The observability function is the energy induced by the initial state $x(t) = x_0$ in the output signal over the time interval $[t, T]$, while the input signal is zero.

$$L_o(x_0, t) = \frac{1}{2} \int_t^T ||y(\tau)||^2 d\tau, \quad x(t) = x_0, \quad u \equiv 0$$

As in the time-invariant case, these functions are quadratic but the Gramians $P(t)$ and $Q(t)$ are now time dependant.

$$L_c(x_0, t) = \frac{1}{2} x_0^T P(t)^{-1} x_0 \qquad\qquad L_o(x_0, t) = \frac{1}{2} x_0^T Q(t) x_0$$

Furthermore, the time-varying generalization of the Lyapunov equations in (1.4) becomes

$$\begin{aligned} \frac{dP}{dt}(t) &= A(t)P(t) + P(t)A(t)^T + B(t)B(t)^T & \qquad P(0) &= 0 \\ \frac{dQ}{dt}(t) &= -Q(t)A(t) - A(t)^T Q(t) - C(t)^T C(t) & \qquad Q(T) &= 0 \end{aligned}$$

Once more, a balanced realization is achieved if a time-varying coordinate change yields diagonal and equal Gramians

$$\tilde{P}(t) = \tilde{Q}(t) = \tilde{\Sigma}(t) = \begin{bmatrix} \sigma_1(t) & & \\ & \ddots & \\ & & \sigma_n(t) \end{bmatrix}$$

The low order model is derived by truncating states corresponding to small singular values $\sigma_i(t)$. It is reasonable to let the reduced model order vary with time when $\sigma_i(t)$ is time varying. A priori error bounds, similar to (1.6), are available for the time-varying case, see [Lall and Beck, 2003; Sandberg and Rantzer, 2004].

This theory will be revisited in Chapter 3 , where the time-varying Gramians are used as tools to reduce nonlinear systems.

Balanced truncation of linear discrete-time time-varying systems

The counterpart to (1.8) is the linear discrete-time time-varying system

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k \\ y_k &= C_k x_k + D_k u_k \end{aligned} \quad k \in [1, N],$$

where x_k is the state vector, u_k the input signal and y_k the output signal at time k. Further, A_k, B_k, C_k and D_k are time-varying matrices of appropriate dimensions. Here the sub-index denotes time.

Here the *controllability energy function* becomes the optimal control problem

$$L_c(x^*, t) = \min_{\substack{u \in L_2(0,t) \\ x_1 = 0 \\ x_t = x^*}} \frac{1}{2} \sum_{k=1}^{t} ||u_k||^2. \tag{1.9}$$

That is, $L_c(x^*, t)$ is the minimal amount of energy in u required to reach a certain state x^* at time t, starting from the zero initial state.

Similarly, the *observability energy function* can in this case be stated as

$$L_o(x^*, t) = \frac{1}{2} \sum_{k=t}^{N} ||y_k||^2, \quad x_t = x^*, \quad u \equiv 0. \tag{1.10}$$

That is, the amount of energy an initial state x^* at time t induces in the output signal over the time interval $[t, N]$.

Similar to the continuous case, the energy functions can be determined through the quadratic forms

$$L_c(x^*, t) = \frac{1}{2} x^{*T} P_t^{-1} x^* \qquad\qquad L_o(x^*, t) = \frac{1}{2} x^{*T} Q_t x^*.$$

where the controllability Gramian P_k and observability Gramian Q_k are given by the Lyapunov equations

$$\begin{aligned} P_{k+1} &= A_k P_k A_k^T + B_k B_k^T, & k \in [1, N] \\ Q_k &= A_k^T Q_{k+1} A_k + C_k^T C_k, & k \in [1, N] \end{aligned}$$

with the boundary conditions $P_1 = 0$ and $Q_{N+1} = 0$. Also here a time-varying balancing coordinate change can be found and error bounds are available, see [Shokoohi and Silverman, 1987; Farhood and Dullerud, 2006].

This time-varying case will also be revisited in Chapter 3.

Descriptor form

Another linear state-space representation is the descriptor form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.11}$$

This representation can be transformed to the standard state-space form in (1.1) if E is invertible. However, if E and A are sparse $E^{-1}A$ can be dense and it might therefore be beneficial to keep the form in (1.11).

If E is singular, the system is a set of algebraic-differential equations and the problem becomes more involved. A generalization of the Gramians defined for the standard state-space form is presented in [Stykel, 2004]. Theory together with numerical methods are defined and also in this case an a priori error bound is available.

1.2 Model reduction of nonlinear systems

Model reduction of nonlinear systems is a research area under heavy development. To the authors best knowledge there is currently no method that generally provides guaranteed preserved stability or error bounds. Common to most methods is that they have their roots in methods developed for linear systems.

Heuristic methods

Probably the most common way to simplify nonlinear models is through heuristic methods. For example, indirect model reduction is performed in all modeling-work when complexity is chosen to match the intended model purpose. There are three common ways to reduce complexity:

- To discard effects that by intuition or experience have a relatively weak impression on the dynamics.
- Separation of time scales and replacing relatively fast dynamics with static gains.
- Averaging several effects into one pseudo-effect.

All three approaches require great knowledge and intuition of the modeled object. However, ways to perform these simplification steps in a systematic automatized manner have been investigated, see for example [Broz *et al.*, 2006]. The second mentioned method is more formally called the *singular perturbation method*. The differential equations of $\dot{x} = f(x,u)$ are divided into two parts, one relatively faster than the other

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, u)\\ \dot{x}_2 &= f_2(x_1, x_2, u)\end{aligned}$$

If x_2 corresponds to the fast dynamics, one introduces a factor ϵ according to

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, u)\\ \epsilon\dot{x}_2 &= f_2(x_1, x_2, u)\end{aligned}$$

and then set $\epsilon = 0$. The original system is now replaced with a set of differential algebraic equations with fewer states, for more details see [Khalil, 2002].

Linearization around equilibrium point or trajectory

In some applications the intended model usage is in the neighborhood of a certain operating point in state space. Then the detailed nonlinear model could be linearized at this point, giving rise to a linear model. This model can then be reduced with a linear reduction method.

Sometimes a nominal input signal is available and one is interested in the effect of deviations from this signal. A linearization around a trajectory in state space is then a valid approximation. This yields a time-varying system as in (1.8) and the theory of balanced truncation of linear time-varying systems can be applied. This procedure has been done successfully, see [Sandberg, 2006].

In both the time-invariant and time-varying case the reduced model will be linear and it will only be a valid approximation in a region close to the operating point. Further, the size of this region depends on how nonlinear the original system is.

Balancing nonlinear systems

Balancing using energy functions An extension to nonlinear systems of the mentioned balanced truncation method is proposed in [Scherpen, 1993]. Here nonlinear systems of the form

$$\begin{aligned}\dot{x} &= f(x) + g(x)u\\ y &= h(x)\end{aligned} \tag{1.12}$$

are considered. Again, the controllability and observability functions in (1.2) and (1.3) are used. For the given nonlinear system it can be shown that, under some conditions, $L_c(x)$ and $L_o(x)$ are the unique smooth solutions of

$$\frac{\partial L_c}{\partial x}(x)f(x) + \frac{1}{2}\frac{\partial L_c}{\partial x}(x)g(x)g^T(x)\frac{\partial^T L_c}{\partial x}(x) = 0, \quad L_c(0) = 0$$

and

$$\frac{\partial L_o}{\partial x}(x)f(x) + \frac{1}{2}h^T(x)h(x) = 0, \quad L_o(0) = 0$$

After a coordinate transformation, $x = \psi(z)$, the functions can be written

$$\tilde{L}_c(z) = \frac{1}{2}z^T z \qquad \tilde{L}_o(z) = \frac{1}{2}z^T \begin{bmatrix} \tau_1(z) & & \\ & \ddots & \\ & & \tau_n(z) \end{bmatrix} z$$

This form is not balanced, in the linear case it is sometimes called "input normalized". However, an additional coordinate change can balance L_c and L_o. For more details see [Scherpen, 1993] and [Scherpen and Fujimoto, 2003]. In analogy with the linear case, the functions $\tau_1(z) \geq \cdots \geq \tau_n(z)$ are called the singular value functions of the system. Model reduction is performed by truncating states in the balanced form corresponding to small singular functions.

A linearized version of the method applied to a linear system yields the same result as the standard balanced truncation method would. Further, the singular value functions become constant and $\tau_i(z) = \sigma_i$ as given in (1.5).

Extensions have been made to the discrete-time counterpart in [Scherpen and Fujimoto, 2004] and nonlinear differential-algebraic systems in [Sjöberg *et al.*, 2007]. A recent contribution to this problem setting is, among others, [Fujimoto and Tsubakino, 2006]. Another contribution, featuring error bounds for certain input signals, is found in [Krener, 2008].

The method has strong mathematical support but due to the required numerical effort only models with very moderate size have so far been considered, see e.g. [Newman and Krishnaprasad, 1998].

Balancing using empirical Gramians A model reduction method for nonlinear systems of the form

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x) \end{aligned}$$

is treated in [Lall *et al.*, 2002; Hahn and Edgar, 2002]. The approach applies ideas concerning linear systems, introduced in [Moore, 1981], to nonlinear systems.

Here state data are collected while impulse input signals in different directions are injected. The data are then used to estimate a constant controllability Gramian matrix. Similarly, a constant observability Gramian matrix is constructed from simulation data generated by different initial values distributed on the unit sphere.

When the Gramians have been computed they are balanced using linear theory, see Section 1.1. The reduced nonlinear model is then derived by applying the corresponding linear coordinate change $Tz = x$

$$\begin{aligned} \dot{z} &= T^{-1} f(Tz, u) \\ y &= h(Tz) \end{aligned}$$

followed by truncation of states, as in the linear case. This method also yields the same reduced system as standard linear balanced truncation if applied to a linear system. In [Liu and Wagner, 2002] the method is applied on an automotive model. In [Hahn *et al.*, 2003] the method is extended so that the input signals do not have to be impulses.

The method is much less computationally intensive than the method using nonlinear balancing. However, the heuristic use of simulations does not leave much room for proofs and analysis.

Proper orthogonal decomposition

Karhunen-Loève expansion [Karhunen, 1946; Loève, 1945], or proper orthogonal decomposition (POD), is a model reduction method for state-space models based on principal component analysis. The method was pioneered for applications in turbulence models in [Lumley, 1967] and is one of the most commonly used tools for model reduction of nonlinear systems. It uses simulation data to find a low-dimensional subspace that captures most of the state dynamics. Figure 1.1 illustrates a possible truncation of state space, (x_1, x_2) to $\hat{x}_2 = 0$.

The method can briefly be described in three steps.

1. Simulate the nonlinear system

$$\dot{x} = f(x, u)$$

and collect snapshots of the state vector in a matrix X.

$$X = [\, x(t_0) \quad x(t_1) \quad \dots \quad x(t_N) \,],\, x(t) \in \mathbf{R}^n$$

2. Factorize X with the singular value decomposition

$$U\Sigma V^T = X$$

3. Choose truncation level after size of singular values in Σ. Truncate $U \in \mathbf{R}^{nxn}$ to $\hat{U} \in \mathbf{R}^{nx\hat{n}}$ so that $x \approx \hat{U}\hat{x}$ where $\hat{x} \in \mathbf{R}^{\hat{n}}$. Then the reduced model becomes

$$\dot{\hat{x}} = \hat{U}^T f(\hat{U}\hat{x}, u) \tag{1.13}$$

This method lacks general error bounds, which can easily be demonstrated. Put short, a state can be important even though it is small. For example, scaling of states by a diagonal coordinate change

$$x^* = \mathrm{diag}(c_1, c_2, ..., c_n)x \quad c_i > 0$$

does not change the dynamic behaviour of the system but can make the method choose an arbitrary subspace. Further, all states are usually not interesting for control purposes and this method does not take any output signal into consideration, so one is forced to use a larger subspace than might be necessary.

A common source of large models is discretization of Partial Differential Equations (PDE's), where the states share the same physical units. In this case size comparison might be feasible. More details and numerous examples can be found in [Astrid, 2004].

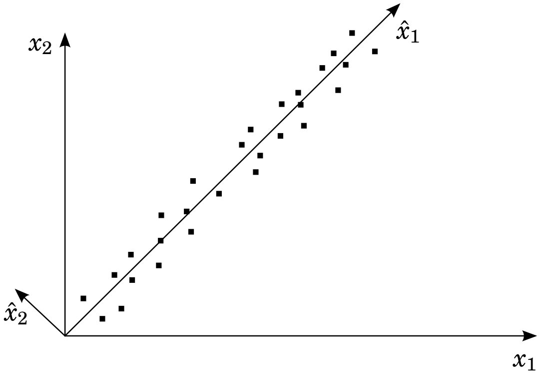

Figure 1.1 $\hat{x}_2 = 0$ is a dominant state subspace in (x_1, x_2)

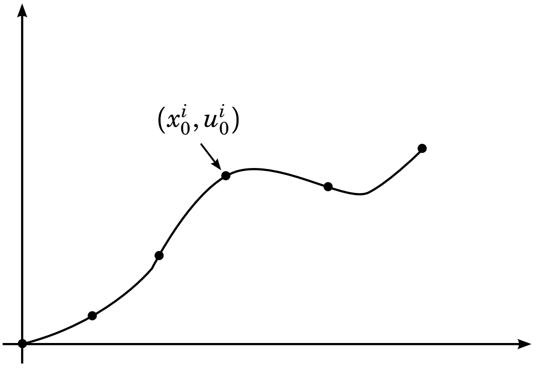

Figure 1.2 Linearizations distributed over a training trajectory

Trajectory piecewise-linear model reduction

A novel approach to nonlinear model reduction is presented in [Rewieński, 2003]. The method is based on linearizations distributed over one or many training trajectories.

First one simulates the nonlinear system

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

with a training input signal $u_0(t)$. Then a set of linearization points (x_0^i, u_0^i) is chosen along the training trajectory, see Figure 1.2. How to pick the location of the points will soon be discussed. Observe that the points are in general not equilibrium points for the system.

Close to the point i, the linearization

$$\begin{aligned} \dot{x} &\simeq f(x_0^i, u_0^i) + A_i(x - x_0^i) + B_i(u - u_0^i) \\ y &\simeq g(x_0^i, u_0^i) + C_i(x - x_0^i) + D_i(u - u_0^i) \end{aligned} \tag{1.14}$$

approximates the nonlinear system, where the matrices A_i, B_i, C_i and D_i are the partial derivatives

$$\begin{aligned} A_i &= \frac{\partial f}{\partial x}(x_0^i, u_0^i) & B_i &= \frac{\partial f}{\partial u}(x_0^i, u_0^i) \\ C_i &= \frac{\partial g}{\partial x}(x_0^i, u_0^i) & D_i &= \frac{\partial g}{\partial u}(x_0^i, u_0^i) \end{aligned}$$

The local linear approximation (1.14) can be rewritten as

$$\begin{aligned} \dot{x} &\simeq f_i(x,u) \\ y &\simeq g_i(x,u) \end{aligned}$$

Now let the original nonlinear system be approximated in a less local way by a weighted sum of the local linearizations.

$$\begin{aligned} \dot{x} &\simeq \sum_i w_i(x,u) f_i(x,u) = \hat{f}(x,u) \\ y &\simeq \sum_i w_i(x,u) g_i(x,u) = \hat{g}(x,u) \end{aligned} \tag{1.15}$$

The weighting function $w_i(x,u)$ is close to one in the neighborhood of linearization i and zero otherwise. Additionally, $w_i(x,u) \geq 0$ and $\sum_i w_i(x,u) = 1$ for all x and u.

There are several ways to decide the location of the linearization points, one approach is Algorithm 1.1, which will be later used in Chapter 3. Observe that (1.16) is not necessarily fulfilled for the whole trajectory even though the relative approximation error is held below the threshold ϵ at all linearization points.

Algorithm 1.1: Choice of linearization point locations

1. Generate linearized model at the initial state
2. Traverse the training trajectory while
$$\frac{|f(x,u) - \hat{f}(x,u)|}{|\hat{f}(x,u)|} < \epsilon \tag{1.16}$$
3. If (1.16) is not fulfilled, aggregate point to linearization collection
4. If x is not the final state return to step 2

So far the original nonlinear system has been approximated but no gain in terms of state dimension or simulation time has been achieved. In this step linear model reduction theory is used to reduce the local linear models.

In [Rewieński, 2003] the Krylov subspace method was used. This method uses the Arnoldi algorithm, which is numerically effective even for very large systems. However, it has the drawback of not generally provide guaranteed preserved stability or error bounds, see [Grimme, 1997]. The use of balanced truncation has also been investigated, see [Vasilyev *et al.*, 2006]. The Krylov subspace method generates an orthonormal projection $z = Wx, W \in \mathbf{R}^{\hat{n}xn}$, which is used globally to reduce all the local models. For further details see [Rewieński, 2003].

Introducing the new coordinates in (1.15) yields

$$\dot{z} = \sum_i w_i(W^T z, u) W f_i(W^T z, u)$$
$$y \simeq \sum_i w_i(W^T z, u) g_i(W^T z, u)$$

This reduced model has fewer states, $\hat{n} < n$. The simulation time is also improved because f_i and g_i are linear. Additionally, most of the weighting functions are zero and there is no need to evaluate the rest of those expressions.

Model reduction through system identification

One alternative way of performing model reduction is to use system identification. System identification is the process of estimating a dynamic model from input and output data. That is, instead of using the internal description of the original model one could estimate its dynamics only using simulation data of input and output signals. This is, in particular, beneficiary when the model description is in a format that is hard to handle, e.g. software code. Model reduction is performed through restricting the complexity of the estimated model to one smaller than the original model. Most system identification methods are defined for discrete-time systems, since simulation and measurement data typically are discrete.

The methodology is applicable to both linear and nonlinear systems. The main difficulty in nonlinear system identification is the fact that a system can be nonlinear in many different ways. It is difficult to find a simple yet general model class. When system identification is used as a tool for model reduction one could try to use internal information of the model to choose how to parametrize the nonlinearities.

Recent contributions in [Zhang *et al.*, 2006; Sou *et al.*, 2008] proposes methods for identification of a block structure with linear and nonlinear parts where no a priori parametrization of the nonlinearity is necessary.

1.3 Summary

In this chapter, a brief background of some methods for linear and nonlinear model reduction has been presented. Model reduction of linear systems is a well developed research area. Methods as balanced truncation provide error bounds and guaranteed preserved stability. Others, e.g. Krylov subspace method [Grimme, 1997], do not but are numerically more adapted to large systems.

All the mentioned methods have the property that internal physical interpretation of the model is lost in the reduction process. Methods that also preserve internal properties include e.g. [Moin and Uddin, 2004] and structure preserving methods such as in [Vandendorpe and Van Dooren, 2004; Li and Paganini, 2005; Sandberg and Murray, 2008].

How to reduce nonlinear systems is however still a quite open problem and there is a large room for improvement of existing methods. Theorems concerning preserved stability or error bounds are sparse. As a general rule nonlinear methods tend to rely on linear ones.

Common for all mentioned nonlinear methods in this chapter, except the piece-wise linear approach, is that even though the order is reduced, simulation time is not necessarily shorter. Commonly, the original set of equations is sparse, i.e. all state equations do not involve all states. The sparsity is lost with a dense coordinate change and truncation of states. Therefore, the total computation time is not necessarily reduced for the right-hand-side functions, which can e.g. be seen in [Liu and Wagner, 2002].

2

A model reduction case study

The contents of this chapter is based on the article [Nilsson *et al.*, 2006].

Low complexity plant models are essential for model-based control design. Often a detailed high order model is available and simplification to a low order approximate model is needed. This chapter presents a case study of two model reduction methodologies applied on the automotive engine air path. The first methodology is based on balanced truncation of models obtained by linearization around equilibria and trajectories. Under appropriate assumptions, this technique yields strict bounds on the approximation error. The second is a heuristic methodology, based on intuition commonly used in modeling of engine dynamics. Although it is successfully used in practice, the approximation error is seldom known. The two methodologies are used to derive simple models for the required fuel charge in a spark ignition engine, given throttle and swirl flap positions and engine speed. Performance, complexity and similarities of the two resulting low order models are compared.

2.1 Introduction

The air path dynamics is a major challenge in automotive engine control. The main problem for spark ignition (SI) engines is to regulate the Air/Fuel Ratio. The electronic fuel control system of a modern SI automobile engine employs individual fuel injectors located in the inlet manifold runners close to the intake valves to deliver precisely timed and accurately metered fuel to all cylinders. This fuel management system acts in concert with the three-way catalytic converter (TWC) to control HC, CO, and NO_x emissions. Figure 2.1 illustrates the conversion efficiencies provided by a

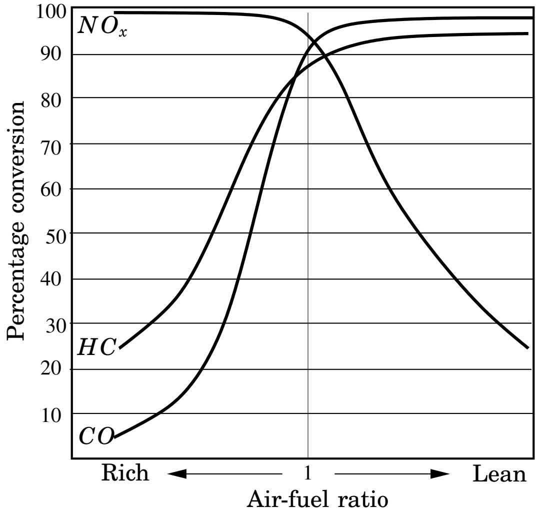

Figure 2.1 Typical conversion efficiency of a three-way catalyst

typical TWC as a function of exhaust air-fuel ratio (A/F) for the three constituents. It can be seen that there is only a very narrow range of A/F near the stoichiometric value (14.64) over which high simultaneous conversion efficiencies may be attained, see [Heywood, 1988]. Feedback from an exhaust gas oxygen (EGO) sensor, treated in Chapter 4, is used to utilize the TWC effectively, see Figure 2.2. If the operating point is changed by, for example, an increased torque demand, the injected fuel amount has to increase. The EGO sensor will not instantly detect the unbalance in the A/F ratio and a good feed-forward control is needed to adapt the injected fuel amount before the deviation is detected by the sensor. To this purpose, a good low complexity model of the air entering the cylinder is essential. Given throttle position, swirl flap position and engine speed the required fuel charge (Fc) has to be estimated to achieve stoichiometric conditions.

2.2 Model properties

The article is focused on the setup shown in Figure 2.3, which shows an illustration of the air path. The throttle is used to get the desired airflow and the swirl flap is used for inducing turbulence and thereby achieving better mixing in the cylinder. Volume 1 and 2 in the figure represent the connecting pipes between the elements.

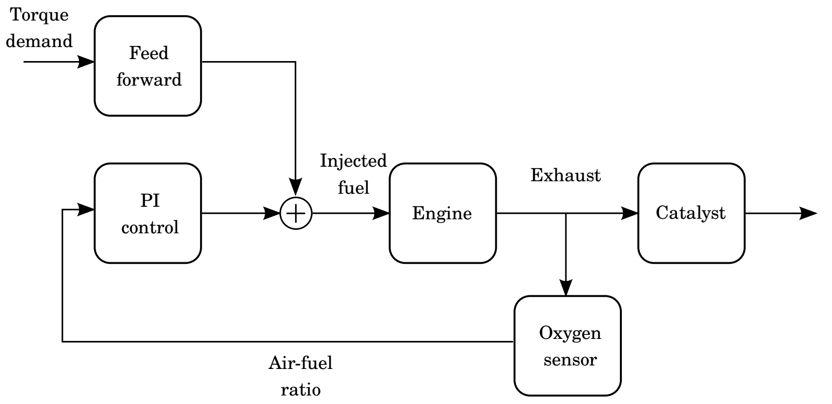

Figure 2.2 A standard air-fuel ratio control scheme

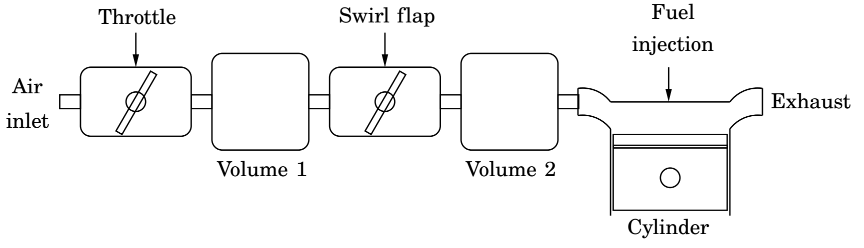

Figure 2.3 Schematic of the engine air path

The base for the model reduction is a detailed one-cylinder model provided by Toyota Motor Corporation. It is written in the Modelica language, see [Fritzson, 2004], and managed with the software tool Dymola, a multi-domain modeling and simulation tool, see [Dynasim AB, 2006]. The model is based on conservation laws such as mass balances. The top view of the Dymola model can be seen in Figure 2.4, which shows the same physical layout as Figure 2.3.

Translating the model, Dymola induces a nonlinear differential algebraic equation (DAE) with 37 continuous-time states, distributed as

- 11 states in volume 1
- 11 states in volume 2
- 15 states in the cylinder

The states in the three objects are among other things mass, energy, momentum and concentrations of the seven species gas mixture. The model

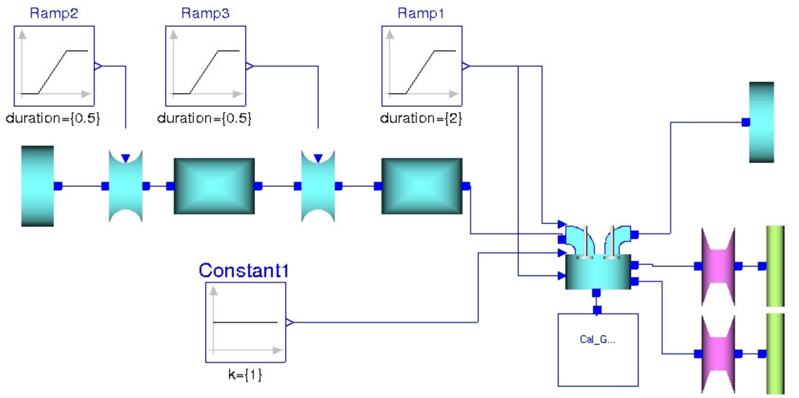

Figure 2.4 Top view of the one cylinder Dymola model

is ideal for simulation but is too complex for model-based control design. By applying model reduction techniques a low-order approximate model can be derived.

2.3 Model reduction by balanced truncation

In this section the balanced truncation method is applied to obtain a low order approximate model. The theory of balanced truncation, as described in section 1.1, is clearly not directly applicable on the Dymola model as it is non-linear, hybrid and is defined both by equations and algorithms. However, with the below described methodology an approximate low order model is obtained. The methodology can be separated into three steps.

Obtain a linear time-varying system by repeated linearization The Dymola model was simulated with constant input signals, i.e. throttle position, swirl flap position and engine speed. The simulation gave rise to a state trajectory around which the non-linear model can be linearized. Due to the cyclic behaviour of the cylinder, the linearized model is time-varying, i.e., the A, B, C and D matrices are time dependent.

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \tag{2.1}$$

The state vector x is the deviation from the nominal state trajectories and all 37 states have known physical interpretations. The fuel charge is defined once per engine cycle (when the inlet valve closes) and is proportional to the amount of oxygen in the cylinder. To linearize the system at any time point a continuous-time version of the signal was defined that coincides with the discrete when the inlet valve closes.

Dymola has the functionality of derivation of linearizations of the nonlinear DAE. By using Dymola's scripting capabilities this can be done repeatedly at times t_k and snapshots, with $50\mu s$ intervals, of the continuous linear time-varying system in (2.1) is obtained. With the assumption that the $A(t)$,$B(t)$,$C(t)$ and $D(t)$ matrices are constant between the times t_k a discrete linear time-varying (LTV) system can be derived by zero order hold sampling. The discrete-time system (2.2) only captures the state vector at the snapshot times, $x(t_k) = x_k$.

$$\begin{aligned} x_{k+1} &= \Phi_k x_k + \Gamma_k u_k \\ y_k &= C_k x_k + D_k u_k \end{aligned} \tag{2.2}$$

Resample the discrete-time LTV system once per engine cycle The required fuel charge is defined once per engine cycle and the system is therefore sampled. Resampling, with the sampling periods defined by the closing of the inlet valve, gives rise to a cycle-to-cycle model for the fuel charge. Here the input signals are assumed to be constant during the cycle and n denotes the number of sampling intervals per cycle.

$$x_{k+n} = \tilde{\Phi} x_k + \tilde{\Gamma} u_k$$

Letting $k = 0$ represent the first sample time in the cycle the matrices $\tilde{\Phi}$ and $\tilde{\Gamma}$ can be computed according to

$$\begin{aligned} \tilde{\Phi} &= \prod_{i=1}^{n} \Phi_{n-i} \\ \tilde{\Gamma} &= \sum_{i=0}^{n-2} \left(\prod_{j=1}^{n-i-1} \Phi_{n-j} \right) \Gamma_i + \Gamma_{n-1} \end{aligned}$$

Linearization around trajectories captures hybrid phenomena such as dynamics changing depending of time and position in state space, but not instantaneous changes such as reset maps. The Dymola model contains reset maps, for example the cylinder mass is instantaneously increased by the amount of fuel injected, similarly the oxygen amount is reset after combustion. These two and others all occur when the inlet valve closes and can all be represented as an instantaneous linear transformation of the state vector

$$x_k^* = H x_k$$

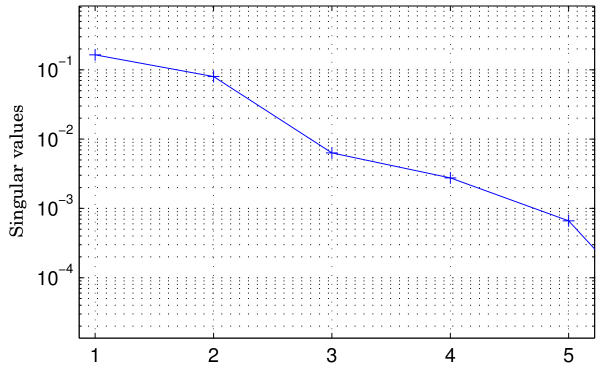

Figure 2.5 The five largest singular values of the balanced realization

This can be included in the LTV system by introducing the transformation in the beginning of the cycle, i.e., when the inlet valve closes.

$$x_1 = \Phi_0 H x_0 + \Gamma_0 u_0$$

$\tilde{\Gamma}$ is not affected and the only alteration is in the calculation of $\tilde{\Phi}$, which becomes

$$\tilde{\Phi} = \left(\prod_{i=1}^{n} \Phi_{k+n-i} \right) H$$

Only slight cycle-to-cycle variations can be seen in $\tilde{\Phi}$ and $\tilde{\Gamma}$ but to improve numerical precision the matrices are calculated by averaging over several cycles.

It should be mentioned that an alternative way to obtain the linear time-invariant model would be to integrate the continuous-time model instead of through sampling and iteration.

Apply balanced truncation to obtain low-order model By linearization and resampling a linear time-invariant model for the required fuel charge has been derived. And now the balanced truncation method, as described in Section 1.1, can be applied.

The model has 37 states, a number that can significantly be reduced without much loss of accuracy. The five largest Hankel singular values described by (1.5) are shown in Figure 2.5. The plot indicates how well the model can be represented with a lower-order approximation. How many states the low-order model should have is a trade off between approximation error and model complexity. In this case simulation shows that

Table 2.1 The most important physical states

State number	Physical interpretation
11	Amount of oxygen in cylinder
15	Mass in volume 1
21	Amount of oxygen in volume 1
32	Amount of oxygen in volume 2

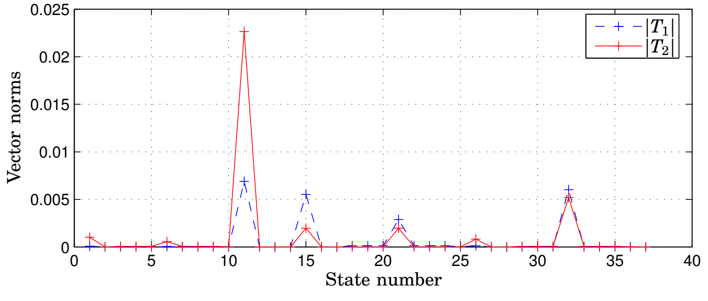

Figure 2.6 Relative importance of physical states in the reduced model

it is reasonable to truncate all but two states. The new low-order state vector will be a linear combination of the physical states and reducing to a second order system yields the coordinate change

$$\bar{x} = Tx = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} x$$

where $T \in \mathbf{R}^{2x37}$.

The absolute value of the elements in T_1 and T_2 can be seen in Figure 2.6, which indicates the relative importance of the physical states in the reduced model[1]. The most important states are listed in Table 2.1.

Results

The nonlinear Dymola model with 37 states has been approximated with a two-state linear time-invariant system. Figure 2.7 shows the required fuel charge computed by the original Dymola model, the linearized model and the reduced model as a response to the illustrated change of input signals.

[1]For numerical reasons the states share approximately the same magnitude.

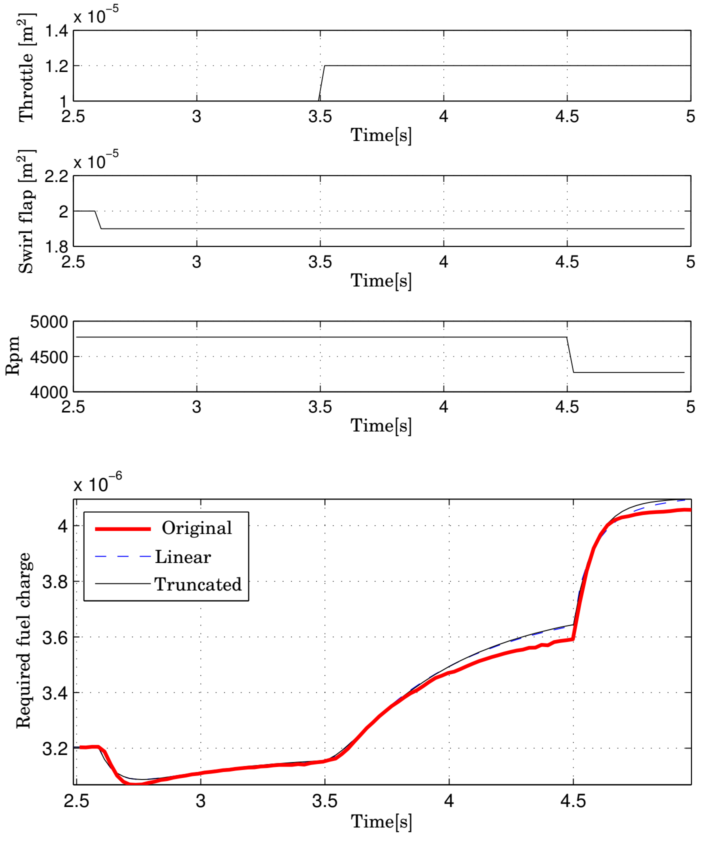

Figure 2.7 Simulation results for the linearized and truncated model

As can be seen, both the linearized and reduced model approximates well the Dymola model result. For this trajectory the approximation error is dominated by the linearization and not the truncation.

Equation (1.6) gives a bound on the approximation error between the two linear models and keeping two states yields

$$\max_u \frac{\|\tilde{y}(t) - y(t)\|_2}{\|u(t)\|_2} \leq 2 \sum_{k=3}^{37} \sigma_k \tag{2.3}$$

where $\tilde{y}$ is the reduced model output. In this case, the input signal $u(t)$ consists of three scalar signals, throttle position, swirl flap position and engine speed. The norm is calculated according to

$$\|u(t)\|_2 = \sqrt{\int_0^\infty u^T(t)u(t)dt}$$

The throttle and swirl flap positions are given in effective area $[m^2] \sim 10^{-5}$ and engine speed in [rpm]$\sim 10^3$, as can be seen the engine speed will greatly dominate the input signal norm. To achieve more reasonable results the inputs are balanced by including a 10^{-8} gain before the throttle and swirl flap positions in the model, which could correspond to a unit change. Now the input signals all have the approximate magnitude of 10^3 and if all but two states are truncated, according to (2.3), the following holds

$$\|\tilde{y}(t) - y(t)\|_2 \leq 2.6784 \cdot 10^{-10} \|u(t)\|_2$$

$y(t)$ and $\tilde{y}(t)$ denotes the outputs of the 37- and 2-state linear models. For the trajectory in Figure 2.7 this implies that the output error is bounded as

$$\|\tilde{y}(t) - y(t)\|_2 \leq 5.0 \cdot 10^{-6}$$

while the actual approximation error is $6.9 \cdot 10^{-7}$. This is not uncommon when applying balanced truncation, the result is often much better than the error bound indicates. There are two reasons for the difference, that the norm is worst-case and that the bound is often conservative.

2.4 Heuristic model reduction

A common way to accomplish model reduction is to use experience and insight into the physics to omit dynamics with little importance. In this case the problem is split into two parts

1. The cylinder dynamics
2. The air path including the two volumes

where the first part has much faster dynamics than the second. Therefore the common technique of time scale dynamics reduction can be performed on the first part. The result is a static mapping from pressure in volume 2 and engine speed to required fuel charge. Simplification of more detailed physical modeling yields that the dynamics of the second part can be modeled as two first order dynamics. The states represent the pressure in each volume and the throttle and swirl flap positions act as input signals in a similar configuration as in Figure 2.3. The effect of varying engine speed is introduced in the model by letting the gains and time constants of

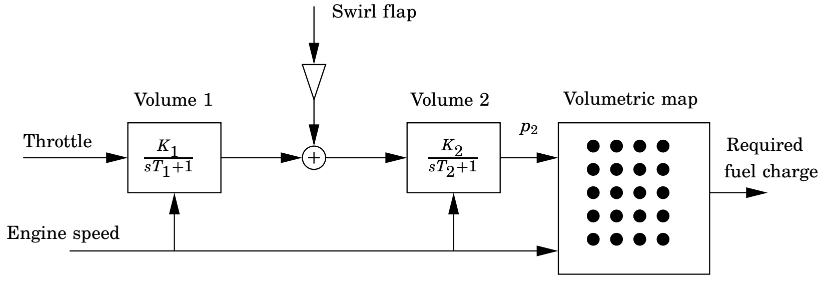

Figure 2.8 Structure of the heuristically derived model

the first order dynamics be dependent of engine speed, N_e. The complete model is the combination of the two parts and is illustrated in Figure 2.8.

A more detailed description of how the model structure was obtained for the two parts is presented below. For further details and background see [Chevalier *et al.*, 2000; Hendricks *et al.*, 1996] and [Føns *et al.*, 1999].

Time scale dynamics reduction

One simple model of the air flow in an intake manifold is the filling and emptying model. The air flow enters the manifold through the throttle and is pumped out of the manifold into the cylinder. Assuming no leaks, the intake mass air flow D_{air}, into the manifold and the flow entering in the cylinder, M_{asp} are identical only in steady state.

Pumping fluctuations Pumping fluctuations are caused by any disturbance initiated at the boundary of inlet manifold such as moving piston, moving valve and moving throttle plate. These disturbances travel along the pipe experiencing many reflections. When the engine is operated in the steady state, they finally settle down into a standing wave. The source of pumping noise is periodic, so the pumping fluctuations are frequency locked to the engine event frequency. Looking at top-dead-center (TDC) gas dynamics leads not to consider these fluctuations.

Mean model of the aspirated flow In spite of the complexity of the fluid dynamic phenomena occurring during a transient (due to fast opening or closing of the throttle), the conventional volumetric efficiency η (function of the engine working point), identified during steady-state conditions, is used to describe the inlet air mass flow rate. So the speed-density gives an accurate description of the air mass flow rate through the inlet valve

$$D_{\mathrm{asp,map}} = \eta_{\mathrm{map}}(\bar{P}_2, N_e)\frac{V_{\mathrm{cyl}}\bar{P}_2}{R\bar{T}_2}\frac{N_e}{120}$$

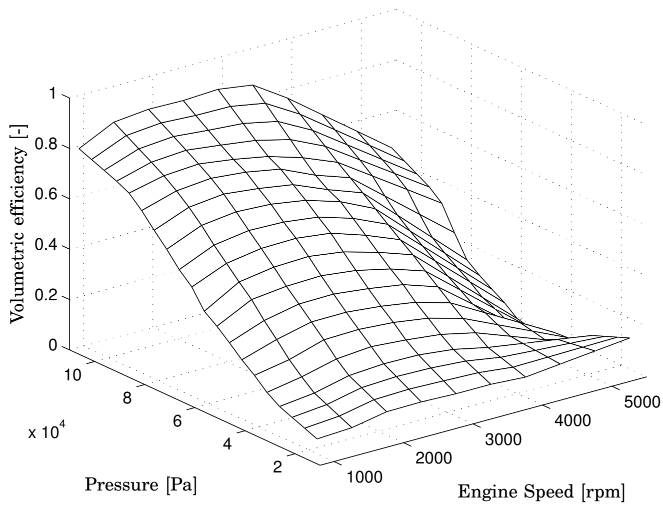

Figure 2.9 The volumetric efficiency map

where:

- $\bar{P}_2$ is the mean pressure over a TDC in volume 2
- η_{map} is the volumetric efficiency
- $\bar{T}_2$ is the mean temperature over a TDC in volume 2
- N_e is the engine speed
- V_{cyl} is the cylinder volume
- R is the universal gas constant

The volumetric efficiency η_{map} is highly nonlinear function of the engine speed (N_e) and manifold pressure ($\bar{P}_2$). It can only be estimated via experimentation. Figure 2.9 shows the volumetric efficiency of a commercial gasoline engine. The used volumetric efficiency should preferably been generated from the Dymola model. The map derived by experiment data is considered to be accurate enough for the purposes of this article.

Air path dynamics reduction

During throttle transients, the difference between these two flows equal the rate of change of the air mass in the manifold plenum. Assuming that the manifold pressure is uniform and the intake manifold temperature is constant, the continuity equation and ideal gas law can be applied to the

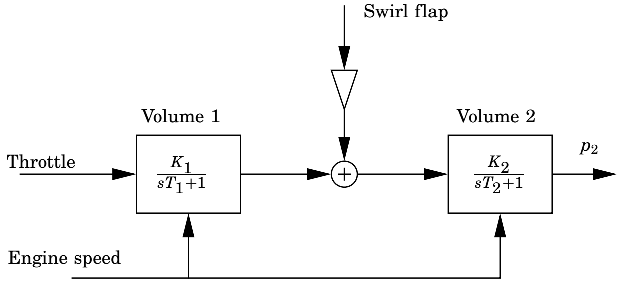

Figure 2.10 Air path dynamics approximation of the two volumes

manifold plenum. Usually, the flow is defined as a function of the total mass M_T and can be factorized as

$$d(M_T, O_{valve}) = p(M_T)M_T$$

where O_{valve} is the effective area of the valve and p is a positive increasing (concave) function with respect to the total mass M_T as proposed in [Heywood, 1988],

$$p(z, O_{valve}) = p_0(O_{valve})\sqrt{2\frac{\gamma}{\gamma-1}((\frac{z}{z_0})^{-\frac{2}{\gamma}} - (\frac{z}{z_0})^{-\frac{\gamma+1}{\gamma}})}$$

Here z_0 is the mass in atmospheric conditions, γ is the heat ratio and p_0 is a function transforming the valve opening into the effective area of the valve. By linearization of the flow equation, the model of the two volumes writes as two first order dynamics with the states p_1 and p_2 (the pressure in the volume 1 and 2 respectively) as described in Figure 2.10. Although the parameters depend on engine speed, the calibration task is easier than for the original nonlinear model. Moreover, first-order dynamics (with the parameters varying with respect to the operating conditions) is well representative of the filling-emptying dynamics.

The combination of the time-scale reduction and the volume dynamics yields the total model showed in Figure 2.8.

Results

The nonlinear Dymola model with 37 states has been approximated with a second-order linear parameter varying system combined with a look-up table. Figure 2.11 shows the required fuel charge computed by this model as a response to the illustrated change of input signals. The model is able to approximately describe the variation of the requested fuel charge.

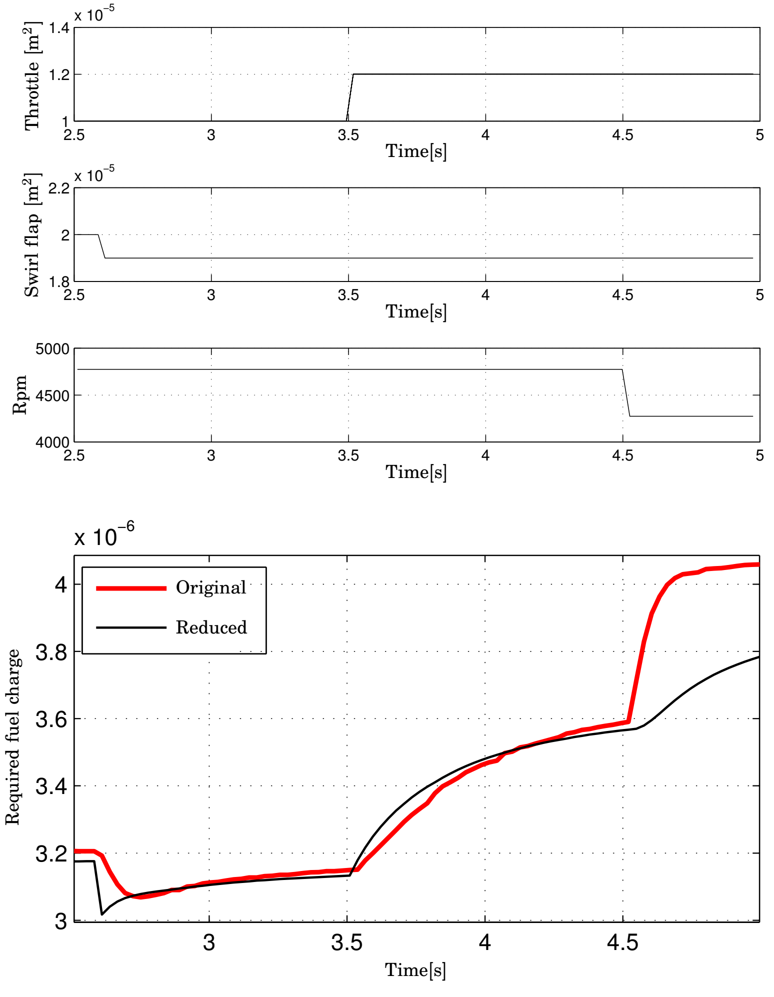

Figure 2.11 Simulation results for the heuristically derived model

Nevertheless, the qualitative response is not so good. Indeed, without observers and correction mapping, a precise estimation of the aspirated flow is not available. It means that the air-fuel ratio controller action will be necessary and predominant as the requested fuel charge is not well predicted.

2.5 Methodology comparison and conclusions

Two low-order models have been derived, one using balanced truncation and one using heuristic methods. The complexity of the two resulting models is almost the same but the methodology is quite different.

Methodology comparison

In the case of the heuristic procedure the methodology is based on modeling and simplifications are made based on intuition and experience. When an appropriate model complexity is chosen, parameters are determined by physical properties or, as in this case, by tuning to fit simulation data. The tuning could also been done to fit experimental data, that is not the case for balanced truncation which needs a detailed model.

The resulting model from the balanced truncation based technique is always a linear time-invariant system, the heuristic procedure does not have this restriction and has therefore greater potential. On the other hand, it can be very hard to tune the parameters, especially for larger nonlinear systems. This time-consuming tuning can be compared to the rather heavy computations needed to derive the linearizations required for balanced truncation, which can be carried out by a computer without human supervision.

The balanced truncation methodology is relatively systematic and does not need physical knowledge of the model, neither is any parameter fitting necessary. It also delivers a bound on the approximation error compared to the linearized model.

In this example better performance was achieved with balanced truncation, this is however more a question of how well you can fit the parameters in the heuristics based model. In cases when nonlinear dynamics are essential the balanced truncation technique used here will not be sufficient.

Similarities in choice of states can be seen in the two methods. For example, both methods neglect the fast dynamics occurring in the cylinder and other gas species than oxygen are ignored. The heuristic method chose the volume pressures as states, which are (at constant temperature) proportional to the amount of oxygen. For the model generated by balanced truncation, the oxygen concentrations are present as components in the two states.

Conclusions

Both methodologies have their advantages and disadvantages. If a detailed model is available and linear behaviour is expected then the balanced truncation technique could be preferred. This technique can require a large computation time but needs very little manual attention. Using the heuristic method requires more experience and knowledge, it may also involve extensive parameter fitting, but renders more insight to the simplifications made.

3

The average Gramian approach to nonlinear model reduction

In Chapter 2, model reduction of an automotive model was performed. Despite promising results, the final model is linear and can only be assumed to be accurate within a certain operating range. This chapter is based on [Nilsson and Rantzer, 2009a] and addresses the problem of state reduction of nonlinear continuous-time and discrete-time systems. A novel method that relates to balanced truncation is presented and applied to examples. The method is computationally efficient and is applicable to relatively large systems.

3.1 Introduction

As mentioned in Chapter 2, model reduction is an attractive tool in many contexts. Despite the increase in computing power a large model complexity still is a potential problem. In particular, costs of embedded systems hardware imply strong restrictions on memory and performance. A low complexity is very beneficial when analysis is performed on the model, such as reachability analysis or optimization.

Model reduction of linear systems is a mature research topic and well-known methods featuring error bounds and preserved stability are available. However, in practice, one is often confronted with nonlinear systems and model reduction for this model class is so far a relatively open research problem. Here, a new method for simplification of nonlinear input-output models is outlined. The method relates to balanced truncation and uses a state transformation followed by truncation of some states. First the continuous-time case is treated, then the discrete-time counterpart.

3.2 The continuous-time case

Preliminaries

The method presented here is based on theory concerning linear time-varying systems and the theory presented in Section 1.1 will be used as a base. Consider the linear continuous-time time-varying system

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \quad t \in [0, t_f], \tag{3.1}$$

where x is the state vector, u the input signal and y the output signal. Further, A, B, C and D are time-varying matrices of appropriate dimensions. As in [Scherpen and Fujimoto, 2003] the notion of so called energy functions is used. The *controllability energy function* is the amount of energy required in the input-signal to reach a specific state. In the linear time-varying case this can be stated as the optimal control problem

$$L_c(x_0, t) = \min_{\substack{u \in L_2(0,t) \\ x(0)=0 \\ x(t)=x_0}} \frac{1}{2} \int_0^t ||u(\tau)||^2 d\tau. \tag{3.2}$$

That is, $L_c(x_0, t)$ is the minimal amount of energy in u required to reach a certain state x_0 at time t, starting from the zero initial state.

Further, the *observability energy function* determines the energy induced in the output, given a certain initial state and a zero input signal. In this case it can be stated as

$$L_o(x_0, t) = \frac{1}{2} \int_t^{t_f} ||y(\tau)||^2 d\tau, \quad x(t) = x_0, \quad u \equiv 0 \tag{3.3}$$

This measures the amount of energy an initial state x_0 at time t induces in the output signal over the time interval $[t, t_f]$. The concept of these energy functions is illustrated in Figure 3.1. The usefulness of these functions for model reduction is clear. If a large amount of input energy is required to reach a certain state and if the same state yields a small output energy, then this state is unimportant for the input-output behaviour of the system.

The energy functions can be determined through the following Lyapunov equations

$$\begin{aligned} \dot{P}(t) &= A(t)P(t) + P(t)A^T(t) + B(t)B^T(t) \\ \dot{Q}(t) &= -Q(t)A(t) - A^T(t)Q(t) - C^T(t)C(t) \end{aligned}$$

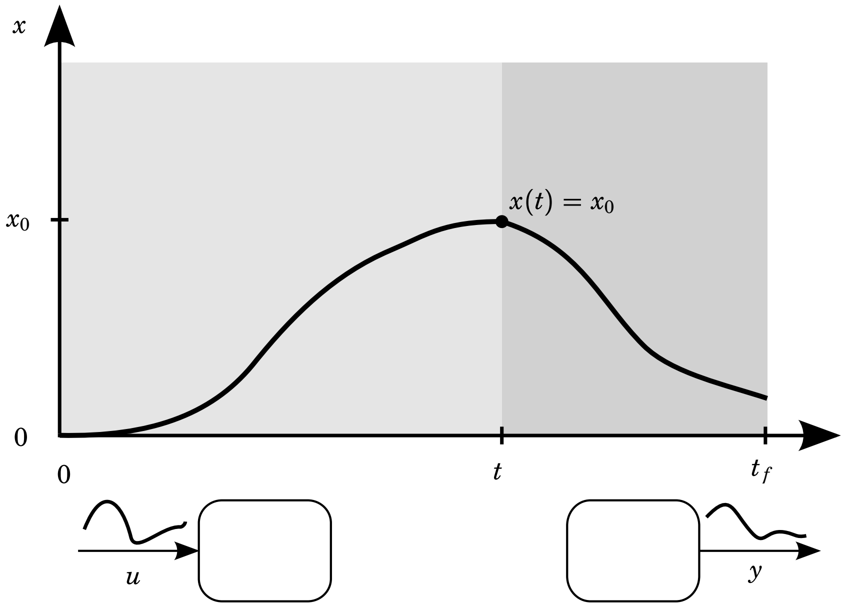

Figure 3.1 Visualization of the energy functions. The left part illustrates the minimal input energy required to reach x_0 at time t. In the right part the initial state x_0 yields the mentioned output energy while the control signal is zero.

with $t \in [0, t_f]$ and the boundary conditions $P(0) = 0$ and $Q(t_f) = 0$. The matrices P and Q are commonly called the controllability Gramian and observability Gramian, respectively. Further, the solutions to (3.2) and (3.3) can be written as the quadratic forms

$$L_c(x_0, t) = \frac{1}{2} x_0^T P^{-1}(t) x_0, \qquad L_o(x_0, t) = \frac{1}{2} x_0^T Q(t) x_0$$

The Gramians P and Q and their analogues for other system classes, are central to many model reduction methods. They show how strongly states are connected to the input and output and thereby supplies essential information of which state subspace is of most significance.

Method description

Let the system to be reduced have the form

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= g(x, u) \end{aligned} \tag{3.4}$$

where $u \in \mathbf{R}^l$, $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^m$. Linearizations of the system dynamics will be used to find states that are redundant or that have small importance for the input-output relationship. Local importance of states would be revealed if one linearizes the system around a stationary point. A combination of several linearization points could then indicate which the important states are in the nonlinear system. However, some states may only have an active role during transient behaviour, which will later be commented in Example 3.4. Instead, linearization around a trajectory will be used as a tool to find an approximate low-order model.

Recall the theory concerning linear time-varying systems presented in the prior section. The time-varying Gramians give information about state importance even in transient regions of state space. These Gramians can be computed in the neighborhood of simulated trajectories using linearization of the system dynamics. When there exists a linear coordinate transformation that disconnects some states from the input-output relationship, this will be revealed in those localized Gramians.

The choice of training trajectory, around which linearization is made, is an important aspect of the reduction procedure. The corresponding training input should be chosen as a typical input signal, which is rich enough to excite all dynamics important to the intended model use.

A possible scenario is that a state is nearly constant but non-zero. It could then potentially be replaced by a constant value without loosing much accuracy. However, the method yields a linear coordinate change (not an affine one), which is followed by truncation of states. Therefore, a preconditioning coordinate change should be applied to (3.4) that shifts the states so that their mean value over the training trajectory is zero. Also, if the model is equipped with multiple input and output signals they should be scaled so that they posses the same amplitude. This scaling also leaves room for the user to specify how important he/she finds the accuracy of the different input and output signals.

Another property that should be considered is the fact that an observable but not controllable state can not be removed in general, which is demonstrated in Example 3.3. When this is suspected a possible remedy is to make these states controllable by introducing additional input signals. The method would then become aware of the state component's importance even though these extra signals would never be used.

The following sections explain the main steps involved in the method.

Linearization along trajectory The first step is to choose the so called training input signal. This is an important step of the method on which the performance is highly dependent. As a general rule the input should be chosen to obey physical restrictions on the signal and to excite all relevant dynamics. To find such a signal might be a challenging task. However, one could also see this as an advantage of the method. If the reduced model is only going to be used for some restricted purposes, the model could probably be reduced to a greater extent. Through the choice of training input the user can show which behaviour is relevant for the reduced system to reproduce. Hence, the signal should be chosen corresponding to realistic usage of the model.

With the training input signal chosen, the system is simulated over the time interval $t = [0, t_f]$. The system is then linearized along the state trajectory the training input gave rise to. The result is a time-varying linear system

$$\begin{aligned} \Delta\dot{x}(t) &= A(t)\Delta x(t) + B(t)\Delta u(t) \\ \Delta y(t) &= C(t)\Delta x(t) + D(t)\Delta u(t) \end{aligned} \qquad t \in [0, t_f]$$

where Δu, Δx and Δy denote deviations from the nominal trajectories. Further, A, B, C and D are time-varying matrices defined by

$$\begin{aligned} A(t) &= \frac{\partial f}{\partial x}(x(t), u(t)) & \qquad B(t) &= \frac{\partial f}{\partial u}(x(t), u(t)) \\ C(t) &= \frac{\partial g}{\partial x}(x(t), u(t)) & \qquad D(t) &= \frac{\partial g}{\partial u}(x(t), u(t)) \end{aligned}$$

Compute the time-varying Gramians Similar to balanced truncation, the method uses the notion of Gramians. As mentioned, for the time-varying systems the controllability Gramian can be computed through simulation of the differential equation

$$\dot{P}(t) = A(t)P(t) + P(t)A^T(t) + B(t)B^T(t) \tag{3.5}$$

with $P(0) = 0$. Similarly, the observability Gramian is determined by

$$\dot{Q}(t) = -Q(t)A(t) - A^T(t)Q(t) - C^T(t)C(t) \tag{3.6}$$

with the boundary condition $Q(t_f) = 0$. The controllability Gramian $P(t)$ reveals how large deviation in input signal is needed to perturb $x(t)$. If a certain state component is hard to perturb for all times, one can suspect that this state is in general hard to affect in the nonlinear system. Similarly, $Q(t)$ shows how much the output signal is affected if $x(t)$ is perturbed. If the output signal is weakly influenced by a certain state perturbation, independently of when the perturbation is made, it can be suspected that this state-output connection is weak also for the nonlinear system.

Determine the average Gramians As mentioned, the Gramians $P(t)$ and $Q(t)$ contain local information along the trajectory of how strongly states are connected to the input and output. In order to remove the time dependency and isolate the overall important states with a constant state transformation, one could use the *average Gramians*

$$\bar{P} = \frac{1}{t_f} \int_0^{t_f} P(\tau) d\tau \qquad\qquad \bar{Q} = \frac{1}{t_f} \int_0^{t_f} Q(\tau) d\tau \tag{3.7}$$

There are several alternative ways of going from the time-varying Gramians $P(t)$ and $Q(t)$ to time-invariant representative matrices. This particular choice is motivated by its simplicity and properties such as if a state is connected to the input or output somewhere along the trajectory it will be shown in the average Gramians. These time-invariant matrices contain information of how strongly the states are connected to the input and output on average over the training trajectory. For example, if a certain linear state combination is unobservable from the output in all points of the trajectory, it will be revealed in $\bar{Q}$. Further, a rank deficiency of the matrix $\bar{P}\bar{Q}$ indicates that some states are obsolete and can be truncated from the model without changing the input-output relationship.

Find balancing coordinate change This step is performed to extract the relevant state subspace using the information gathered in the average Gramians. The chosen approach treats $\bar{P}$ and $\bar{Q}$ as if they belonged to a linear time-invariant system. By following the standard balanced truncation procedure for linear time-invariant systems, a coordinate change $z = Tx$ can be found, see [Zhou and Doyle, 1998], such that the average Gramians become equal and diagonal with decreasing diagonal elements.

$$T\bar{P}T^T = T^{-T}\bar{Q}T^{-1} = \bar{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}$$

The diagonal elements $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$ correspond to the Hankel singular values in balanced truncation of linear systems, where they show how important states are for the input-output relationship. In contrast to the linear case, no error bound is available but these values will be used as an indication to which model order to choose for the reduced system.

Truncate states Truncating states corresponding to relatively small singular values and keeping $\hat{n}$ states is equivalent to removing rows and columns in T and T^{-1}, respectively. T_l is the top $\hat{n}$ rows of T and T_r is the $\hat{n}$ leftmost columns of T^{-1}.

$$\begin{aligned} T \in \mathbf{R}^{n\times n} \quad &\Rightarrow \quad T_l \in \mathbf{R}^{\hat{n}\times n} \\ T^{-1} \in \mathbf{R}^{n\times n} \quad &\Rightarrow \quad T_r \in \mathbf{R}^{n\times \hat{n}} \end{aligned} \tag{3.8}$$

Applying the truncated coordinate change to the original system formulation in (3.4) gives rise to the reduced order system

$$\begin{aligned} \dot{\hat{z}} &= T_l f(T_r \hat{z}, u) \\ \hat{y} &= g(T_r \hat{z}, u) \end{aligned} \tag{3.9}$$

where $\hat{z} \in \mathbf{R}^{\hat{n}}$. The method is summarized in Algorithm 3.1.

Method properties

Intuitively, the proposed method should perform well when the time-varying Gramians do not change too much over the trajectory. The average Gramians are then better representatives for the local properties.

In this thesis a single training input signal is used but one could consider basing the reduction on several trajectories. Distinct average Gramians $\bar{P}_i$ and $\bar{Q}_i$ could be determined for each input independently. A pair of total average Gramians could then be obtained through a weighted sum where the weights depend on how important each input scenario is.

$$\bar{P} = \sum_i w_i \bar{P}_i \qquad\qquad \bar{Q} = \sum_i w_i \bar{Q}_i$$

The balancing coordinate change is then determined based on these total average Gramians.

Depending on the model, deriving the reduced system through symbolic substitution in (3.9) may be an unattractive option. Commonly, the original set of equations is sparse, i.e. all state equations do not involve all states. The sparsity is lost with a dense coordinate change and truncation of states. Therefore, the total computation time is not necessarily reduced for the right-hand-side functions, which can be seen e.g. in [Liu and Wagner, 2002]. One possibility to redeem this is to extend the presented method with a piece-wise approximation of f and g, as in [Vasilyev *et al.*, 2006]. Additionally, for continuous-time systems not only evaluation time of the right-hand-side functions are of importance. Integration time does also depend on the choice of solver, numerical stiffness etc. These properties have not been considered in this work.

Algorithm 3.1: The average Gramian method, continuous time

1. Choose training input and simulate the system

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

   over $[0,t_f]$.

2. Linearize around the training trajectory to obtain $A(t)$, $B(t)$, $C(t)$.

$$A(t) = \frac{\partial f}{\partial x}(x(t),u(t)), \quad B(t) = \frac{\partial f}{\partial u}(x(t),u(t)), \quad C(t) = \frac{\partial g}{\partial x}(x(t),u(t))$$

3. Calculate time-varying Gramians

$$\begin{aligned} \dot{P}(t) &= A(t)P(t) + P(t)A^T(t) + B(t)B^T(t) & P(0) &= 0 \\ \dot{Q}(t) &= -Q(t)A(t) - A^T(t)Q(t) - C^T(t)C(t) & Q(t_f) &= 0 \end{aligned}$$

4. Determine the average Gramians

$$\bar{P} = \frac{1}{t_f}\int_0^{t_f} P(\tau)d\tau \qquad \bar{Q} = \frac{1}{t_f}\int_0^{t_f} Q(\tau)d\tau$$

5. Apply the standard balanced truncation method on $\bar{P}$ and $\bar{Q}$, which yields a balancing coordinate change $z = Tx$ and the singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$. See [Zhou and Doyle, 1998].

6. Determine the reduced model order $\hat{n}$ from the relative size of the singular values and truncate the coordinate change. T_l is the top $\hat{n}$ rows of T and T_r is the $\hat{n}$ leftmost columns of T^{-1}.

$$\begin{aligned} T \in \mathbf{R}^{n\times n} \quad &\Rightarrow \quad T_l \in \mathbf{R}^{\hat{n}\times n} \\ T^{-1} \in \mathbf{R}^{n\times n} \quad &\Rightarrow \quad T_r \in \mathbf{R}^{n\times\hat{n}} \end{aligned}$$

7. Apply the truncated coordinate change to the original system

$$\begin{aligned} \dot{\hat{z}} &= T_l f(T_r\hat{z},u) \\ \hat{y} &= g(T_r\hat{z},u) \end{aligned}$$

Comparison with other methods The proposed method, the empirical Gramian approach [Lall *et al.*, 2002] , the proper orthogonal decomposition method [Lumley, 1967] and the trajectory piece-wise linear approach [Vasilyev *et al.*, 2006] all use a linear coordinate change combined with truncation of states. The difference is in how the coordinate change is found. What makes this method different from others is mainly the use of linearization along trajectories and the average Gramians.

In the trajectory piece-wise linear approach, the linear coordinate change is typically determined from one linearization around the initial state. A more elaborate procedure is suggested in [Vasilyev *et al.*, 2003] and [Rewieński, 2003], where several local coordinate changes are independently determined from linearizations distributed over a trajectory. A single global coordinate change is then obtained through an aggregation and biorthonormalization procedure. However, using this method, a state that is locally controllable in some regions and observable in others but never both simultaneously will be neglected. In the proposed method, the time-varying Gramians and their averaged value will show the importance of such states. On the other hand, this property comes with the numerical expense of solving (3.5) and (3.6).

The empirical Gramian approach and the proper orthogonal decomposition mehthod use simulation data to produce the analogues of the average Gramians. This is in contrast to the proposed method, which uses linearizations.

Parallels can also be seen with the nonlinear balancing performed in [Scherpen and Fujimoto, 2003], such as the use of energy functions. The restriction to a linear coordinate change is of course a coarse approximation, which, on the other hand, greatly facilitates computation.

For further illustration, the method is demonstrated in the following examples.

Examples

EXAMPLE 3.1—EXACT REDUCTION

Just as a demonstration, the method will here be applied to the following toy example. The nonlinear system

$$\begin{aligned} \dot{x}_1 &= -3x_1^3 + x_1^2 x_2 + 2x_1 x_2^2 - x_2^3 \\ \dot{x}_2 &= 2x_1^3 - 10x_1^2 x_2 + 10x_1 x_2^2 - 3x_2^3 - u \\ y &= 2x_1 - x_2 \end{aligned} \tag{3.10}$$

has exactly the same input-output relationship as the system

$$\dot{y} = -y^3 + u. \tag{3.11}$$

It is a challenge for any model-reduction procedure to detect that a reduction like this is possible. A general methodology for such problems has been presented, but first a simple proof of the equivalence in this particular case is given.

Note that the system can be rewritten as

$$\begin{aligned}
\dot{x}_1 &= -(2x_1 - x_2)^2 x_1 + (x_1 - x_2)^3 \\
\dot{x}_2 &= -(2x_1 - x_2)^2 x_2 + 2(x_1 - x_2)^3 - u \\
y &= 2x_1 - x_2
\end{aligned}$$

With the new variables $z_1 = 2x_1 - x_2$, $z_2 = x_2 - x_1$, this means that

$$\begin{aligned}
\dot{z}_1 &= -z_1^3 + u \\
\dot{z}_2 &= -z_1^2 z_2 - z_2^3 - u \\
y &= z_1
\end{aligned}$$

In particular, the state z_2 does not appear in the output and does not affect z_1. Hence, it can be truncated and (3.11) holds.

In the example, a linear coordinate transformation followed by state truncation gave a simplified model without approximation error. The goal of the described method is to provide a systematic way to find such transformations whenever they exist and otherwise to find good approximations.

Now the method will be applied to the system description in (3.10). Simulating the system along various trajectories and computing the observability Gramian according to the differential equation

$$\dot{Q}(t) = -Q(t)A(t) - A^T(t)Q(t) - C^T(t)C(t)$$

with $Q(t_f) = 0$ one observes that the rank of $Q(t)$ never exceeds one for any trajectory. In this case, $\bar{Q}$ and $\bar{P}\bar{Q}$ are singular and one state can be truncated without affecting the input-output relationship. For demonstration, the nonlinear system was simulated with a certain input signal. Following the method the average Gramians were determined to

$$\bar{P} = \begin{bmatrix} 0.0018 & -0.0145 \\ -0.0145 & 0.2936 \end{bmatrix} \qquad \bar{Q} = \begin{bmatrix} 1.3058 & -0.6529 \\ -0.6529 & 0.3264 \end{bmatrix}$$

and the matrix $\bar{Q}$ is, as expected, singular[1]. The corresponding coordinate change $z = Tx$ is then determined according to the standard balanced

[1]using a larger numerical precision of $\bar{P}$ and $\bar{Q}$ than printed here

truncation method. Further, the Hankel singular values become in this case $\sigma_1 = 0.3422$ and σ_2 is very close to zero (not exactly, due to numerical reasons). In accordance with the size of σ_2, z_2 is truncated and substitution according to (3.9) yields the nonlinear system

$$\begin{aligned}\dot{z}_1 &= -1.23z_1^3 - 0.901u\\ y &= -1.11z_1\end{aligned}$$

which is equivalent to $\dot{y} = -y^3 + u$. □

EXAMPLE 3.2—A SEVEN-STATE SYSTEM

The procedure can be applied to larger examples and also when loss-less truncation is not possible. Consider the seven-state system

$$\begin{aligned}\dot{x}_1 &= -x_1^3 + u\\ \dot{x}_2 &= -x_2^3 - x_1^2x_2 + 3x_1x_2^2 - u\\ \dot{x}_3 &= -x_3^3 + x_5 + u\\ \dot{x}_4 &= -x_4^3 + x_1 - x_2 + x_3 + 2u\\ \dot{x}_5 &= x_1x_2x_3 - x_5^3 + u\\ \dot{x}_6 &= x_5 - x_6^3 - x_5^3 + 2u\\ \dot{x}_7 &= -2x_6^3 + 2x_5 - x_7 - x_5^3 + 4u\\ y &= x_1 - x_2^2 + x_3 + x_4x_3 + x_5 - 2x_6 + 2x_7\end{aligned}$$

Following the described procedure, the system is linearized along a simulated training trajectory. As mentioned, the training input signal should be chosen to reflect intended model use. However, this system lacks physical interpretation and just as an example the input signal is chosen as a 10Hz square-wave signal with amplitude one. Again, following the procedure, the Gramians are calculated according to (3.5) and (3.6). Further, the balancing coordinate change T is computed and the Hankel singular values are shown in Figure 3.2. The relative size of these values indicates the importance of the new states for the input-output relationship.

If, for example, the nonlinear system is truncated to one state the reduced system becomes

$$\begin{aligned}\dot{z}_1 &= -0.492z_1 - 0.0879z_1^3 + 5.08u\\ \tilde{y} &= 1.34z_1 + 0.0792z_1^2\end{aligned}$$

A comparison achieved by simulating the original and reduced system with the same input signal $u(t)$ can be seen in Figure 3.3. The input signal is different from the one used for model reduction and has been chosen to be the sum of a sinusoidal and a square-wave signal. □

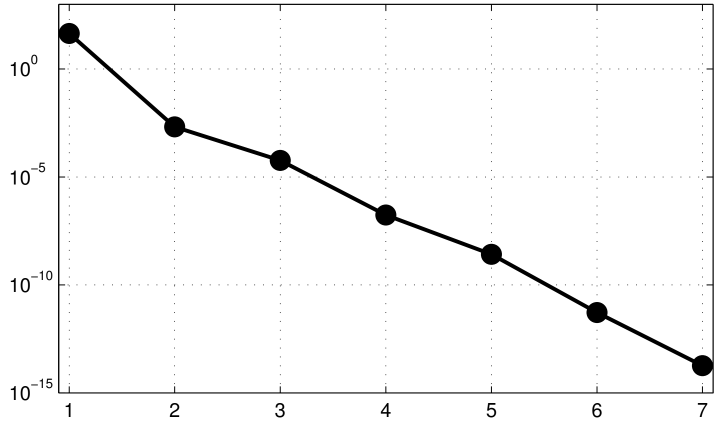

Figure 3.2 Hankel singular values for the system in Example 3.2

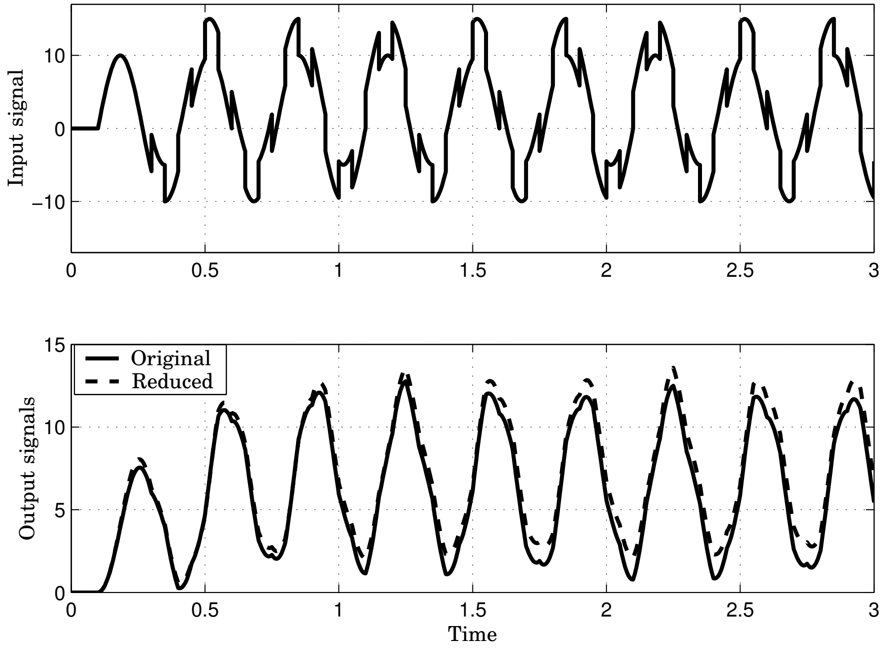

Figure 3.3 Validation results for the system in Example 3.2

EXAMPLE 3.3—TIME DEPENDENCY
Consider the time-varying nonlinear system

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

In order to make the system class match the one of (3.4) one can extend the state vector. Introducing $x_e = [x^T t]^T$ and $\dot{t} = 1$, the system can be written in the form

$$\begin{aligned} \dot{x}_e &= f_e(x_e,u) \\ y &= g_e(x_e,u) \end{aligned}$$

Applying the average Gramian method on this system will show that time is not controllable and can therefore be removed. However, the time dependency could be strong and removing this state component might not be advisable.

This example demonstrates a case when the method produces undesirable results. As mentioned, if uncontrollable states are suspected to be important, additional input signals can be introduced, making them controllable. In this case one could introduce $\dot{t} = 1 + \beta u_t$ and letting $u_t = 0$ both in the training input and in the reduced model. The gain β is chosen large enough to ensure that the additional input signal is not negligible compared to the others, from a controllability point of view.

□

EXAMPLE 3.4—A MASS-SPRING-DAMPER SYSTEM
In this example, the method is applied to a two-dimensional multiple-input multiple-output mass-spring-damper system. Figure 3.4 shows six masses connected with springs and dampers. The input signal is external force, in horizontal and vertical directions, on the leftmost mass. The output signal is the position coordinates of the top middle mass.

A thin line in the figure represents a linear spring-damper with an unforced length l_0 according to the figure. The masses, except the two rightmost ones, are also connected to the ground with linear spring-dampers.

The motion equations for each mass consist of four differential equations

$$\begin{aligned} \dot{p}_x &= v_x & \qquad \dot{p}_y &= v_y \\ \dot{v}_x &= \frac{1}{M}\sum_i F_{x,i} & \qquad \dot{v}_y &= \frac{1}{M}\sum_i F_{y,i} \end{aligned}$$

where p_x and p_y are the position coordinates with the corresponding velocities v_x and v_y. The mass is denoted M and the forces $F_{x,i}$ and $F_{y,i}$ are

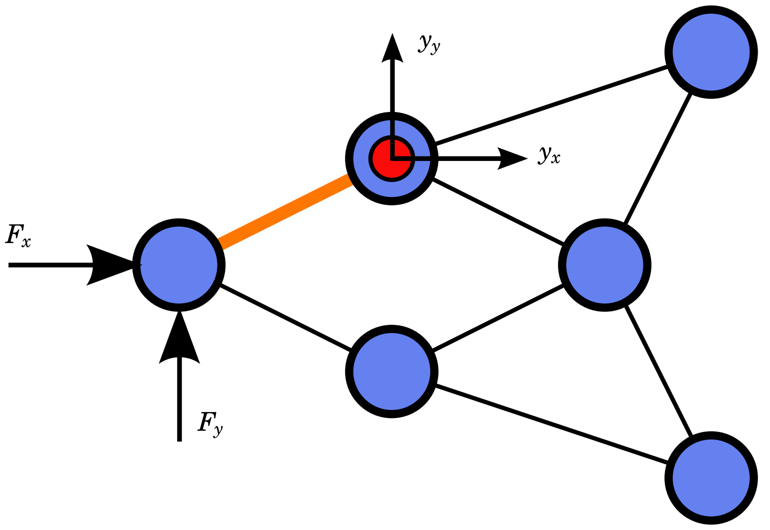

Figure 3.4 Mass-spring-damper system in Example 3.4. Forces on the left mass as input signals and position of the marked mass as output.

the forces in horizontal and vertical directions inflicted by spring-damper i

$$F_{x,i} = \left(K(l_i - l_{0i}) + D\frac{d}{dt}(l_i) \right) \cos \theta_i$$
$$F_{y,i} = \left(K(l_i - l_{0i}) + D\frac{d}{dt}(l_i) \right) \sin \theta_i$$

Here l_i is the length of spring-damper i, D the damping coefficient and K the spring coefficient. In this example all coefficients have been set to one, $M = K = D = 1$. Further, the angle θ_i is the angle of the spring-damper. Here, only small angle perturbations are considered and θ_i is therefore assumed to be constant.

The thick line is a nonlinear damper that gives a force proportional to the deformation rate to the power of three,

$$F_{x,i} = D\left(\frac{d}{dt}(l_i)\right)^3 \cos \theta_i \qquad F_{y,i} = D\left(\frac{d}{dt}(l_i)\right)^3 \sin \theta_i$$

Linearization of the model around any stationary point would neglect this nonlinear damper, it only affects the linearization during transient

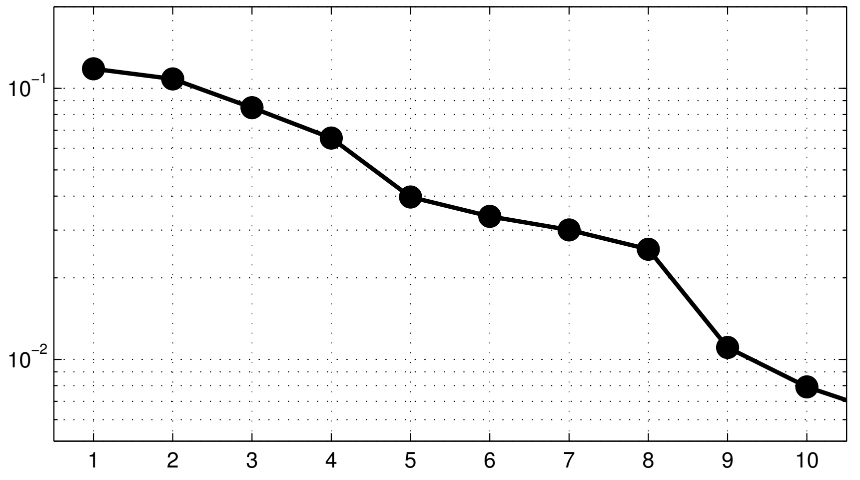

Figure 3.5 The 10 largest Hankel singular values for the mechanical system in Example 3.4.

behaviour. In the case of the left-most mass, the external forces also contribute to the equations.

The model has four states per mass, yielding a total of 24 states, and can be written on the form

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{aligned}$$

In this example the described method is compared to the Proper Orthogonal Decomposition method as described in Section 1.2.

Reduction to 8 states is performed with both methods using the same training trajectory. The resulting singular values, corresponding to the average Gramians, are shown in Figure 3.5. A simulation result can be seen in Figure 3.6 where the input is different from the training input. A qualitatively better result is obtained with the described method, which partly is due to the fact that the Proper Orthogonal Decomposition method does not take the output function $g(x,u)$ into consideration. It should also be mentioned that the POD method performed the reduction with much less numerical effort.

□

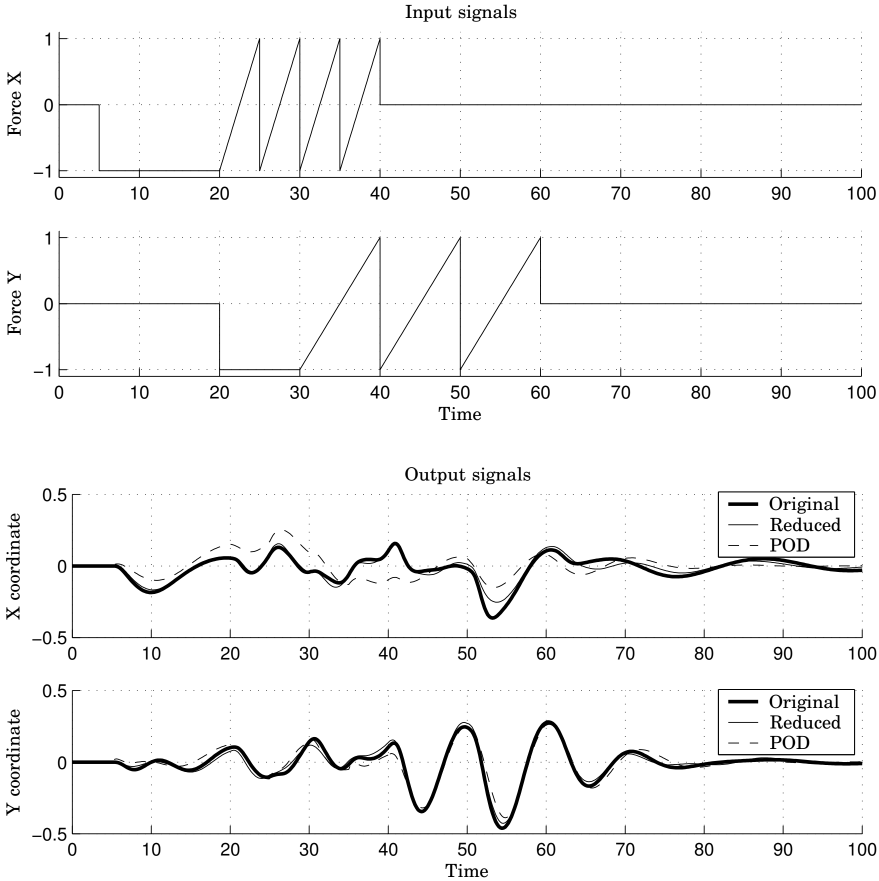

Figure 3.6 Simulation results for Example 3.4. Method comparison for model reduction from 24 to 8 states.

EXAMPLE 3.5—HEAT EXCHANGE REACTOR

Here a model of a novel heat exchange reactor is considered, see Figure 3.7. It is a tubular reactor and a simplified description is that flows of reactants A and B are injected in one end, inside the reactor they form the product C and heat.

Modeling heat, flow and chemical reactions of the reactor gives rise to nonlinear partial differential equations. Spatial discretization of these equations yields a set of ordinary differential equations, as in (3.4), where states correspond to temperatures and concentrations at different locations in the reactor. The model to be treated here is such a discretization

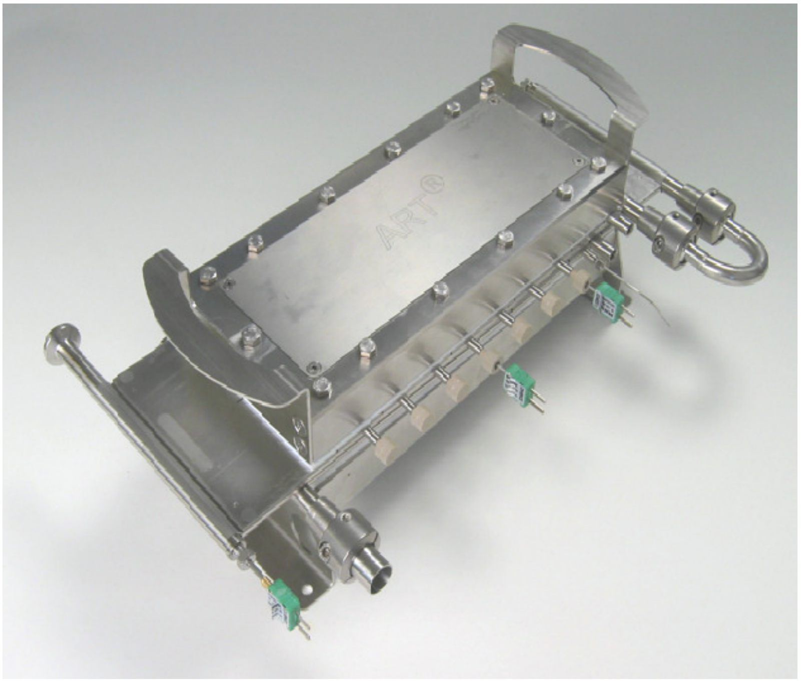

Figure 3.7 A lab-scale version of the open plate reactor treated in Example 3.5. Courtesy of Alfa Laval AB.

yielding a total of 52 states with flow rate of Reactant B as input signal and the outlet temperature as output signal. Other variables, such as reactant A or cooling, are held constant. The model is defined in the Modelica language, see [Fritzson, 2004], and managed with the modeling and simulation tool Dymola, see [Dynasim AB, 2006]. For more information about the reactor and the model see [Haugwitz *et al.*, 2007] and [Haugwitz, 2007].

It would be possible to do symbolic substitution when introducing the coordinate change in the Modelica based model. However, in this case the piece-wise linear approximation proposed in [Vasilyev *et al.*, 2006] will be used to approximate the right-hand-side functions. The idea is to approximate the functions with a weighted sum of linearizations. The algorithm starts with using a linearization at the start time as an approximant of the functions f and g. Following the training trajectory the algorithm then adds a linearization to the collection when the approximation error supersedes a certain threshold. For more information about this algorithm

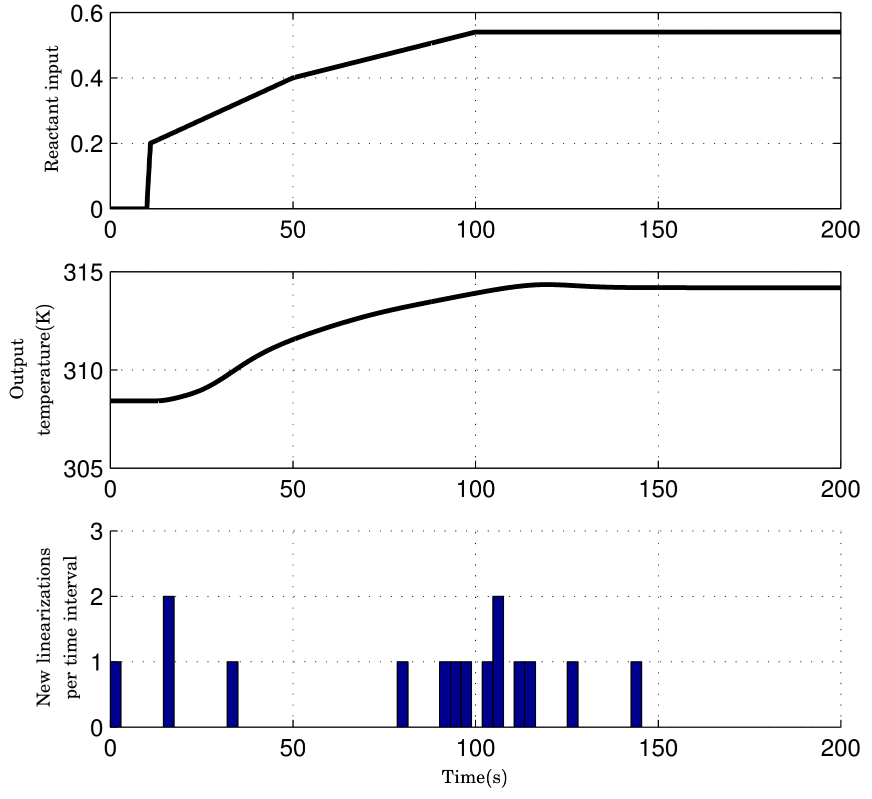

Figure 3.8 Reactor start-up scenario used as training trajectory in Example 3.5.

see Algorithm 1.1 in Section 1.2 or [Rewieński, 2003].

The top plot in Figure 3.8 shows the training input signal. It is a reactor start-up scenario where the flow of reactant B is increased from zero. Further, the second plot shows the implied increase in outlet temperature. The bottom plot shows how the linearizations are aggregated along the trajectory. In total the approximant consists of 15 linearizations.

Figure 3.9 shows the largest 30 of the 52 singular values. As can be seen, it is not clear what an appropriate reduced order could be. Through trial and error a reduced order of 8 states is found sufficient.

The reactant input shown in the top plot in Figure 3.10 is used as a validation test. The bottom plot shows the outlet temperature provided by the original and reduced model.

$\square$

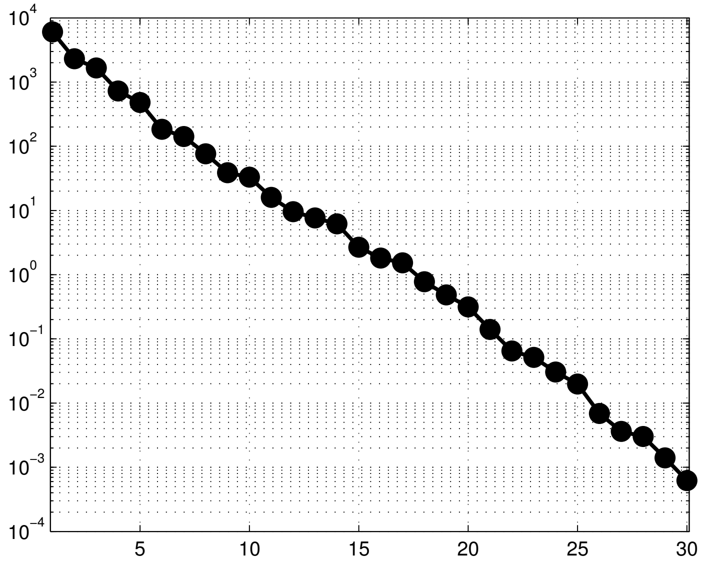

Figure 3.9 The largest 30 of the 52 singular values in Example 3.5.

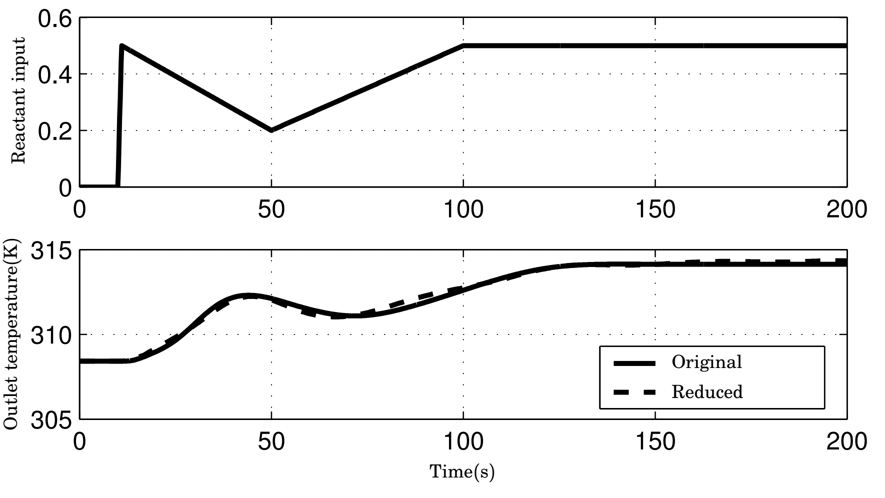

Figure 3.10 Validation results for Example 3.5. The model is reduced from 52 to 8 states.

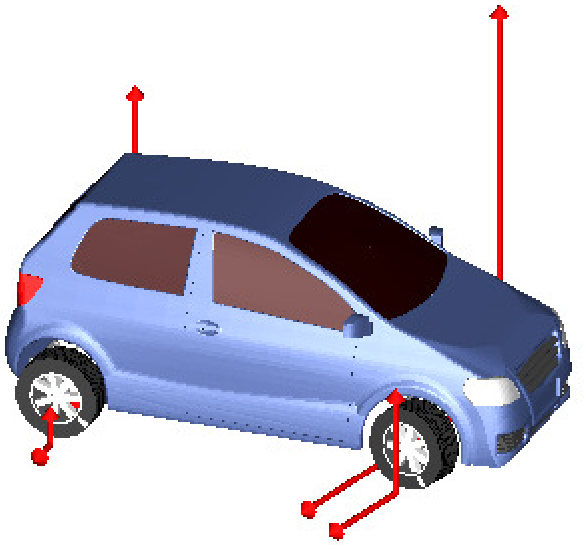

Figure 3.11 Visualization of a turning maneuver with the VehicleDynamics Library [Modelon, 2007] used in Example 3.6.

EXAMPLE 3.6—A VEHICLE DYNAMICS MODEL
In this example a model from the VehicleDynamics Library [Modelon, 2007] will be treated. Also in this case the model is described in the Modelica language. The model is fairly detailed and contains 41 states distributed over the car body, the suspension and the wheels. Figure 3.11 shows a 3D visualisation of the car together with the tire contact forced during cornering. The chosen input-output pair is the steer wheel angle as input signal and the car yaw rate as output signal.

As a training trajectory a double lane change maneuver shown in Figure 3.12 is used. As in the previous example, the right-hand-side functions are difficult to access and therefore the piece-wise linear approximation is applied. The total number of linearizations for this training trajectory is 33.

Following the procedure the singular values were calculated and are shown in Figure 3.13. Here the reduced model order is chosen to 8, i.e., a reduction from 41 to 8 states.

The validation scenario is shown in Figure 3.14 where the reduced model shows good performance except for the unmatched initial condition. A zoom-in of the same validation test can be seen in Figure 3.15.

□

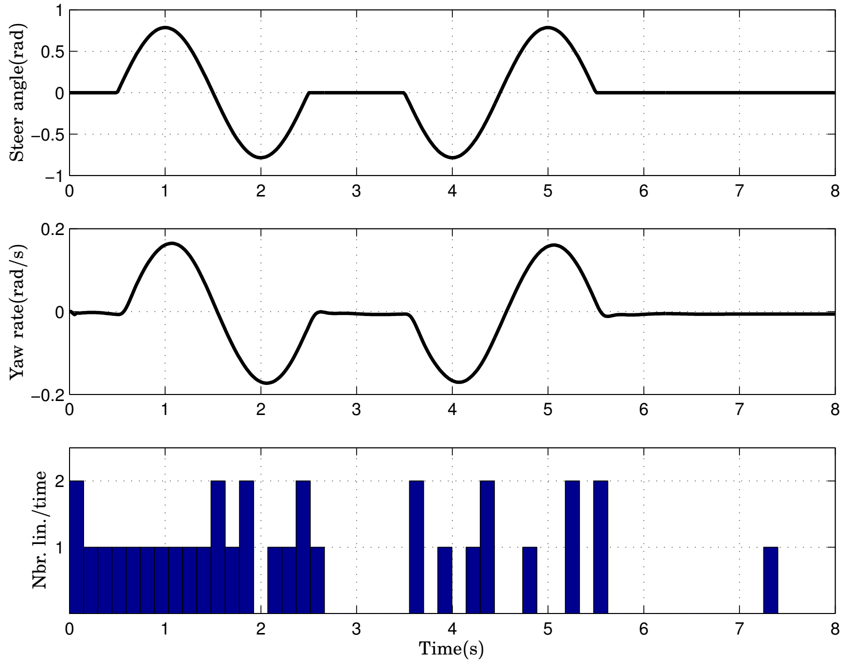

Figure 3.12 Cornering maneuver used as training trajectory in Example 3.6. In total the piece-wise linear approximation consists of 33 linearizations.

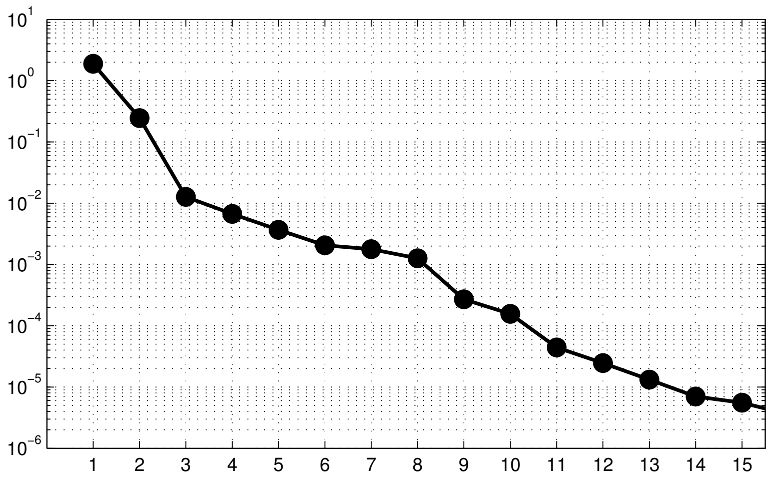

Figure 3.13 The 15 largest singular values for Example 3.6.

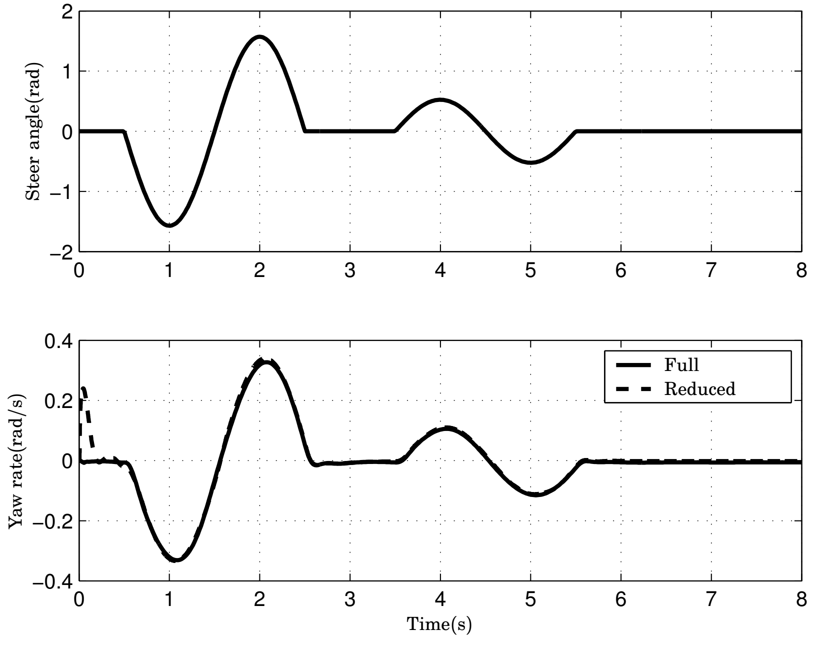

Figure 3.14 Validation results for Example 3.6. The model is reduced from 41 to 8 states.

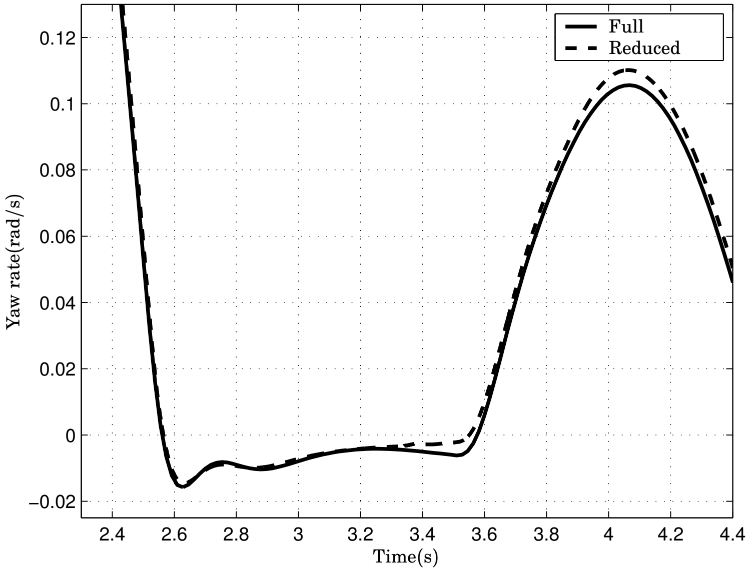

Figure 3.15 Zoom-in on the validation results in Figure 3.14

3.3 The discrete-time case

The discrete-time version of the method has many things in common with continuous-time case. The largest difference is that simulation time is directly proportional to the evaluation time of the right-hand-side functions, in contrast to the continuous-time case where numerical stiffness and integration step size also are important. Additionally, method steps such as linearization and computation of Gramians are numerically easier to handle.

Preliminaries

Also in the discrete-time case the method is based on theory concerning linear time-varying systems and the theory in Section 1.1 is revisited. The counterpart to (3.1) is the linear discrete-time time-varying system

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k \\ y_k &= C_k x_k + D_k u_k \end{aligned} \quad k \in [1, N],$$

where x_k is the state vector, u_k the input signal and y_k the output signal at time k. Further, A_k, B_k, C_k and D_k are time varying matrices of appropriate dimensions. Here the sub-index denotes time.

Again the notion of energy functions will be used. Here the *controllability energy function* becomes the optimal control problem

$$L_c(x^*, t) = \min_{\substack{u \in L_2(0,t) \\ x_1 = 0 \\ x_t = x^*}} \frac{1}{2} \sum_{k=1}^{t} ||u_k||^2. \tag{3.12}$$

That is, $L_c(x^*, t)$ is the minimal amount of energy in u required to reach a certain state x^* at time t, starting from the zero initial state.

Similarly, the *observability energy function* can in this case be stated as

$$L_o(x^*, t) = \frac{1}{2} \sum_{k=t}^{N} ||y_k||^2, \quad x_t = x^*, \quad u \equiv 0. \tag{3.13}$$

That is, the amount of energy an initial state x^* at time t induces in the output signal over the time interval $[t, N]$. The concept of these energy functions is illustrated in Figure 3.16.

Similar to the continuous case, the energy functions can be determined through the quadratic forms

$$L_c(x^*, t) = \frac{1}{2} x^{*T} P_t^{-1} x^* \qquad\qquad L_o(x^*, t) = \frac{1}{2} x^{*T} Q_t x^*.$$

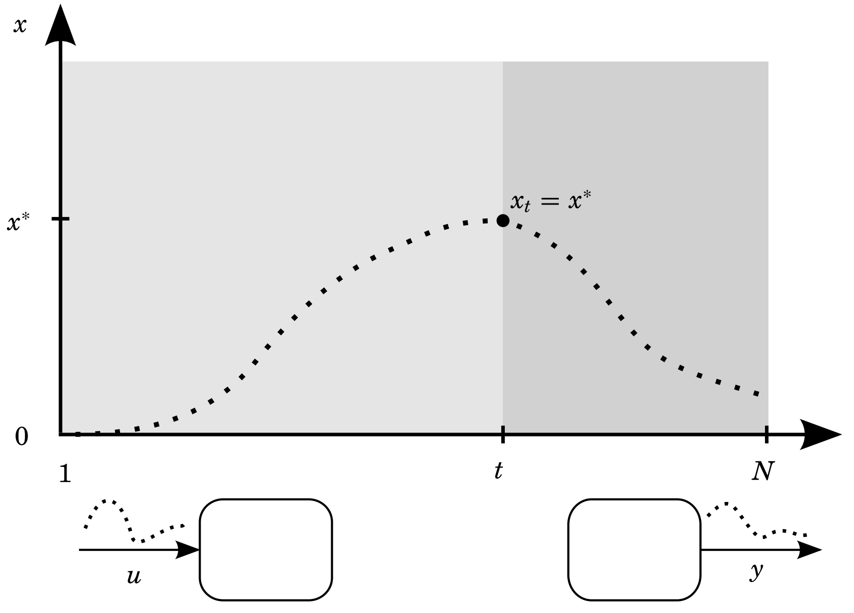

Figure 3.16 Visualization of the energy functions. The left part illustrates the minimal input energy required to reach x^* at time t. In the right part the control signal is zero and the initial state x^* yields the mentioned output energy.

where the controllability Gramian P_k and observability Gramian Q_k are given by the Lyapunov equations

$$\begin{aligned} P_{k+1} &= A_k P_k A_k^T + B_k B_k^T, \qquad && k \in [1, N] \\ Q_k &= A_k^T Q_{k+1} A_k + C_k^T C_k, \qquad && k \in [1, N] \end{aligned}$$

with the boundary conditions $P_1 = 0$ and $Q_{N+1} = 0$.

Method description

The discrete-time analogue of the continuous-time system class is the general nonlinear system

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ y_k &= g(x_k, u_k) \end{aligned} \tag{3.14}$$

where $u_k \in \mathbf{R}^l$, $x_k \in \mathbf{R}^n$ and $y_k \in \mathbf{R}^m$.

The following sections briefly explain the main steps involved in the method. Due to the strong similarity to the continuous case, the considerations in Section 3.2 also apply here.

Linearization along trajectory After choosing a training input signal, the system is linearized along the state trajectory the training input gave rise to. The result is the time-varying linear system

$$\begin{aligned} \Delta x_{k+1} &= A_k \Delta x_k + B_k \Delta u_k \\ \Delta y_k &= C_k \Delta x_k + D_k \Delta u_k \end{aligned} \quad k \in [1, N]$$

where Δu, Δx and Δy denote deviations from the nominal trajectories. Further, A_k, B_k, C_k and D_k are time-varying matrices of appropriate size that vary with k.

Compute the time-varying Gramians Similar to balanced truncation the method uses the notion of Gramians. As mentioned, for time-varying systems the controllability Gramian can be computed according to the difference equation

$$P_{k+1} = A_k P_k A_k^T + B_k B_k^T, \quad k \in [1, N] \tag{3.15}$$

with $P_1 = 0$. Similarly, the observability Gramian is determined by

$$Q_k = A_k^T Q_{k+1} A_k + C_k^T C_k, \quad k \in [1, N] \tag{3.16}$$

with the boundary condition $Q_{N+1} = 0$.

Determine the average Gramians The average Gramians are as in the continuous case defined as the average value over the time interval

$$\bar{P} = \frac{1}{N} \sum_{k=1}^{N} P_k \qquad \bar{Q} = \frac{1}{N} \sum_{k=1}^{N} Q_k. \tag{3.17}$$

Find balancing coordinate change Again the average Gramians $\bar{P}$ and $\bar{Q}$ are treated as if they belonged to a linear time-invariant system. By following the standard balanced truncation procedure for linear systems, a coordinate change is found such that the average Gramians become equal and diagonal with decreasing diagonal elements.

$$T\bar{P}T^T = T^{-T}\bar{Q}T^{-1} = \bar{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \tag{3.18}$$

The diagonal elements $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$ corresponds to the Hankel singular values in balanced truncation of linear systems, where they show how important states are for the input-output relationship. Also in this case no error bound is available but these values will be used to determine which model order to choose for the reduced system.

Truncate states Truncation of states is performed in the same way as for the continuous case. Keeping $\hat{n}$ states corresponding to relatively large singular values is equivalent to removing rows and columns in T and T^{-1}, respectively.

$$\begin{aligned} T \in \mathbf{R}^{n\times n} \quad &\Rightarrow \quad T_l \in \mathbf{R}^{\hat{n}\times n} \\ T^{-1} \in \mathbf{R}^{n\times n} \quad &\Rightarrow \quad T_r \in \mathbf{R}^{n\times \hat{n}} \end{aligned} \tag{3.19}$$

Applying the truncated coordinate change to the original system formulation in (3.14) gives rise to the reduced order system

$$\begin{aligned} \hat{z}_{k+1} &= T_l f(T_r \hat{z}_k, u_k) \\ y_k &= g(T_r \hat{z}_k, u_k) \end{aligned} \tag{3.20}$$

where $\hat{z} \in \mathbf{R}^{\hat{n}}$. How to derive analytical expressions for the reduced system in (3.20) is highly dependent on the format the original model is implemented in. This matter will be further discussed in Example 3.7. The method is summarized in Algorithm 3.2.

Examples

The method will here be used on a real-world automotive application to demonstrate its applicability. The automotive industry is experiencing tightening emission legislations together with high demands on performance and driveability. As a counteraction, controller software tends to become more and more complex. However, intricate controller software has several downsides. The large number of controller parameters yields an exhaustive calibration task, often performed through costly experiments. In addition, to guarantee reliability, validation and verification analysis is performed on the controller in combination with the engine. This task would also greatly benefit from a less complex controller structure.

EXAMPLE 3.7—AUTOMOTIVE CONTROLLER SOFTWARE
Here the method is applied to an engine controller used in current production cars. The result is a nonlinear piece-wise affine system with improved simulation speed.

Model description The model in this example is closely related to the simulation model treated in Chapter 2. It consists of software used for online air-path dynamics estimation in current production cars. In particular, the model estimates the air charge in a spark ignition engine, i.e., the amount of air the cylinder is loaded with when the inlet valve closes. The amount of fuel to inject is then determined from this value

Algorithm 3.2: The average Gramian method, discrete time

1. Choose training input and simulate the system

$$
\begin{aligned}
x_{k+1} &= f(x_k, u_k) \\
y_k &= g(x_k, u_k)
\end{aligned}
$$

   over $k \in [1, N]$.

2. Linearize around the training trajectory to obtain A_k, B_k, C_k.

$$
A_k = \frac{\partial f}{\partial x}(x_k, u_k), \qquad B_k = \frac{\partial f}{\partial u}(x_k, u_k), \qquad C_k = \frac{\partial g}{\partial x}(x_k, u_k)
$$

3. Calculate time-varying Gramians

$$
\begin{aligned}
P_{k+1} &= A_k P_k A_k^T + B_k B_k^T, & P_1 &= 0 \\
Q_k &= A_k^T Q_{k+1} A_k + C_k^T C_k, & Q_{N+1} &= 0
\end{aligned}
$$

4. Determine the average Gramians

$$
\bar{P} = \frac{1}{N} \sum_{k=1}^{N} P_k \qquad\qquad \bar{Q} = \frac{1}{N} \sum_{k=1}^{N} Q_k
$$

5. Apply the standard balanced truncation method on $\bar{P}$ and $\bar{Q}$, which yields a balancing coordinate change $z = Tx$ and the singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$. See [Zhou and Doyle, 1998].

6. Determine the reduced model order $\hat{n}$ from the relative size of the singular values and truncate the coordinate change. T_l is the top $\hat{n}$ rows of T and T_r is the $\hat{n}$ leftmost columns of T^{-1}.

$$
\begin{aligned}
T \in \mathbf{R}^{n \times n} \quad &\Rightarrow \quad T_l \in \mathbf{R}^{\hat{n} \times n} \\
T^{-1} \in \mathbf{R}^{n \times n} \quad &\Rightarrow \quad T_r \in \mathbf{R}^{n \times \hat{n}}
\end{aligned}
$$

7. Apply the truncated coordinate change to the original system

$$
\begin{aligned}
\hat{z}_{k+1} &= T_l f(T_r \hat{z}_k, u_k) \\
y_k &= g(T_r \hat{z}_k, u_k)
\end{aligned}
$$

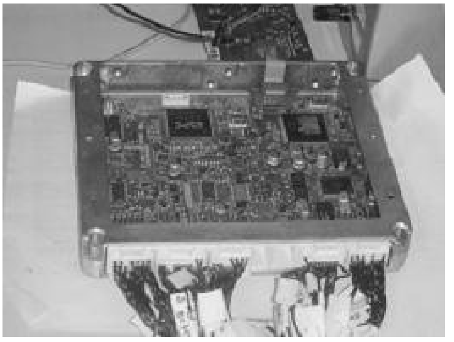

Figure 3.17 Engine Control Unit with the embedded controller software.

in order to achieve a certain air-fuel ratio. The exhaust treatment system requires a precise air-fuel ratio, therefore high fidelity of the estimation is crucial since a mismatch would yield suboptimal performance. For further information see [Heywood, 1988].

The model is implemented in *MATLAB®* / *Simulink®* and can be compiled through *Real-Time Workshop®*. The resulting binary runs in real-time in the Engine Control Unit(ECU), shown in Figure 3.17.

The model is devised for real-time purposes with limited hardware resources. For example, only discrete-time components are present and fixed-point arithmetic is used. The most common arrangement, used in automotive industry, to achieve high performance for a wide area of operating conditions is to divide the problem into regions and perform local tuning of variables. The model therefore contains a large amount of logical branches and look-up tables.

The model is a so called mean-value model, see [Hendricks *et al.*, 1996], but details concerning model implementation are proprietary information and are therefore not disclosed.

The model estimates several variables using various measurements. As a proof of concept, only one input-output pair is treated here. The chosen input signal is the throttle angle measured in degrees and the output signal is the air charge given in percentage, as illustrated in Figure 3.18.

Model reduction of the Simulink control algorithm implementation would ideally yield a binary file that runs faster and uses less memory. Hence, smaller hardware resources would be required yielding a lower controller hardware cost. In addition, formal validation and verification of the controller in combination with the engine would be facilitated. Moreover, the original model structure is hard to overview and it might be easier to understand the reduced model behaviour and visualize its components.

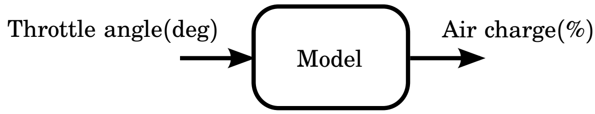

Figure 3.18 Input-output pair chosen for model reduction.

Model reduction applied to the controller model With some slight modification the controller software fits within the model class of (3.14). The state vector x represents the ECU memory used to store data between samples. The controller turned out to contain six states. The input signal u is the throttle angle and the output y is the air charge. Simulink provides tools for extraction of simulation data and linearizations. With this data available the model can be reduced following the procedure described in Section 3.3.

As stated, the first step is to choose a training trajectory. Here, a simple ramp-like throttle opening profile was chosen, see Figure 3.19, starting from closed throttle and then linearly increasing until fully open. Of course, different choices are possible depending of the purpose of the reduced model. Notice the nonlinear effect in Figure 3.19, the output has almost settled after half a second while the input continue to increase half a second more. This is because the flow over the throttle is much more sensitive to an increase in throttle angle when it is almost closed, see e.g. [Heywood, 1988].

With the training input signal chosen, the model can be linearized around the corresponding state-space trajectory. Doing so gives rise to the linearizations (A_k, B_k, C_k, D_k), for this model the D_k matrix is zero for all k. Following the procedure, the time-varying Gramians P_k and Q_k are computed through (3.15) and (3.16). The average Gramians $\bar{P}$ and $\bar{Q}$ are then obtained by (3.17). The singular values in (3.18) are shown in Figure 3.20, three of the six values turned out to be exactly zero and are not plotted. The relative size of these values indicates the importance of the states. Here, there is a factor 10^4 difference between the largest and second largest value. In model reduction of linear systems, one could easily reduce to one state. Despite the absence of a formal error bound this will be done for the nonlinear system as well. Calculating the balancing coordinate change and truncating to one state according to (3.19) yield the two matrices T_l and T_r.

The next and final step of the method consists of applying the coordinate change to the original nonlinear system. In this case, the functions f and g are not explicitly available but embedded in the Simulink program. Hence, symbolic manipulation of the functions is not a straight

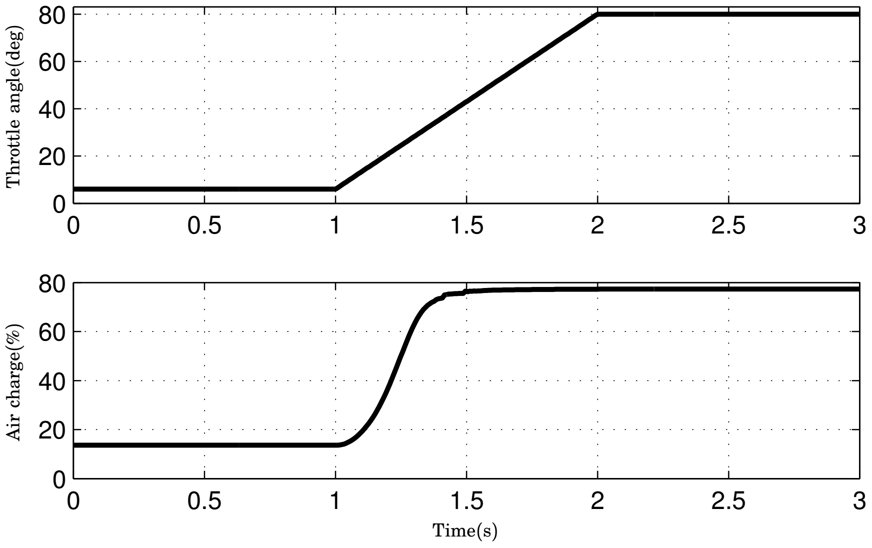

Figure 3.19 The ramp-like training input signal (throttle angle) and the corresponding output signal (air charge).

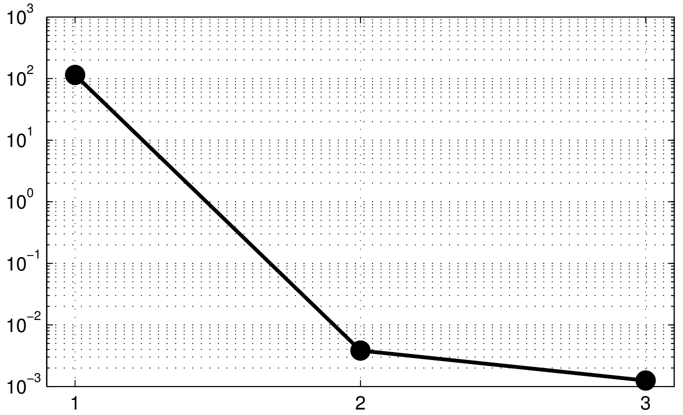

Figure 3.20 The three largest Hankel singular values, notice the 10^4 drop between the first and second value.

forward process. One could consider using the piece-wise affine approach as in Example 3.5. However, using linearizations as basis functions is not tractable in this case. Due to the logical branches and non-smooth look-up tables the function f becomes “noisy”. Therefore, a local linearization provides inadequate information about the neighbouring state space and an unreasonably large number of linearizations would be required.

By reducing to one state, the right-hand-side function in (3.20)

$$\hat{z}_{k+1} = \hat{f}(\hat{z}_k, u_k) = T_l f(T_r \hat{z}_k, u_k)$$

becomes a two-dimensional map $\hat{f} : \mathbf{R}^2 \to \mathbf{R}$. An alternative approach, used below, is to let the map $\hat{f}$ be generated from simulation data. State-trajectories induced by some input-signal could be projected with T_r and provide values for $\hat{z}_k$. Value triplets of $\hat{z}_{k+1}$, $\hat{z}_k$ and u_k supply point-wise information of the map. A drawback is that the input-signal must be rich enough to make sufficient state-space coverage in $(\hat{z}_k, u_k)$ and the choice can be nontrivial. The difficulties relate to the choice of training input in the first step of the model reduction procedure and although the purpose is different, the same input-signal could be used. A chirp signal with maximal amplitude, shown in Figure 3.21, is chosen as the exciting input-signal.

For simulation of the reduced model, point-wise information of the map is not sufficient, an analytical expression of $\hat{f}$ is needed. Through local averaging followed by linear interpolation and extrapolation, a piece-wise affine surface is generated to approximate the data cloud, see Figure 3.22. To clarify the structure of the map the incremental form $\hat{f}(\hat{z}, u) - \hat{z}$ instead of $\hat{f}(\hat{z}, u)$ is used. As the map is piece-wise affine, so is the reduced system. One can notice the rough areas in the upper part of the map, their origin is most probably the non-smooth look-up tables present in the model.

In general the same procedure would be applied to the output function g. Here, the output function $g(T_r \hat{z}_k, u_k) = \hat{g}(\hat{z})$ turned out to be nearly affine and a least-squares fit showed to be an adequate approximation.

With analytical expressions for $\hat{f}$ and $\hat{g}$ at hand, simulation of the reduced system is possible. Figure 3.23 shows a validation result where the original and reduced model are simulated with a ramp-like opening and closing of the throttle. The initial value of $\hat{z}$ is arbitrarily set, hence the initial mismatch between the two output signals. In contrast, after 0.5 seconds the worst case error is less than 1.2% air charge.

In Figure 3.24 the validation trajectory together with the data points used for map generation are shown. The smaller dots are from the chirp-signal simulation in Figure 3.21 and the connected dots from the validation simulation in Figure 3.23. As can be seen, the validation trajectory is covered by the look-up table. For a different validation scenario this might

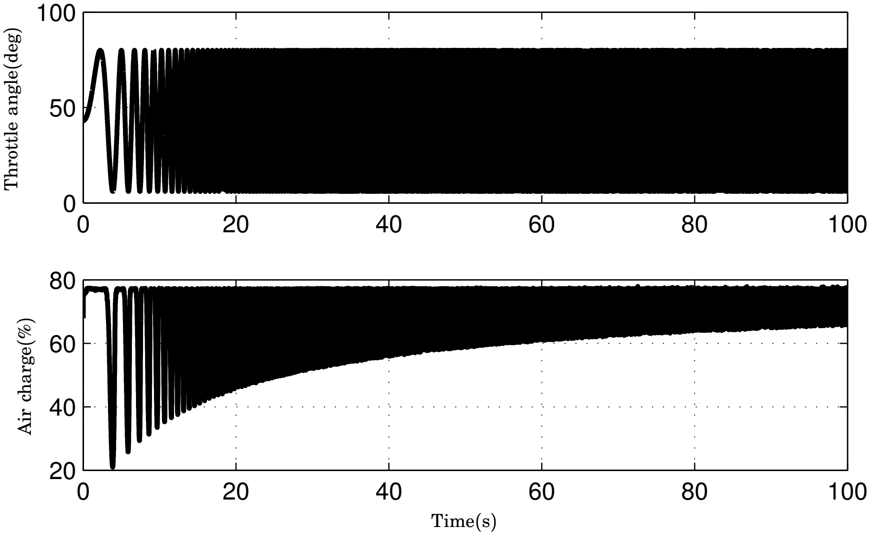

Figure 3.21 The chirp input signal used for map generation and the corresponding output signal.

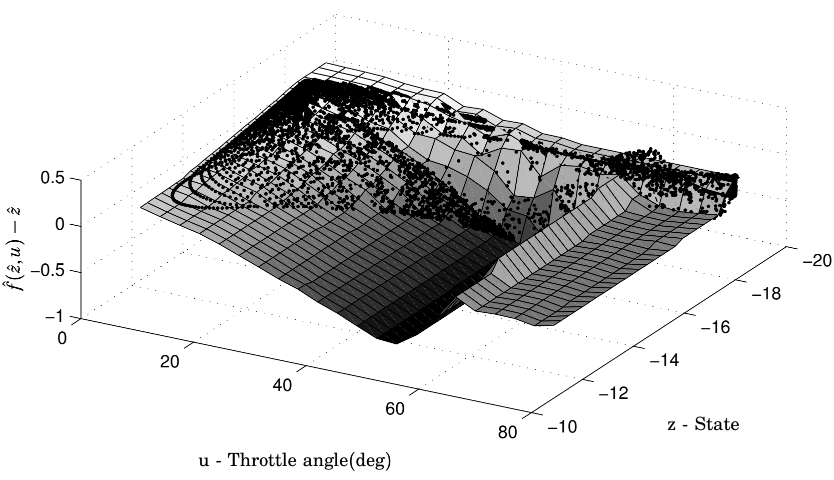

Figure 3.22 The data points and generated surface representing the right-hand-side function $\hat{f}(\hat{z},u)$, here visualized in incremental form.

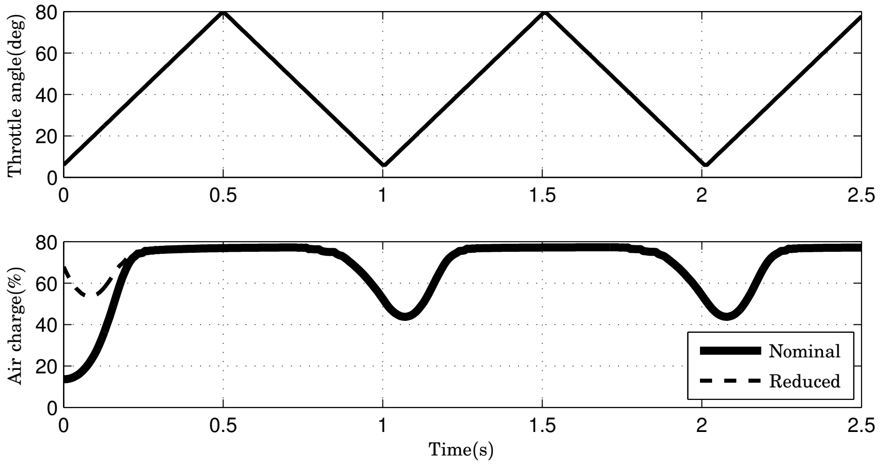

Figure 3.23 Validation result of the reduced model. The lower plot shows the output signal of the original model together with the reduced one. The initial output error is due to an unmatched initial condition.

not necessarily be the case. However, if necessary, a richer input signal could be designed to cover a larger area and a more general map could be generated. Due to the low dimensionality of this case, the coverage could be graphically examined. However, in a more general setting with higher dimensionality an automatized verification method would be required.

A compact Simulink® implementation of the reduced model is depicted in Figure 3.25. The model runs more than 100 times faster than the full original model. However, to carry out a fair comparison, the reduced model should also be equipped with the same amount of input and output signals.

□

3.4 Summary

A method for simplification of nonlinear input-output models has been outlined. The given procedure is focused on reducing the number of states using information obtained by linearization around trajectories. The use of trajectories seems to be an attractive approach. A model reduction method would greatly benefit of information about intended model usage. Implying that the reduced model should perform well for all possible input signals does not leave much room reduction. Furthermore, the assumptions made when constructing a physical model are probably only valid within

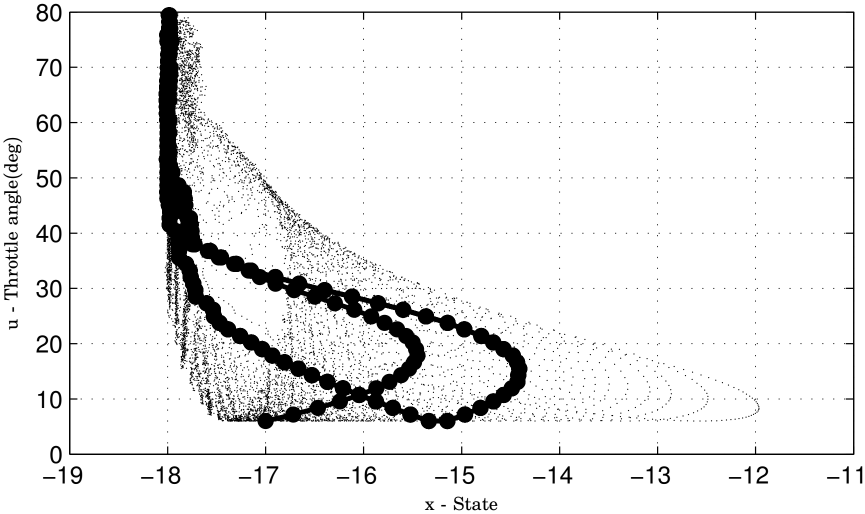

Figure 3.24 The validation trajectory(the connected dots) is covered by the span of data points(the smaller dots) used for the look-up table $\hat{f}(\hat{z}_k, u_k)$.

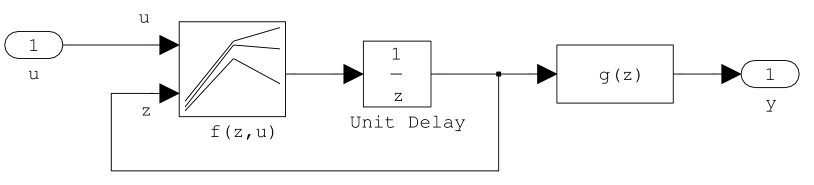

Figure 3.25 Simulink implementation of the reduced model.

certain intervals in state space and signals. Training trajectories is one way of introducing this information to the reduction method.

The number of states is only one factor contributing to simulation time and even though the method not necessarily provides a speed-up in the general case, simulation time has been reduced in the presented examples.

No proofs concerning preserved stability or error bounds are given. However, the methodology is closely tied to existing theory on error bounds and promising results are shown in form of examples and simulation data.

The method only requires linearizations, which makes systems that are hybrid or that are not given on a closed form approachable, see Example 3.7.

In the discrete-time example the software of an engine controller used in current production cars was reduced. The number of states were reduced from six to one and the resulting nonlinear piece-wise affine system showed a 100-fold improved simulation speed, with little loss of accuracy. Despite the model complexity in terms of look-up tables and logical switches the method demonstrated its applicability. The method also provided information for analysis of overall controller behaviour, such as the software visualisation in Figure 3.22.

4

Modeling the exhaust gas oxygen sensor

In this chapter an exhaust gas oxygen sensor model is developed, the chapter is based on [Nilsson, 2006].

4.1 Introduction

Lambda sensors, or exhaust gas oxygen (EGO) sensors, are core components in the emission control in modern spark ignition combustion engines. The sensor, shown in Figure 4.1, is typically placed in the exhaust gas manifold between the engine and the catalyst. The performance of catalysts is highly dependent on exhaust gas composition and, e.g., the air-fuel ratio needs to be precisely controlled, as mentioned in Chapter 2. A common air-fuel ratio control setup is illustrated in Figure 4.2.

There exists many different kinds of oxygen sensors but the most commonly used is the zirconia switch-type sensor, this chapter is focused on this type. The sensor generates a voltage of roughly one Volt if the air-fuel ratio is rich and zero Volt otherwise, see Figure 4.3. The lambda value is another name for air-fuel ratio and in this chapter, a normalized lambda

Figure 4.1 An exhaust gas oxygen sensor

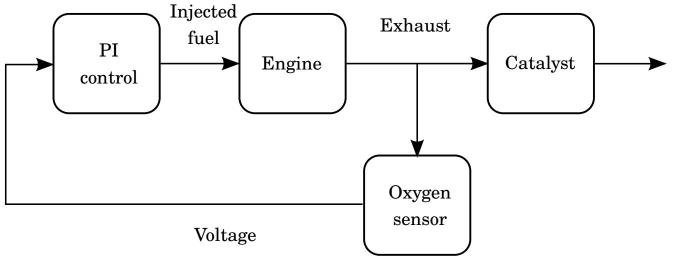

Figure 4.2 An air-fuel ratio control scheme

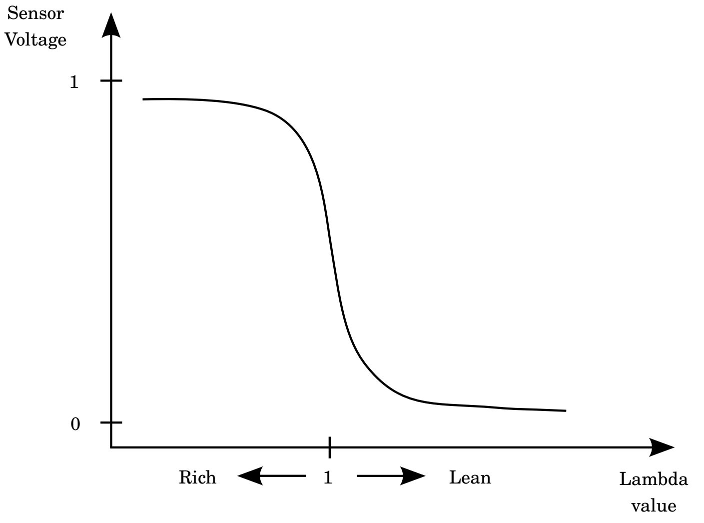

Figure 4.3 Lambda sensor characteristics

value is considered, where the value one implies stoichiometric conditions.

To meet future emission legislations, it is required to refine and extend current lambda control strategies. Good understanding of catalyst operation is essential to improve emission performance. It is necessary to understand the interaction of the catalyst and the lambda control system, including the lambda sensors, to optimize the exhaust gas treatment. Physically based simulation models are then vital tools to analyze and evaluate new control strategies. An important part in this task is the sensor models, and their ability to correctly reproduce effects of significance to catalyst operation. Of particular interest is the shift in voltage characteristics with respect to lambda value that is observed when the exhaust gas is diluted with hydrogen or carbon monoxide.

4.2 Modeling the exhaust gas oxygen sensor

A model with moderate complexity, which captures the lambda characteristics and its dependency of hydrogen and carbon monoxide is sought for. A model of reasonable complexity level is developed in [Fleming, 1977] but with the drawback of not being able to model hydrogen dependency. The model presented in [Auckenthaler *et al.*, 2002] is a very detailed and complex model based on state-of-the-art methods in literature. It models the hydrogen dependency along with many other effects. Unfortunately it suffers from numerical ill-conditioning. Due to the great time-scale difference between the electrode dynamics and the diffusion, the model becomes very stiff and is therefore difficult to use, e.g. in simulation. Possible model extensions could be

- Extend Fleming's model with hydrogen dependency.
- Derive an equilibrium approximation to Auckenthaler's model in order to avoid the numerical stiffness problem.

Both alternatives have been investigated, but more progress was obtained by following a model description found in [Barrick *et al.*, 1996]. This model is compact, considers hydrogen dependency and is static, so no numerical stiffness problem arises. Whether or not the dynamics of the sensor can be neglected depends on the control scheme and sensor placement. Here it is assumed that the dynamics can be disregarded.

The physics of the sensor

The sensor mainly consists of four manifold layers between the exhaust and reference gas (outside air), as can be seen in the cross section in Figure 4.4.

A close-up of the four layers is shown in Figure 4.5. As the figure indicates, the model by Barrick takes into account the species H_2, CO, O_2, CO_2 and H_2O, other species are assumed not to affect the sensor voltage.

Firstly, the exhaust gases have to diffuse through the porous spinel layer to affect the sensor voltage. Different species have different mass, and therefore different diffusion velocities, so the concentrations at the platinum surface are different compared to the ones in the exhaust gas close to the sensor. Secondly, at the cathode surface the platinum acts as a catalyst for the chemical reactions bringing the modified gas mixture to chemical equilibrium. And finally, the difference in gas concentrations at the electrodes yields the sensor voltage.

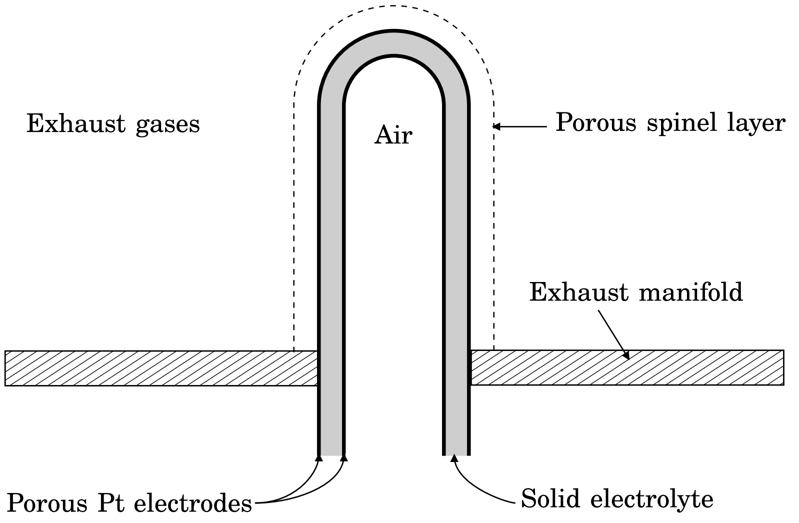

Figure 4.4 Sensor cross section

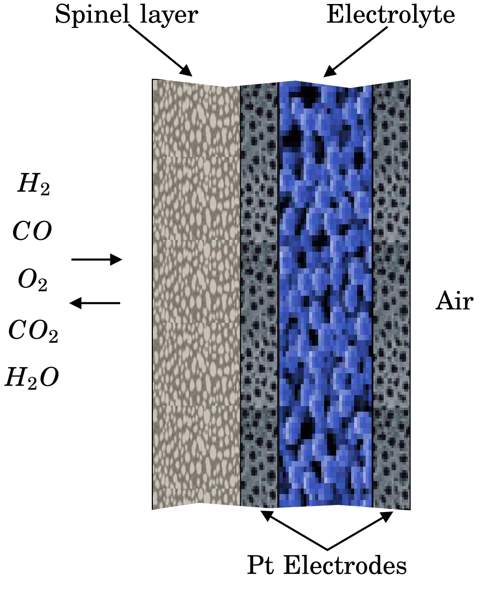

Figure 4.5 Sensor layer close-up

The sensor model can thus be divided into three parts

- Diffusion through the porous spinel layer
- Platinum surface reactions
- The resulting sensor voltage

Here these three parts will be studied in more detail.

Diffusion N_2 is assumed to be abundant in the exhaust gas and is viewed as a media for the other species. Binary diffusion (also called Fick's law diffusion) is then a valid approximation. In [Barrick *et al.*, 1996] a more detailed transport model with Stefan-Maxwell diffusion was investigated without gaining much accuracy. Thus, in principle the species diffuse through the porous layer without interacting with each other. When the platinum surface is reached they will combine according to the reactions in Equation 4.2 until chemical equilibrium is reached and diffuse out of the sensor again. Balancing the steady-state flux of the three kinds of atoms gives rise to three linear equations

$$\begin{aligned} D_{CO_2}(X_{CO_2} - X_{CO_2}^{exh}) &= -D_{CO}(X_{CO} - X_{CO}^{exh}) \\ D_{H_2O}(X_{H_2O} - X_{H_2O}^{exh}) &= -D_{H_2}(X_{H_2} - X_{H_2}^{exh}) \\ D_{O_2}(X_{O_2} - X_{O_2}^{exh}) &= \frac{1}{2}D_{CO}(X_{CO} - X_{CO}^{exh}) + \frac{1}{2}D_{H_2}(X_{H_2} - X_{H_2}^{exh}) \end{aligned} \tag{4.1}$$

where

- X_i, X_i^{exh} are the molar fractions of gas i at the platinum surface resp. in the exhaust.
- D_i is the diffusion coefficient of gas i in N_2, which is dependent of temperature, pressure and tortuosity of the material.

Platinum surface reactions The model contains five active species interacting through two simplified reactions

$$\begin{aligned} CO + \frac{1}{2}O_2 &\rightleftharpoons CO_2 \\ H_2 + \frac{1}{2}O_2 &\rightleftharpoons H_2O \end{aligned} \tag{4.2}$$

A key simplification is to assume chemical equilibrium at the platinum surface, which induces two nonlinear algebraic equations

$$\begin{aligned} X_{CO}\sqrt{X_{O_2}} &= k_C X_{CO_2} \\ X_{H_2}\sqrt{X_{O_2}} &= k_H X_{H_2O} \end{aligned} \tag{4.3}$$

where k_C and k_H are temperature dependent constants arising from reaction velocities.

Sensor voltage In order to keep down model complexity, the three-phase boundary cites (where species can be adsorbed) are assumed to be abundant. There is no competition for vacant sites, so the voltage model only depends on O_2 concentration, see [Barrick *et al.*, 1996]. The sensor voltage is then obtained by

$$V = -\frac{RT}{4F} \ln \frac{X_{O_2}}{0.21} \tag{4.4}$$

where R is the universal gas constant, T temperature in Kelvin, F Faraday's constant and 0.21 is the molar fraction of oxygen in the reference air.

An extension to this voltage model was described in [Fleming, 1977] where the carbon monoxide chemical effect on the voltage was included. It is however equivalent to (4.4) when assuming chemical equilibrium at the platinum surface so the simpler version was chosen in favour of low complexity.

Parameter estimation

For this model no calibration experiments are needed since all parameters are physical constants. Some are very well known, e.g., Faraday's constant, and others can be estimated using semi-empirical formulas.

Diffusion velocities The diffusion velocities have been estimated using the method by Chapman and Enskog, see [Reid *et al.*, 1977]. This method has an accuracy of about 6% error margin. As mentioned, the velocity is temperature and pressure dependant. However, all velocities have approximately the same dependencies, so by dividing with a nominal velocity in equation 4.1, the temperature and pressure dependencies can be omitted.

The method by Chapman and Enskog estimates the diffusion velocity in an open geometry. In the sensor however, gases diffuse through a porous media and this has to be taken into account. The standard way to deal with this, see [Smith, 1981], is to multiply the nominal velocity with a material dependent factor

$$D_i^* = \frac{\epsilon}{\tau} D_i$$

The tortuosity, τ, and the porosity, ϵ, are material specific and the same for all species i. Inserted in (4.1), the factor does not have an impact and can therefore also be disregarded.

Chemical equilibrium constants The program *HSC Chemistry*^{TM} was used to estimate the chemical equilibrium constants k_C and k_H.

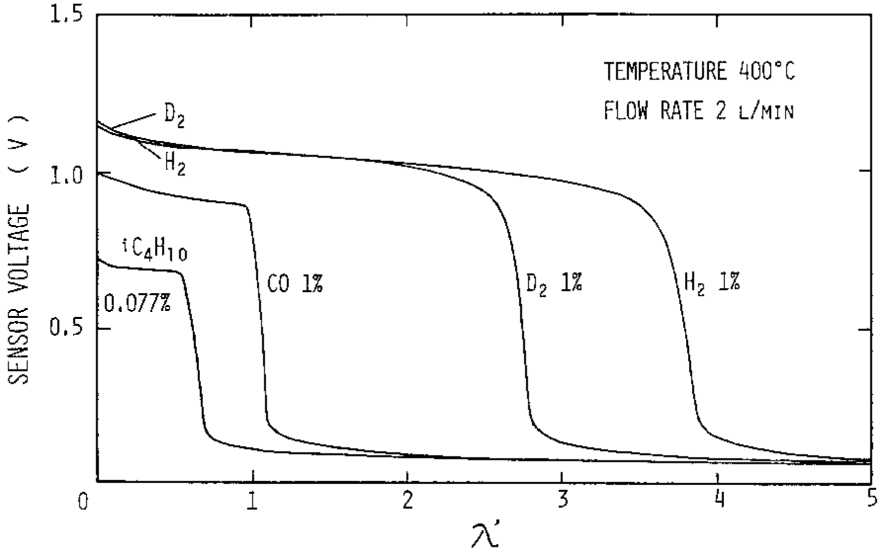

Figure 4.6 Lambda characteristics for non-equilibrium gas concentrations from [Saji *et al.*, 1988]. Reproduced by permission of ECS - The Electrochemical Society

Lambda characteristics perturbation

This effect is analyzed in [Saji *et al.*, 1988], where a lambda sensor was exposed to different test gas mixtures. Some results presented in the article are illustrated in Figure 4.6, where it can be seen that deuterium has a different impact than H_2 although they have the same chemical properties. This displays that the perturbation effect is not due to chemical reactions but to physical properties, i.e. diffusion. The article also claims that this effect depends on non-equilibrium gas concentrations in combination with diffusion.

For example, for lambda equal to 1.1 a gas mixture in chemical equilibrium would contain O_2 and almost no H_2. The O_2 would be able to undisturbed diffuse to the cathode and as the sensor mainly produces voltage in function of difference in O_2 concentration, no(or low) voltage would be obtained. However, if the gas is not in chemical equilibrium there is O_2 and H_2 present in the exhaust. H_2 diffuses faster than oxygen and the cathode would be exposed to a disproportionally large amount of H_2. The cathode acts as a catalyst for the reaction

$$H_2 + \frac{1}{2}O_2 \rightleftharpoons H_2O$$

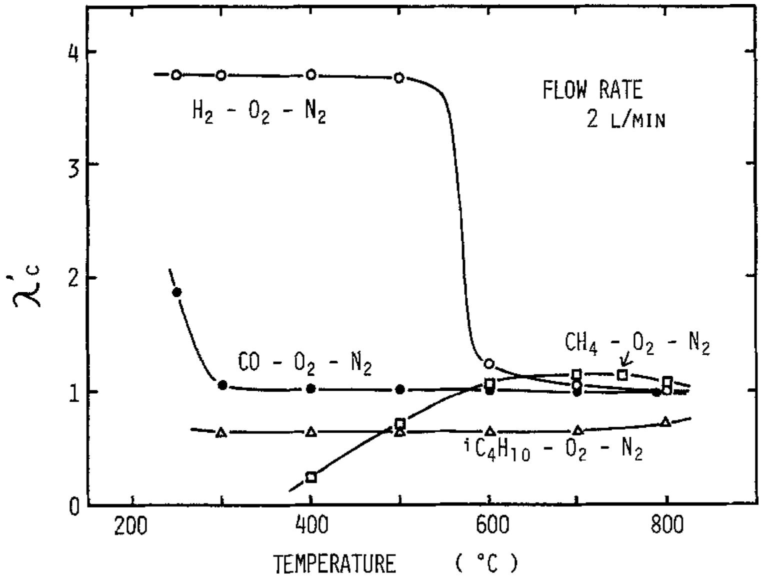

Figure 4.7 Switch point sensibility to temperature from [Saji *et al.*, 1988]. Reproduced by permission of ECS - The Electrochemical Society

and the oxygen is depleted inducing a high sensor voltage. Similar reasoning can be applied to other disturbing gases, the deviation depends on the degree of non-equilibrium and the mass difference between the species. This explains the perturbations in Figure 4.6, taken from [Saji *et al.*, 1988].

If the exhaust gas is heated to a higher temperature the gas would have a higher probability to reach equilibrium before exposing the sensor as can be seen in Figure 4.7, also taken from [Saji *et al.*, 1988].

Possible model extensions

- Sensor dynamics can probably be neglected, but should preferably be modeled in case it has importance.

  One heuristic way to introduce dynamics could be by adding, to the current static model, a linear first order filter with a time constant corresponding to the diffusion time of oxygen.

- Consider, and if necessary include in model, effects of NO_x and methane in the exhaust gas.

  A first attempt could be to model the effect NO_x and methane has on gas outside the sensor and in that way perturb the voltage. This can be motivated since NO_x and methane are both heavy species, and therefore diffuse slowly. For this reason they probably do not have an active role at the cathode layer.

4.3 Implementation

The implementation has been done in the Modelica language, see [Fritzson, 2004]. The model sums up into a set of nonlinear algebraic equations

$$
\begin{aligned}
D_{CO_2}(X_{CO_2} - X_{CO_2}^{exh}) &= -D_{CO}(X_{CO} - X_{CO}^{exh}) \\
D_{H_2O}(X_{H_2O} - X_{H_2O}^{exh}) &= -D_{H_2}(X_{H_2} - X_{H_2}^{exh}) \\
D_{O_2}(X_{O_2} - X_{O_2}^{exh}) &= \frac{1}{2}D_{CO}(X_{CO} - X_{CO}^{exh}) + \frac{1}{2}D_{H_2}(X_{H_2} - X_{H_2}^{exh}) \\
X_{CO}\sqrt{X_{O_2}} &= k_C X_{CO_2} \\
X_{H_2}\sqrt{X_{O_2}} &= k_H X_{H_2O} \\
V &= -\frac{RT}{4F}\ln\frac{X_{O_2}}{0.21}
\end{aligned}
$$

where the exhaust molar fractions, X_i^{exh}, are considered as model inputs and the sensor voltage, V, as model output. The sole alteration that has to be done to get a working Modelica code is the coordinate change

$$Y_i = \log(X_i)$$

for molar fractions inside the sensor. The new coordinates give far better numerical results for calculating small concentrations. The Modelica code has been simulated with the software tool Dymola, see [Dynasim AB, 2006]. The top view diagram is shown in Figure 4.8.

4.4 Simulations

The top plot in Figure 4.9 shows a 6 species (N_2 is omitted) varying gas configuration, this gas is at all lambda values in chemical equilibrium. The lower plot of the same figure show the same gas mixtures but the species were manually modified to deviate from chemical equilibrium, by pushing the reactions in (4.2) from the equilibrium points.

Exposing the model to these gas mixtures yields the voltage in Figure 4.10. As can be seen, the model shows promising and reasonable results, the voltage shift due to non-equilibrium H_2 is clearly visible.

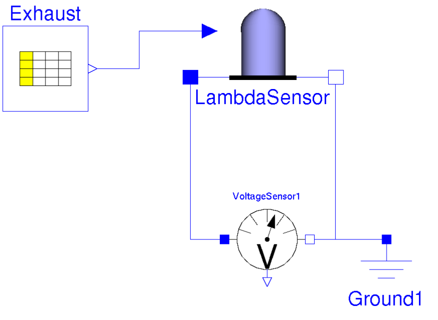

Figure 4.8 Layout of Dymola model

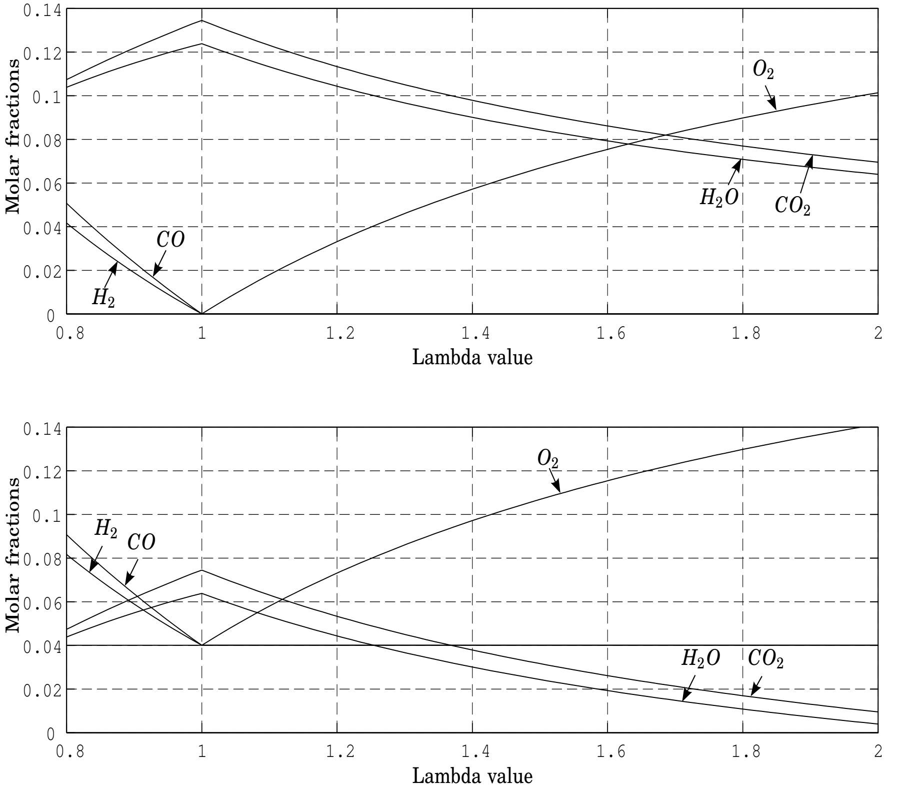

Figure 4.9 Gas mixtures with the corresponding lambda value

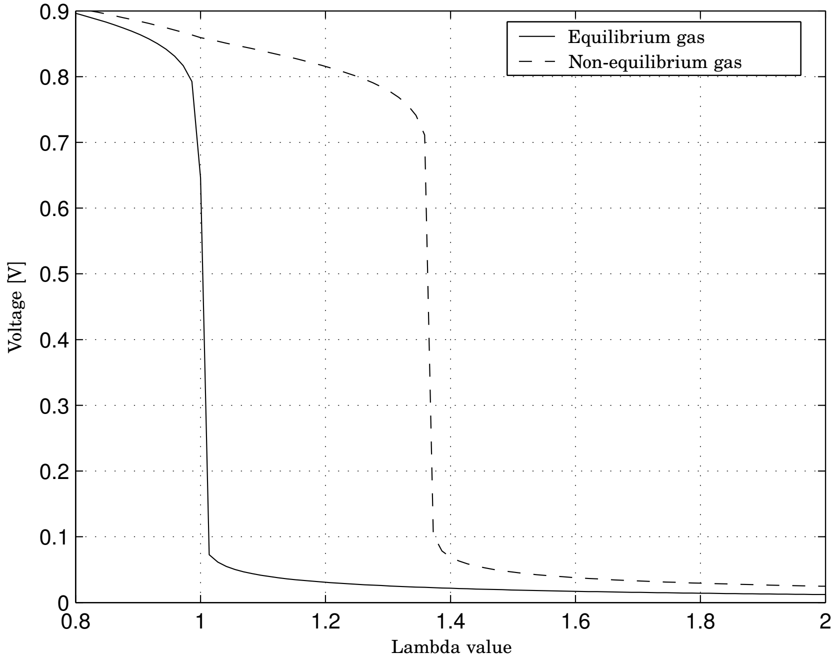

Figure 4.10 Lambda characteristics simulation

4.5 Model validation

Data from test-gas experiments has been used in validation purposes. In-house experiments were conducted but the equipment for oxygen concentration measurement turned out to have inadequate resolution. Instead, other data was used that, unfortunately, are proprietary information and not publishable.

Two types of experiments were used for validation, a constant flow of hydrogen or carbon monoxide were mixed with oxygen and nitrogen. The gas mixture was then heated to 500°C and exposed to the sensor, see Figure 4.11. During the tests the sensor voltage together with the lambda value were measured while the gas composition was changed according to Figure 4.12.

The lambda value for the hydrogen experiment is defined by the combustion reaction

$$2H_2 + \lambda O_2 \rightarrow n_p(X_{H_2}H_2 + X_{O_2}O_2 + X_{H_2O}H_2O)$$

where n_p is the total mole amount of the gas. An expression for the lambda

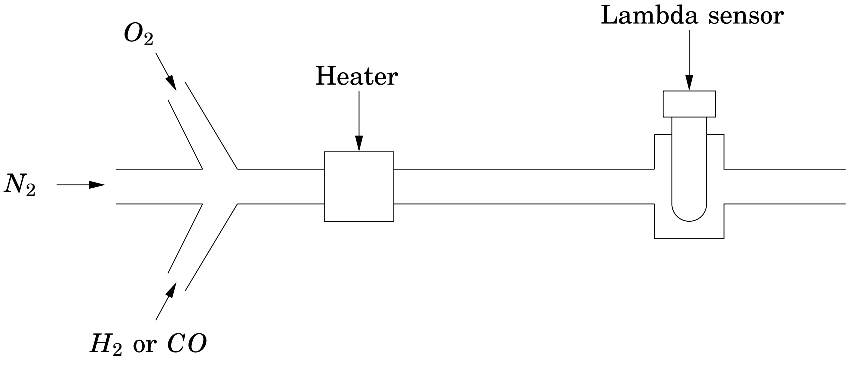

Figure 4.11 Experiment setup

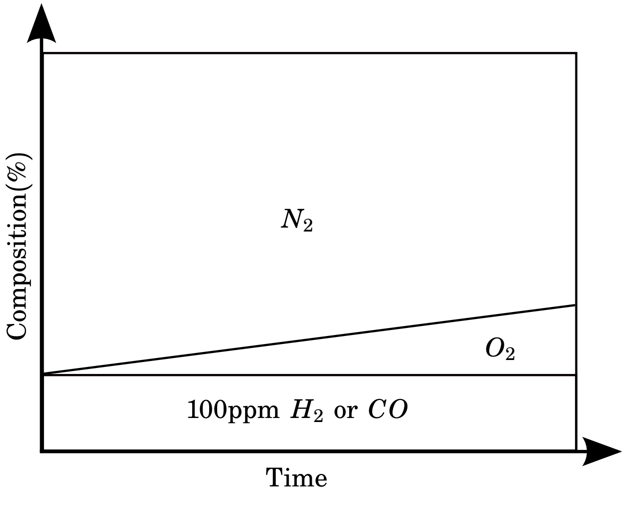

Figure 4.12 Experiment gas composition

value is achieved by balancing the amount of hydrogen and oxygen atoms.

$$\lambda_H = \frac{X_{H_2O} + 2X_{O_2}}{X_{H_2} + X_{H_2O}} \tag{4.5}$$

The same procedure for carbon monoxide yields

$$\lambda_C = \frac{X_{CO_2} + 2X_{O_2}}{X_{CO} + X_{CO_2}} \tag{4.6}$$

As can be seen, the gas configuration is not uniquely defined even though the hydrogen (resp. carbon monoxide) concentration is known. To analyse the completeness of combustion reactions at the sensor position in Figure 4.11, the software *Cantera* [Cantera, 2006] was used to simulate the reaction dynamics. It turned out that a temperature above 700°C is needed to activate the reactions in the mixed gas. The reaction speeds can be seen in Figure 4.13 and 4.14. For lower temperatures it's a good approximation to assume that the gases do not react and the concentrations remain unchanged until they reach the sensor.

The absence of H_2O and CO_2 in the mixed gas change (4.5) and (4.6) into

$$\lambda_H = \frac{2X_{O_2}}{X_{H_2}}$$
$$\lambda_C = \frac{2X_{O_2}}{X_{CO}}$$

Now the gas composition is uniquely defined by the lambda value and the corresponding experiment can be simulated with the model derived in Section 4.2.

Validation results The simulated model has been compared with the experiment data, with 100ppm of H_2 resp. CO at a temperature of 500°C. As mentioned, the experiment data are proprietary information and is not publishable. However, the model output, shown in Figure 4.15, captures the sensor behaviour well. The curves have approximately the same switching point as the measured data and the voltage, at the rich and lean sides, does not deviate much. The mean absolute error compared to experiment data is 0.14V for CO and 0.057V for H_2.

Calibration Most of the model parameters described in Section 4.2 are estimated with a level of uncertainty. If they are treated more as non-fixed parameters than natural constants, then higher accuracy to the experiment data can be achieved. In Figure 4.16 the calibrated model's output is shown. Now the mean absolute error is reduced to 0.033V for CO and 0.0261V for H_2. Here the diffusion velocities are modified but kept inside the Chapman and Enskog method's error margin. The reaction constant was modified corresponding to a 100°C change to match the voltage level for rich mixtures of carbon monoxide.

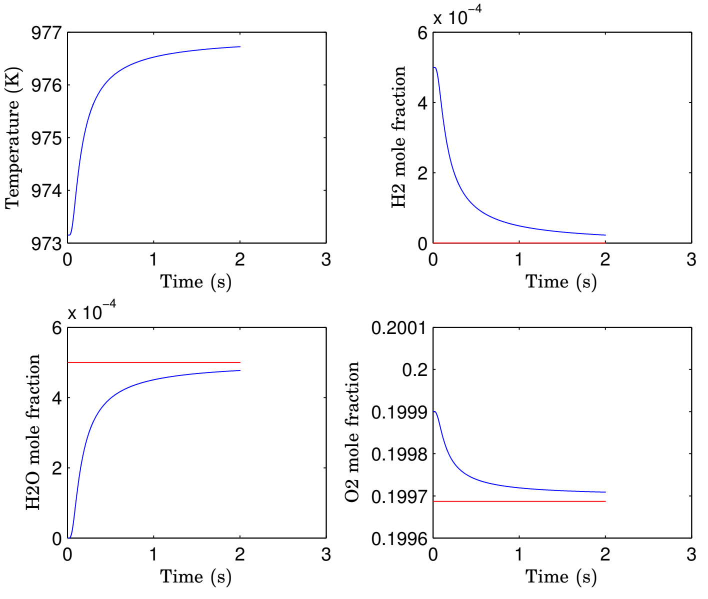

Figure 4.13 Reaction speed with hydrogen

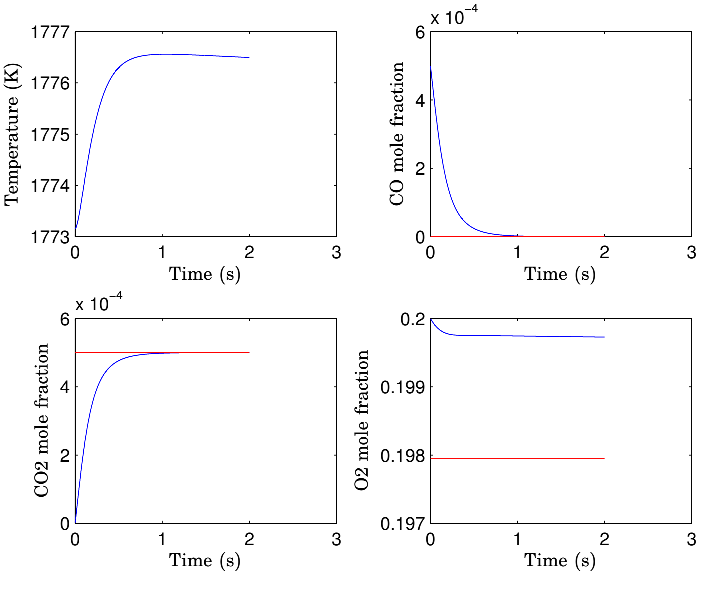

Figure 4.14 Reaction speed with carbon monoxide

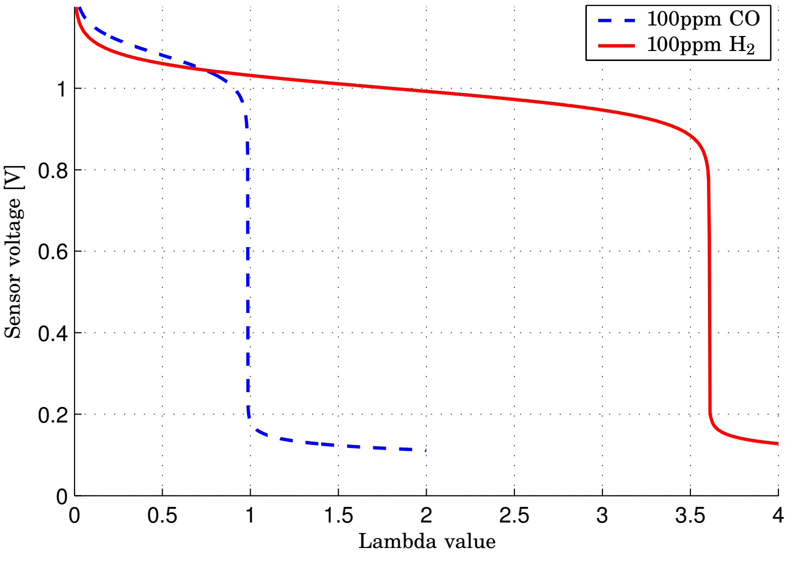

Figure 4.15 Model output for a gas mixture with 100ppm CO or H_2 at 500°C

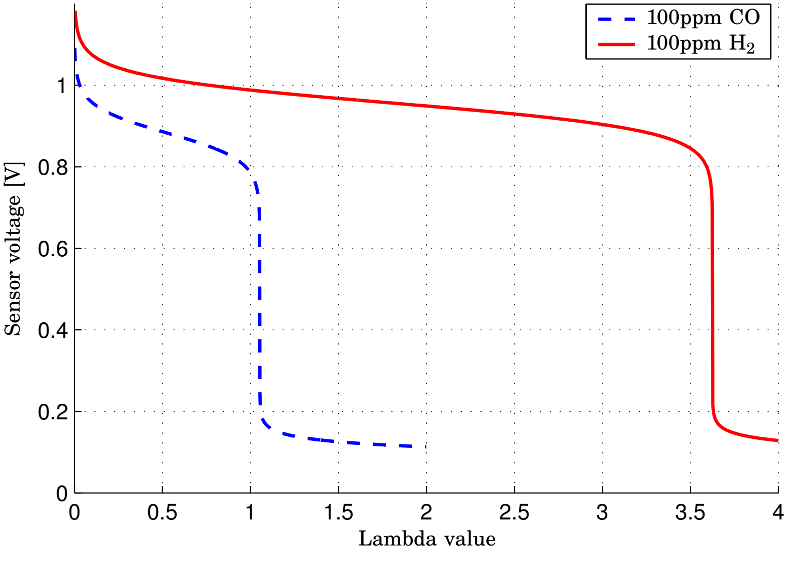

Figure 4.16 Calibrated model output for a gas mixture with 100ppm CO or H_2 at 500°C

4.6 Conclusions

A simple and static model with H_2 dependency was developed and implemented in the Modelica language. Simulations show reasonable results where the effects of H_2 in a non-equilibrium gas can be observed.

The model has successfully been validated with test gas experiment data. By adapting parameters within reasonable physical limits, higher fidelity to experiment measurements was achieved. The mean error in sensor output voltage did not exceed 3% of the maximum output, when the model was compared to experiment data.

5

Conclusions

The current control design development process in automotive industry involves many expensive experiments and hand-tuning of control parameters. Model based control design is a promising approach to reduce costs and development time. In this process low complexity models are essential. This thesis combines the areas of modeling and model reduction of automotive systems.

In Chapter 2 a model reduction method comparison is conducted on an engine air path model. The heuristic method commonly used when modeling engine dynamics is compared with a more systematic method based on balanced truncation. Both methodologies have their advantages and disadvantages. If a detailed model is available and linear behaviour is expected then the balanced truncation methodology could be preferred. This technique may require a large computation time but needs very little manual attention. Using the heuristic method requires more experience and knowledge, it may also involve extensive parameter fitting, but renders more insight to the simplifications made.

In Chapter 3 a method for model reduction of nonlinear input-output models is presented. The given procedure is focused on reducing the number of states using information obtained by linearization around trajectories. No proofs concerning preserved stability or error bounds are presented. However, the methodology is closely tied to existing theory on error bounds and good results are shown in form of examples and simulation data.

Finally, in Chapter 4 a model of the exhaust gas oxygen sensor is developed. The end result is a simple and static model that is sensitive to non-equilibrium concentrations of H_2. The model is implemented in the Modelica language and was successfully validated with test gas experiment data. The mean error in sensor output voltage did not exceed 3% of the maximum output, when the model was compared to experiment data.

6

Bibliography

Al-Saggaf, U. and G. Franklin (1987): "An error-bound for a discrete reduced-order model of a linear multivariable system." *IEEE Transactions on Automatic Control*, **32**, pp. 815–819.

Antoulas, A. and D. Sorensen (2001): "Approximation of large-scale dynamical systems: An overview." *International Journal of Applied Mathematics and Computational Science*, **11**, pp. 1093–1121.

Astrid, P. (2004): *Reduction of process simulation models: a proper orthogonal decomposition approach*. PhD thesis, Technische Universiteit Eindhoven, Netherlands.

Auckenthaler, T. S., C. H. Onder, and H. P. Geering (2002): "Modelling of a solid-electrolyte oxygen sensor." SAE Technical Paper 2002-01-1293.

Barrick, G., D. Calhoun, X. Chen, B. von Dohlen, X. Huang, T. Leise, H. Lomeli, R. Michler, S. Othman, P. Worfolk, and D. Baker (1996): "A mathematical model for an exhaust oxygen sensor." Technical Report IMA Pre-print Series #1422-6. Institute for Mathematics & its Applications, University of Minnesota – Minneapolis, Minnesota 55455.

Broz, J., C. Clauss, T. Halfmann, P. Lang, R. Martin, and P. Schwarz (2006): "Automated symbolic model reduction for mechatronical systems." In *Proc. of 2006 IEEE International Symposium on Computer-Aided Control Systems Design*, pp. 408–415. IEEE, Munich, Germany.

Cantera (2006): "Object-Oriented Software for Reacting Flows." Home page: http://cantera.sourceforge.net.

Chevalier, A., C. Vigild, and E. Hendricks (2000): "Predicting the port air mass flow of SI engine in Air/Fuel ratio control applications." Number 2000-01-0260. SAE Technical Paper.

Dynasim AB (2006): "Dymola-Dynamic Modeling Laboratory, Users Manual." http://www.dynasim.se.

Farhood, M. and G. Dullerud (2006): "On the balanced truncation of LTV systems." *IEEE Transactions on Automatic Control*, **51:2**, pp. 315– 320.

Fleming, W. J. (1977): "Physical principles governing nonideal behavior of the zirconia oxygen sensor." *Journal of the Electrochemical Society*, **124:1**, pp. 21–28.

Føns, M., C. Vigild, A. Chevalier, E. Hendricks, S. Sorenson, and M. Müller (1999): "Mean value engine modelling of an SI engine with EGR." Number 1999-01-0909. SAE Technical Paper.

Fritzson, P. (2004): *Principles of object-oriented modeling and simulation with Modelica 2.1*. Wiley-IEEE Press.

Fujimoto, K. and D. Tsubakino (2006): "On computation of nonlinear balanced realization and model reduction." In *Proceedings of American Control Conference*, pp. 460–465. IEEE, Minneapolis, USA.

Grimme, E. J. (1997): *Krylov Projection Methods for Model Reduction*. PhD thesis, University of Illinois at Urbana-Champaign.

Hahn, J. and T. Edgar (2002): "An improved method for nonlinear model reduction using balancing of empirical gramians." *Computers and Chemical Engineering*, **26:10**, pp. 1379–1397.

Hahn, J., T. Edgar, and W. Marquardt (2003): "Controllability and observability covariance matrices for the analysis and order reduction of stable nonlinear systems." *Journal of Process Control*, **13:2**, pp. 115–127.

Haugwitz, S. (2007): *Modeling, Control and Optimization of a Plate Reactor*. PhD thesis ISRN LUTFD2/TFRT--1080--SE, Department of Automatic Control, Lund University, Sweden.

Haugwitz, S., P. Hagander, and T. Norén (2007): "Modeling and control of a novel heat exchange reactor, the open plate reactor." *Control Engineering Practice*, **15:7**, pp. 779–792.

Hendricks, E., A. Chevalier, M. Jensen, S. Sorenson, D. Trumpy, and J. Asik (1996): "Modelling of the intake manifold filling dynamics." Number 960037. SAE Technical Paper.

Heywood, J. (1988): *Internal Combustion Engine Fundamentals*. McGraw-Hill, New York.

Karhunen, K. (1946): "Zur spektraltheorie stochastischer prozesse." *Ann. Acad. Sci. Fennicae*, **Ser.A1:34**.

Khalil, H. K. (2002): *Nonlinear systems*, third edition. Prentice Hall, Upper Saddle River, New Jersey.

Krener, A. J. (2008): *Analysis and Design of Nonlinear Control Systems - In Honor of Alberto Isidori*, chapter Reduced Order Modeling of Nonlinear Control Systems, pp. 41–62. Springer-Verlag, Berlin,Germany.

Lall, S. and C. Beck (2003): “Error-bounds for balanced model-reduction of linear time-varying systems.” *IEEE Transactions on Automatic Control*, **48:6**, pp. 946–956.

Lall, S., J. Marsden, and S. Glavaski (2002): “A subspace approach to balanced truncation for model reduction of nonlinear control systems.” *International Journal of Robust and Nonlinear Control*, **12**, pp. 519–535.

Li, J.-R. (2000): *Model Reduction of Large Linear Systems via Low Rank System Grammians*. PhD thesis, Massachusetts Institute of Technology.

Li, L. and F. Paganini (2005): “Structured coprime factor model reduction based on LMIs.” *Automatica*, **41:1**, pp. 145–151.

Liu, Z. and J. Wagner (2002): “Nonlinear model reduction for dynamic and automotive system descriptions.” *Journal of Dynamic Systems, Measurement, and Control*, **124:4**, pp. 637–647.

Loève, M. (1945): “Fonctions aléatoires de second ordre.” *Comptes Rendus Acad. Sci. Paris*, pp. 220–469.

Lumley, J. (1967): “The structure of inhomogeneous turbulence.” *Atmospheric turbulence and wave propagation*, **4**, pp. 166–178.

Modelon (2007): “Vehicle dynamics library.” http://www.modelon.se.

Moin, L. and V. Uddin (2004): “A unified modeling approach using bond graph method and its application for model order reduction and simulation.” In *Proceedings of IEEE International Multitopic Conference*, pp. 536– 541. IEEE, Lahore Pakistan.

Moore, B. (1981): “Principal component analysis in linear systems: controllability, observability, and model reduction.” *IEEE Transactions on Automatic Control*, **26:1**, pp. 17–32.

Newman, A. J. and P. S. Krishnaprasad (1998): “Computation for nonlinear balancing.” In *Proceedings of the 37th IEEE Conference on Decision & Control*, vol. 4, pp. 4103–4104. IEEE, Tampa, Florida, USA.

Nilsson, O. (2006): “Modeling and model reduction in automotive systems.” Licentiate Thesis ISRN LUTFD2/TFRT--3242--SE. Department of Automatic Control, Lund University, Sweden.

Nilsson, O. and A. Rantzer (2009a): “The average Gramian approach to nonlinear model reduction.” *IEEE Transactions on Control Systems Technology*. Preprint, submitted.

Nilsson, O. and A. Rantzer (2009b): “A novel approach to balanced truncation of nonlinear systems.” In *European Control Conference*. Preprint, submitted.

Nilsson, O. and A. Rantzer (2009c): “A novel nonlinear model reduction method applied to automotive controller software.” In *American Control Conference*. Preprint, accepted.

Nilsson, O., A. Rantzer, and J. Chauvin (2006): “A model reduction case study: Automotive engine air path.” In *Proceedings of the IEEE International Conference on Control Applications*. Munich, Germany.

Obinata, G. and B. Anderson (2001): *Model Reduction for Control System Design*. Springer-Verlag, London.

Phillips, J. (2003): “A statistical perspective on nonlinear model reduction.” In *2003 IEEE International Behavioral Modeling and Simulation Conference*, pp. 41–46. IEEE, San José, California, USA.

Phillips, J. and L. Silveira (2005): “Poor man’s TBR: A simple model reduction scheme.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **24:1**, pp. 43–55.

Prajna, S. and H. Sandberg (2005): “On model reduction of polynomial dynamical systems.” In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005*. Seville, Spain.

Reid, R., J. Prausnitz, and T. Sherwood (1977): *The Properties of Gases and Liquids*. McGraw-Hill, New York.

Rewieński, M. J. (2003): *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Saji, K., H. Kondo, T. Takeuchi, and I. Igarashi (1988): “Voltage step characteristics of oxygen concentration cell sensors for nonequilibrium gas mixtures.” *Journal of the Electrochemical Society: Electrochemical Science and Technology*, **135:7**, pp. 1686–1691.

Sandberg, H. (2006): "A case study in model reduction of linear time-varying systems." *Automatica*, **42:3**, pp. 467–472.

Sandberg, H. and R. M. Murray (2008): "Model reduction of interconnected linear systems using structured gramians." In *Proceedings of the IFAC World Congress*. Seoul, Korea.

Sandberg, H. and A. Rantzer (2004): "Balanced truncation of linear time-varying systems." *IEEE Transactions on Automatic Control*, **49:2**, pp. 217–229.

Scherpen, J. (1993): "Balancing for nonlinear systems." *Systems and Control Letters*, **21:2**, pp. 143–153.

Scherpen, J. and K. Fujimoto (2003): "Nonlinear balanced realization based on singular value analysis of Hankel operators." In *Proceedings of the 42nd IEEE Conference on Decision & Control*, pp. 6072–6077. IEEE, Maui, USA.

Scherpen, J. and K. Fujimoto (2004): "Balancing and model reduction for discrete-time nonlinear systems based on Hankel singular value analysis." In *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*. Leuven, Belgium.

Shokoohi, S., L. Silverman, and P. van Dooren (1983): "Linear time-variable systems: Balancing and model reduction." *IEEE Transactions on Automatic Control*, **28:8**, pp. 810–822.

Shokoohi, S. and L. M. Silverman (1987): "Identification and model reduction of time-varying discrete-time systems." *Automatica*, **23:4**, pp. 509–521.

Siahaan, H. B. (2008): "A balancing approach to model reduction of polynomial nonlinear systems." In *Proceedings of the IFAC World Congress*. Seoul, Korea.

Sjöberg, J., T. Glad, and K. Fujimoto (2007): "Model reduction of nonlinear differential-algebraic equations." In *NOLCOS 07*. IFAC.

Smith, J. M. (1981): *Chemical Engineering Kinetics*. McGraw-Hill, New York.

Sou, K. C., A. Megretski, and L. Daniel (2008): "Convex relaxation approach to the identification of the Wiener-Hammerstein model." In *Proceedings of the 47th IEEE Conference on Decision & Control*, pp. 1375–1382. IEEE, Cancun Mexico.

Stykel, T. (2004): "Gramian based model reduction for descriptor systems." *Mathematics of Control, Signals and Systems*, **16**, pp. 297–319.

Vandendorpe, A. and P. Van Dooren (2004): "On model reduction of interconnected systems." In *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*. Leuven, Belgium.

Vasilyev, D., M. Rewienski, and J. White (2003): "A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS." In *Proceedings of Design Automation Conference*, pp. 490–495. IEEE, Anaheim, USA.

Vasilyev, D., M. J. Rewieński, and J. White (2006): "Macromodel generation for biomems components using a stabilized balanced truncation plus trajectory piecewise-linear approach." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **25:2**.

Verriest, E. I. and T. Kailath (1983): "On generalized balanced realizations." *IEEE Transactions on Automatic Control*, **28:8**, pp. 833–844.

Zhang, Q., A. Iouditski, and L. Ljung (2006): "Identification of wiener system with monotonous nonlinearity." In *Proc. IFAC Symposium on System Identification SYSID 06*. Newcastle, Australia.

Zhou, K. and J. C. Doyle (1998): *Essentials of Robust Control*. Prentice Hall, Upper Saddle River, New Jersey.