

Optimization based LPV-approximation of multi-model systems

Daniel Petersson, Johan Löfberg

Division of Automatic Control

E-mail: petersson@isy.liu.se, johanl@isy.liu.se

2nd March 2010

Report no.: LiTH-ISY-R-2936

Submitted to European Control Conference (ECC) 2009, Budapest, Hungary

Address:

Department of Electrical Engineering

Linköpings universitet

SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET



Abstract

In this paper we have formulated the problem of finding an LPV-approximation to a system as an optimization problem. For this optimization problem we have presented two possible ways to solve this. The problem is posed as a model reduction problem and formulated such that it should try to preserve the input-output behavior of the system. In the two examples in the paper the potential of the new methods are shown. We have also shown the benefits of using model reduction techniques to capture the input-output behavior to obtain accurate low order LPV-approximations.

One method uses SDP-optimization to solve the problem. SDP-optimization has been a hot topic during the last years, but the problem with the SDP method is that it scales badly with the dimension of the problem. Also here it has bilinear constraints which makes the problem really difficult. With the other method we try to use a more general nonlinear approach which seem to be more suitable for this problem. For this method we have also calculated a gradient that can be used to apply a descent or Newton-like method to solve the problem.

Keywords: Optimization, LPV

Optimization based LPV-approximation of multi-model systems

Daniel Petersson and Johan Löfberg

Abstract—In this paper we have formulated the problem of finding an LPV-approximation to a system as an optimization problem. For this optimization problem we have presented two possible ways to solve this. The problem is posed as a model reduction problem and formulated such that it should try to preserve the input-output behavior of the system. In the two examples in the paper the potential of the new methods are shown. We have also shown the benefits of using model reduction techniques to capture the input-output behavior to obtain accurate low order LPV-approximations.

One method uses SDP-optimization to solve the problem. SDP-optimization has been a hot topic during the last years, but the problem with the SDP method is that it scales badly with the dimension of the problem. Also here it has bilinear constraints which makes the problem really difficult. With the other method we try to use a more general nonlinear approach which seem to be more suitable for this problem. For this method we have also calculated a gradient that can be used to apply a descent or Newton-like method to solve the problem.

I. INTRODUCTION

The behaviour of a linear parameter varying (LPV) system can be described by

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

where x are the states, u and y are the input and output signals and $p(t)$ is the vector of model parameters. In flight control applications, the components of $p(t)$ are often model parameters for instance mass, position of center of gravity, various aerodynamic coefficients, but can also include state dependent parameters such as altitude and velocity specifying current flight conditions. In this paper we will be interested in the case when the parameters vary very slowly and not take into account the time dependence of the parameters. Some advanced robustness analysis methods, *e.g.* μ -analysis, see *e.g.* [1], requires a conversion of the LPV-system into a linear fractional representation (LFR). For this purpose, it is necessary that the parametric matrices $\mathbf{A}(p)$, $\mathbf{B}(p)$, $\mathbf{C}(p)$, $\mathbf{D}(p)$ of the LPV-system depend rationally on the parameters in p . This requirement is often not fulfilled for LPV-models directly generated from a nonlinear system description, either due to presence of non-rational nonlinear parametric expressions or tabulated data in the model. In both cases, rational approximations must be used to obtain a suitable system.

Financial support from the European Commission under Contract No AST5-CT-2006-030768-COFCLUO is gratefully acknowledged

D. Petersson and J. Löfberg are with Division of Automatic Control, Department of Electrical Engineering, Linköpings universitet, SE-581 83 Sweden {petersson@isy.liu.se, johanl@isy.liu.se}

LPV-models are often generated by starting from a multi-model system in state space form

$$G_{p^{(i)}} := \left[\begin{array}{c|c} \mathbf{A}_{p^{(i)}} & \mathbf{B}_{p^{(i)}} \\ \hline \mathbf{C}_{p^{(i)}} & \mathbf{D}_{p^{(i)}} \end{array} \right]$$

where each system $G_{p^{(i)}}$ corresponds to a parameter vector sampled in the point $p^{(i)}$, for $i = 1, \dots, N$. The modelling goal is to approximate this multi-model system with a single LPV-system

$$\hat{G}(p) = \left[\begin{array}{c|c} \hat{\mathbf{A}}(p) & \hat{\mathbf{B}}(p) \\ \hline \hat{\mathbf{C}}(p) & \hat{\mathbf{D}}(p) \end{array} \right]$$

whose state space realization depends rationally on p . A frequently used method today is the element-wise approximation method *e.g.* see [2]. This method interpolates the elements in the system matrices individually with rational or polynomial functions. The problem with this approach is that it is naive and does not take system properties into account, additionally a prerequisite for the application of this method is that all dimensions of the matrices in the multi-model system are the same and the matrices correspond to the same ordering of state, input and output variables.

Other methods that also use interpolation are *e.g.* [3], [4], but they transform the models into canonical state-space forms before doing the interpolation. Also there are input-output relation based methods that for example uses linear regression [5] or nonlinear optimization [6]. An excellent survey over existing methods can be found in [7].

In this paper we first formulate an optimization problem to find an LPV-system that tries to approximate the multi-model system and capture the input-output behavior using the \mathcal{H}_2 -norm. Then we present two approaches to solve the optimization problem that arise, and compare them in terms of computational results.

II. THE OPTIMIZATION PROBLEM

In this section we formulate the optimization problem that arise when we try to approximate a multi-model system sampled at different p -values with an LPV-system. The optimization problem is formulated such that the sought system should capture important properties of the sampled systems. The objective is to minimize the error between the true systems and the sought LPV-system in the sampled points in the \mathcal{H}_2 -norm, *i.e.* we formulate the optimization problem

$$\min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}} \sum_i \left\| G_{p^{(i)}} - \hat{G}(p^{(i)}) \right\|_{\mathcal{H}_2}^2 \quad (1)$$

where

$$G_{p^{(i)}} = \left[\begin{array}{c|c} \mathbf{A}_{p^{(i)}} & \mathbf{B}_{p^{(i)}} \\ \hline \mathbf{C}_{p^{(i)}} & \mathbf{D}_{p^{(i)}} \end{array} \right]$$

are the sampled (true) models and

$$\hat{G}(p) = \left[\begin{array}{c|c} \hat{\mathbf{A}}(p) & \hat{\mathbf{B}}(p) \\ \hline \hat{\mathbf{C}}(p) & \hat{\mathbf{D}}(p) \end{array} \right]$$

is the resulting LPV-model depending on the parameter p . Define the error systems as

$$E_{p^{(i)}} = G_{p^{(i)}} - \hat{G}(p^{(i)}).$$

We start by looking at the models in one sample point. Later we will generalize this to the case where we have multiple models. The error system can be realized in state-space form as

$$E = \left[\begin{array}{c|c} \mathbf{A}_e & \mathbf{B}_e \\ \hline \mathbf{C}_e & \mathbf{D}_e \end{array} \right] := \left[\begin{array}{c|c} \left(\begin{array}{cc} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{A}} \end{array} \right) & \left(\begin{array}{c} \mathbf{B} \\ \hat{\mathbf{B}} \end{array} \right) \\ \hline (\mathbf{C} \quad -\hat{\mathbf{C}}) & (\mathbf{D} - \hat{\mathbf{D}}) \end{array} \right]. \quad (2)$$

This realization of the error system will later prove beneficial in rewriting the optimization problem. Notice that the \mathcal{H}_2 -norm is unbounded if the system is not strictly proper, *i.e.* we can not have a direct term. Therefore we assume that $\mathbf{D} = \mathbf{0}$ as well as $\hat{\mathbf{D}} = \mathbf{0}$. Additionally, for simplicity we will only work with polynomial dependence of the parameters in this paper.

By definition the \mathcal{H}_2 -norm of a system, see [1], is

$$\|E\|_{\mathcal{H}_2}^2 = \text{tr} \frac{1}{2\pi} \int_{-\infty}^{\infty} E(j\omega)E(j\omega)^H d\omega \quad (3)$$

To calculate the cost function efficiently it is necessary to rewrite (3) to a numerically more suitable form. This can be done using the Gramians for the system, see [1]. The observability and controllability Gramians, \mathbf{P}_e and \mathbf{Q}_e respectively, for the error system E are defined as

$$\mathbf{P}_e := \int_0^{\infty} e^{\mathbf{A}_e t} \mathbf{B}_e (e^{\mathbf{A}_e t} \mathbf{B}_e)^T dt, \quad (4a)$$

$$\mathbf{Q}_e := \int_0^{\infty} (\mathbf{C}_e e^{\mathbf{A}_e t})^T \mathbf{C}_e e^{\mathbf{A}_e t} dt \quad (4b)$$

where \mathbf{P}_e and \mathbf{Q}_e satisfy the Lyapunov equations, see [1],

$$\mathbf{A}_e \mathbf{P}_e + \mathbf{P}_e \mathbf{A}_e^T + \mathbf{B}_e \mathbf{B}_e^T = \mathbf{0}, \quad (5a)$$

$$\mathbf{A}_e^T \mathbf{Q}_e + \mathbf{Q}_e \mathbf{A}_e + \mathbf{C}_e^T \mathbf{C}_e = \mathbf{0}. \quad (5b)$$

By using (4), (5) and Parseval's identity it is possible to rewrite (3) as

$$\|E\|_{\mathcal{H}_2}^2 = \text{tr} \mathbf{B}_e^T \mathbf{Q}_e \mathbf{B}_e = \text{tr} \mathbf{C}_e \mathbf{P}_e \mathbf{C}_e^T. \quad (6)$$

A. Model class

This optimization problem is based on model reduction techniques, and tries to capture the input-output behavior of the system. A benefit with this is that, even though some elements can depend non-polynomially on the parameters, it can make use of the fact that the realization is non-unique.

The system $\hat{G} = \left[\begin{array}{c|c} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hline \hat{\mathbf{C}} & \mathbf{0} \end{array} \right]$ has the same transfer function and input-output behavior as $\hat{G}_{\mathbf{T}}$, with

$$\hat{G}_{\mathbf{T}} = \left[\begin{array}{c|c} \hat{\mathbf{A}}_{\mathbf{T}} & \hat{\mathbf{B}}_{\mathbf{T}} \\ \hline \hat{\mathbf{C}}_{\mathbf{T}} & \mathbf{0} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{T}^{-1} \hat{\mathbf{A}} \mathbf{T} & \mathbf{T}^{-1} \hat{\mathbf{B}} \\ \hline \hat{\mathbf{C}} \mathbf{T} & \mathbf{0} \end{array} \right],$$

where \mathbf{T} is a non-singular transformation matrix. This means that for every model $G_{p^{(i)}}$ we are not only limited to find the best approximation between $G_{p^{(i)}} = \left[\begin{array}{c|c} \mathbf{A}_{p^{(i)}} & \mathbf{B}_{p^{(i)}} \\ \hline \mathbf{C}_{p^{(i)}} & \mathbf{0} \end{array} \right]$ and $G(p^{(i)}) = \left[\begin{array}{c|c} \hat{\mathbf{A}}(p^{(i)}) & \hat{\mathbf{B}}(p^{(i)}) \\ \hline \hat{\mathbf{C}}(p^{(i)}) & \mathbf{0} \end{array} \right]$, but to find the match between $G_{p^{(i)}} = \left[\begin{array}{c|c} \mathbf{T}_{p^{(i)}}^{-1} \mathbf{A}_{p^{(i)}} \mathbf{T}_{p^{(i)}} & \mathbf{T}_{p^{(i)}}^{-1} \mathbf{B}_{p^{(i)}} \\ \hline \mathbf{C}_{p^{(i)}} \mathbf{T}_{p^{(i)}} & \mathbf{0} \end{array} \right]$ and $G(p^{(i)}) = \left[\begin{array}{c|c} \hat{\mathbf{A}}(p^{(i)}) & \hat{\mathbf{B}}(p^{(i)}) \\ \hline \hat{\mathbf{C}}(p^{(i)}) & \mathbf{0} \end{array} \right]$. This will be illustrated in examples later by showing that for a given system it is possible to find low order LPV-system that approximates the input-output behavior well even though the original system can have a complicated dependence of the parameters.

B. Constraints

In some cases it can be necessary to preserve the physical interpretation of the states, *i.e.* try to mimic the functions in the elements of the system matrices, or to introduce or preserve some structure in the system matrices. Introducing structure to the system matrices is straightforward, just choose which elements you which to optimize over and leave the other elements constant. To mimic a function in a specific element, introduce linear constraints such that the function values in an element does not differ from a given value in that element in the models more than a predetermined value, *i.e.* do an interpolation of the samples from the element at the same time as minimizing the \mathcal{H}_2 -norm for the error system.

III. METHOD 1: GENERAL NONLINEAR OPTIMIZATION

In this algorithm, we try to solve the optimization problem by simply addressing it as a general nonlinear optimization problem.

We see that, because of the realization (2) of E and equations (5), if we partition the Gramians \mathbf{P}_e and \mathbf{Q}_e as

$$\mathbf{P}_e = \begin{pmatrix} \mathbf{P} & \mathbf{X} \\ \mathbf{X}^T & \hat{\mathbf{P}} \end{pmatrix}, \quad \mathbf{Q}_e = \begin{pmatrix} \mathbf{Q} & \mathbf{Y} \\ \mathbf{Y}^T & \hat{\mathbf{Q}} \end{pmatrix}$$

then \mathbf{P} and \mathbf{Q} will satisfy the Lyapunov equations for the controllability and the observability Gramians for the true system, and $\hat{\mathbf{P}}$ and $\hat{\mathbf{Q}}$ will satisfy the Lyapunov equations for the controllability and the observability Gramians for the sought system. $\hat{\mathbf{P}}, \hat{\mathbf{Q}}, \mathbf{X}$ and \mathbf{Y} satisfy, due to (5), the Sylvester and Lyapunov equations

$$\mathbf{A} \mathbf{X} + \mathbf{X} \hat{\mathbf{A}}^T + \mathbf{B} \hat{\mathbf{B}}^T = \mathbf{0}, \quad \hat{\mathbf{A}} \hat{\mathbf{P}} + \hat{\mathbf{P}} \hat{\mathbf{A}}^T + \hat{\mathbf{B}} \hat{\mathbf{B}}^T = \mathbf{0} \quad (7a)$$

$$\mathbf{A}^T \mathbf{Y} + \mathbf{Y} \hat{\mathbf{A}} - \mathbf{C}^T \hat{\mathbf{C}} = \mathbf{0}, \quad \hat{\mathbf{A}}^T \hat{\mathbf{Q}} + \hat{\mathbf{Q}} \hat{\mathbf{A}} + \hat{\mathbf{C}}^T \hat{\mathbf{C}} = \mathbf{0}. \quad (7b)$$

With the partitioning of \mathbf{P}_e and \mathbf{Q}_e it is possible to rewrite (6) as

$$\|E\|_{\mathcal{H}_2}^2 = \text{tr} \left(\mathbf{B}^T \mathbf{Q} \mathbf{B} + 2\mathbf{B}^T \mathbf{Y} \hat{\mathbf{B}} + \hat{\mathbf{B}}^T \hat{\mathbf{Q}} \hat{\mathbf{B}} \right) \quad (8a)$$

$$\|E\|_{\mathcal{H}_2}^2 = \text{tr} \left(\mathbf{C} \mathbf{P} \mathbf{C}^T - 2\mathbf{C} \mathbf{X} \hat{\mathbf{C}}^T + \hat{\mathbf{C}} \hat{\mathbf{P}} \hat{\mathbf{C}}^T \right) \quad (8b)$$

which is now expressed in the system matrices for the true system and the sought system and partitions of the Gramians. It is now straightforward to express this for the more general case when we have multiple models.

$$\sum_i \|E_{p^{(i)}}\|_{\mathcal{H}_2}^2 = \sum_i \text{tr} \left(\mathbf{B}_{p^{(i)}}^T \mathbf{Q}_i \mathbf{B}_{p^{(i)}} + 2\mathbf{B}_{p^{(i)}}^T \mathbf{Y}_i \hat{\mathbf{B}}(p^{(i)}) + \hat{\mathbf{B}}(p^{(i)})^T \hat{\mathbf{Q}}_i \hat{\mathbf{B}}(p^{(i)}) \right) \quad (9)$$

The optimization problem can now be written as

$$\min_{\hat{\mathbf{A}}_{p^k}, \hat{\mathbf{B}}_{p^k}, \hat{\mathbf{C}}_{p^k}} \sum_i \text{tr} \left(\mathbf{B}_{p^{(i)}}^T \mathbf{Q}_i \mathbf{B}_{p^{(i)}} + 2\mathbf{B}_{p^{(i)}}^T \mathbf{Y}_i \hat{\mathbf{B}}(p^{(i)}) + \hat{\mathbf{B}}(p^{(i)})^T \hat{\mathbf{Q}}_i \hat{\mathbf{B}}(p^{(i)}) \right)$$

where the system matrices are expressed as $\hat{\mathbf{B}}(p) = \hat{\mathbf{B}}_{p^0} + \hat{\mathbf{B}}_{p^1} p + \hat{\mathbf{B}}_{p^2} p^2 + \dots + \hat{\mathbf{B}}_{p^k} p^k$.

A. Gradient

The equations (8) are differentiable in the system matrices for the approximated system, *i.e.* it is possible to calculate the gradient w.r.t. $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ and obtain a closed-form expression (see [8], [9]). The calculations will be shown in this section.

Definition 1: The gradient of a real scalar function $f(\mathbf{X})$ of a real matrix valued variable $\mathbf{X} \in \mathbb{R}^{n \times m}$ is the real matrix $\nabla_{\mathbf{X}} f(\mathbf{X}) \in \mathbb{R}^{n \times m}$ defined by

$$[\nabla_{\mathbf{X}} f(\mathbf{X})]_{i,j} = \frac{d}{d\mathbf{X}_{i,j}} f(\mathbf{X}), \quad i = 1, \dots, n, \quad j = 1, \dots, m.$$

Using the definition we get the first order expansion

$$f(\mathbf{X} + \Delta) = f(\mathbf{X}) + \langle \nabla_{\mathbf{X}} f(\mathbf{X}), \Delta \rangle + \mathcal{O}(\|\Delta\|^2),$$

where $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr} \mathbf{A}^T \mathbf{B}$.

To simplify the calculations of the gradient we need the following Lemma (see [10]).

Lemma 1: If \mathbf{M} and \mathbf{N} satisfies the following Sylvester equations

$$\mathbf{A} \mathbf{M} + \mathbf{M} \mathbf{B} + \mathbf{C} = \mathbf{0}, \quad \mathbf{N} \mathbf{A} + \mathbf{B} \mathbf{N} + \mathbf{D} = \mathbf{0}$$

then $\text{tr} \mathbf{C} \mathbf{N} = \text{tr} \mathbf{D} \mathbf{M}$.

Now start by calculating the gradient with respect to $\hat{\mathbf{A}}$, *i.e.* $\nabla_{\hat{\mathbf{A}}} \|E\|_{\mathcal{H}_2}^2$. $\hat{\mathbf{Q}}$ and \mathbf{Y} depend on $\hat{\mathbf{A}}$ so a first order perturbation in (8a), $\Delta_{\|E\|_{\mathcal{H}_2}^2}$, with respect to $\Delta_{\hat{\mathbf{A}}}$ is

$$\Delta_{\|E\|_{\mathcal{H}_2}^2} = \text{tr} \left(2\hat{\mathbf{B}} \mathbf{B}^T \Delta_{\mathbf{Y}} + \hat{\mathbf{B}} \hat{\mathbf{B}}^T \Delta_{\hat{\mathbf{Q}}} \right) \quad (10)$$

where $\Delta_{\mathbf{Y}}$ and $\Delta_{\hat{\mathbf{Q}}}$ depend on $\Delta_{\hat{\mathbf{A}}}$ via

$$\begin{aligned} \mathbf{A}^T \Delta_{\mathbf{Y}} + \Delta_{\mathbf{Y}} \hat{\mathbf{A}} + \mathbf{Y} \Delta_{\hat{\mathbf{A}}} &= \mathbf{0}, \\ \hat{\mathbf{A}}^T \Delta_{\hat{\mathbf{Q}}} + \Delta_{\hat{\mathbf{Q}}} \hat{\mathbf{A}} + \Delta_{\hat{\mathbf{A}}}^T \hat{\mathbf{Q}} + \hat{\mathbf{Q}} \Delta_{\hat{\mathbf{A}}} &= \mathbf{0} \end{aligned} \quad (11)$$

Applying Lemma 1 on (7a) and (11) entails

$$\begin{aligned} \text{tr} \hat{\mathbf{B}} \mathbf{B}^T \Delta_{\mathbf{Y}} &= \text{tr} \mathbf{X}^T \mathbf{Y} \Delta_{\hat{\mathbf{A}}}, \\ \text{tr} \hat{\mathbf{B}} \hat{\mathbf{B}}^T \Delta_{\hat{\mathbf{Q}}} &= \text{tr} \hat{\mathbf{P}} (\Delta_{\hat{\mathbf{A}}}^T \hat{\mathbf{Q}} + \hat{\mathbf{Q}} \Delta_{\hat{\mathbf{A}}}). \end{aligned}$$

Inserting this in (10) entails

$$\begin{aligned} \Delta_{\|E\|_{\mathcal{H}_2}^2} &= \text{tr} \left(2\mathbf{X}^T \mathbf{Y} \Delta_{\hat{\mathbf{A}}} + \hat{\mathbf{P}} (\Delta_{\hat{\mathbf{A}}}^T \hat{\mathbf{Q}} + \hat{\mathbf{Q}} \Delta_{\hat{\mathbf{A}}}) \right) = \\ &= \left\langle 2 \left(\hat{\mathbf{Q}} \hat{\mathbf{P}} + \mathbf{Y}^T \mathbf{X} \right), \Delta_{\hat{\mathbf{A}}} \right\rangle. \end{aligned}$$

It follows that $\nabla_{\hat{\mathbf{A}}} \|E\|_{\mathcal{H}_2}^2 = 2 \left(\hat{\mathbf{Q}} \hat{\mathbf{P}} + \mathbf{Y}^T \mathbf{X} \right)$. Analogously we can calculate the gradients with respect to $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$.

$$\nabla_{\hat{\mathbf{A}}} \|E\|_{\mathcal{H}_2}^2 = 2 \left(\hat{\mathbf{Q}} \hat{\mathbf{P}} + \mathbf{Y}^T \mathbf{X} \right) \quad (12a)$$

$$\nabla_{\hat{\mathbf{B}}} \|E\|_{\mathcal{H}_2}^2 = 2 \left(\hat{\mathbf{Q}} \hat{\mathbf{B}} + \mathbf{Y}^T \mathbf{B} \right) \quad (12b)$$

$$\nabla_{\hat{\mathbf{C}}} \|E\|_{\mathcal{H}_2}^2 = 2 \left(\hat{\mathbf{C}} \hat{\mathbf{P}} - \mathbf{C} \mathbf{X} \right) \quad (12c)$$

If we look at the more general form, with polynomial dependence in the parameters *i.e.* $\hat{\mathbf{A}}(p) = \hat{\mathbf{A}}_{p^0} + \hat{\mathbf{A}}_{p^1} p + \hat{\mathbf{A}}_{p^2} p^2 + \dots + \hat{\mathbf{A}}_{p^k} p^k$, and express the gradient of (9) with respect to the coefficient matrices $\hat{\mathbf{A}}_{p^k}, \hat{\mathbf{B}}_{p^k}, \hat{\mathbf{C}}_{p^k}$.

$$\nabla_{\hat{\mathbf{A}}_{p^k}} \left(\sum_i \|E_{p^{(i)}}\|_{\mathcal{H}_2}^2 \right) = 2 \sum_i (p^{(i)})^k \left(\hat{\mathbf{Q}}_i \hat{\mathbf{P}}_i + \mathbf{Y}_i^T \mathbf{X}_i \right)$$

$$\nabla_{\hat{\mathbf{B}}_{p^k}} \left(\sum_i \|E_{p^{(i)}}\|_{\mathcal{H}_2}^2 \right) = 2 \sum_i (p^{(i)})^k \left(\hat{\mathbf{Q}}_i \hat{\mathbf{B}}_i + \mathbf{Y}_i^T \mathbf{B}_i \right)$$

$$\nabla_{\hat{\mathbf{C}}_{p^k}} \left(\sum_i \|E_{p^{(i)}}\|_{\mathcal{H}_2}^2 \right) = 2 \sum_i (p^{(i)})^k \left(\hat{\mathbf{C}}_i \hat{\mathbf{P}}_i - \mathbf{C}_i \mathbf{X}_i \right)$$

To calculate the cost function (9), one Lyapunov equation (5a) needs to be solved for every i . Crucial to notice is that the extra cost to compute the gradient is merely to solve one additional Lyapunov equation (5b). This extra equation can be solved very effectively if solved simultaneously with (5a), utilizing that we have the same \mathbf{A} matrix in both equations, see [11]. This is however not implemented in the algorithm at the moment.

IV. METHOD 2: SEMIDEFINITE PROGRAMMING

In this algorithm we try to minimize the error system using a semidefinite programming (SDP) approach. With the realization (2) and equations (5) and (6) we rewrite the \mathcal{H}_2 -norm for a system as a minimization problem

$$\begin{aligned} \min_{\mathbf{Q}} \text{tr} \mathbf{B}_e^T \mathbf{Q} \mathbf{B}_e \\ \text{s.t. } \mathbf{A}_e^T \mathbf{Q} + \mathbf{Q} \mathbf{A}_e + \mathbf{C}_e^T \mathbf{C}_e = \mathbf{0}, \quad \mathbf{Q} \succ \mathbf{0} \end{aligned}$$

This can be rewritten, using a Schur complement (see [12]), to get an equivalent problem

$$\begin{aligned} \min_{\gamma, \mathbf{Q}} \gamma \\ \text{s.t. } \begin{pmatrix} \mathbf{Q} \mathbf{A}_e + \mathbf{A}_e^T \mathbf{Q} & \mathbf{C}_e^T \\ \mathbf{C}_e & -\mathbf{I} \end{pmatrix} \prec \mathbf{0}, \quad \begin{pmatrix} \gamma \mathbf{I} & \mathbf{B}_e^T \mathbf{Q} \\ \mathbf{Q} \mathbf{B}_e & \mathbf{Q} \end{pmatrix} \succ \mathbf{0}. \end{aligned} \quad (13)$$

Our objective is to find the system $\hat{G} = \left[\begin{array}{c|c} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hline \hat{\mathbf{C}} & \mathbf{0} \end{array} \right]$ that approximates the system $G = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{0} \end{array} \right]$ well in the \mathcal{H}_2 -norm *i.e.* we also want to minimize over the matrices $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}$.

Since the system E is the augmented system

$$E = \left[\begin{array}{c|c} \mathbf{A}_e & \mathbf{B}_e \\ \hline \mathbf{C}_e & \mathbf{0} \end{array} \right] = \left[\begin{array}{cc|c} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hline \mathbf{C} & -\hat{\mathbf{C}} & \mathbf{0} \end{array} \right]$$

and if we partition \mathbf{Q} as

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22} \end{pmatrix}$$

we can rewrite the minimization problem (13) as a new problem, now also optimizing over the sought system matrices, as

$$\begin{aligned} & \min_{\gamma, \mathbf{Q}, \hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}} \gamma \\ \text{s.t.} & \begin{pmatrix} \mathbf{Q}_{11}\mathbf{A} + \mathbf{A}^T\mathbf{Q}_{11} & \mathbf{Q}_{12}\hat{\mathbf{A}} + \mathbf{A}^T\mathbf{Q}_{12} & \mathbf{C}^T \\ \mathbf{Q}_{12}^T\mathbf{A} + \hat{\mathbf{A}}^T\mathbf{Q}_{12} & \mathbf{Q}_{22}\hat{\mathbf{A}} + \hat{\mathbf{A}}^T\mathbf{Q}_{22} & -\hat{\mathbf{C}}^T \\ \mathbf{C} & -\hat{\mathbf{C}} & -\mathbf{I} \end{pmatrix} < 0, \\ & \begin{pmatrix} \gamma\mathbf{I} & \mathbf{B}^T\mathbf{Q}_{11} + \hat{\mathbf{B}}^T\mathbf{Q}_{12}^T & \mathbf{B}^T\mathbf{Q}_{12} + \hat{\mathbf{B}}^T\mathbf{Q}_{22} \\ \mathbf{Q}_{11}\mathbf{B} + \mathbf{Q}_{12}\hat{\mathbf{B}} & \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{12}^T\mathbf{B} + \mathbf{Q}_{22}\hat{\mathbf{B}} & \mathbf{Q}_{12}^T & \mathbf{Q}_{22} \end{pmatrix} \succ 0. \end{aligned} \quad (14)$$

Generalizing this to the case when we have a true system sampled at different p -values and want to find an LPV-approximation is straightforward. Rewriting (14) again for this problem we get

$$\begin{aligned} & \min_{\gamma_i, \mathbf{Q}_i, \hat{\mathbf{A}}_{p^k}, \hat{\mathbf{B}}_{p^k}, \hat{\mathbf{C}}_{p^k}} \sum_{i=1}^N \gamma_i, \quad k = 1, \dots, L \\ \text{s.t.} & \begin{pmatrix} \mathbf{Q}_{11,i}\mathbf{A}_i + \mathbf{A}_i^T\mathbf{Q}_{11,i} & \mathbf{Q}_{12,i}\hat{\mathbf{A}}_i + \mathbf{A}_i^T\mathbf{Q}_{12,i} & \mathbf{C}_i^T \\ \mathbf{Q}_{12,i}^T\mathbf{A}_i + \hat{\mathbf{A}}_i^T\mathbf{Q}_{12,i} & \mathbf{Q}_{22,i}\hat{\mathbf{A}}_i + \hat{\mathbf{A}}_i^T\mathbf{Q}_{22,i} & -\hat{\mathbf{C}}_i^T \\ \mathbf{C}_i & -\hat{\mathbf{C}}_i & -\mathbf{I} \end{pmatrix} < 0, \\ & \begin{pmatrix} \gamma_i\mathbf{I} & \mathbf{B}_i^T\mathbf{Q}_{11,i} + \hat{\mathbf{B}}_i^T\mathbf{Q}_{12,i}^T & \mathbf{B}_i^T\mathbf{Q}_{12,i} + \hat{\mathbf{B}}_i^T\mathbf{Q}_{22,i} \\ \mathbf{Q}_{11,i}\mathbf{B}_i + \mathbf{Q}_{12,i}\hat{\mathbf{B}}_i & \mathbf{Q}_{11,i} & \mathbf{Q}_{12,i} \\ \mathbf{Q}_{12,i}^T\mathbf{B}_i + \mathbf{Q}_{22,i}\hat{\mathbf{B}}_i & \mathbf{Q}_{12,i}^T & \mathbf{Q}_{22,i} \end{pmatrix} \succ 0 \\ & i = 1, \dots, N \end{aligned} \quad (15)$$

where $\hat{\mathbf{A}}_{p^k}$ is the coefficient matrices in $\hat{\mathbf{A}} = \hat{\mathbf{A}}_{p^0} + \hat{\mathbf{A}}_{p^1}p + \hat{\mathbf{A}}_{p^2}p^2 + \dots + \hat{\mathbf{A}}_{p^k}p^k$ and L is the highest degree of p in $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ or $\hat{\mathbf{C}}$. Looking at (15) we see that it is bilinear in the variables. Generally bilinear SDPs are very hard to solve [13], which is one of the reasons for us to not only look at this method but also on the more general nonlinear approach in section III. To try to solve this a local iterative two-step algorithm can be used, [14]. Start by keeping $\hat{\mathbf{A}}_{p^k}, \hat{\mathbf{B}}_{p^k}$ constant, then solve (15) for $\mathbf{Q}_i, \hat{\mathbf{C}}_{p^k}$. Then keep $\mathbf{Q}_{12,i}, \mathbf{Q}_{22,i}$ constant and solve (15) for $\mathbf{Q}_{11,i}, \hat{\mathbf{A}}_{p^k}, \hat{\mathbf{B}}_{p^k}, \hat{\mathbf{C}}_{p^k}$. Continue doing this until convergence.

V. NUMERICAL EXAMPLES

In this section two small academic examples are presented to show the potential of the new methods and to show the importance of addressing system properties. In these examples we disregard from the time dependence in the parameters and only tries to find an LPV-model that approximates the linear models in a good way.

Example V.1

The system in this example is a balanced realization of the system $G = G_1G_2$ where $G_1 = \frac{\omega_1^2}{s^2+2\omega_1\zeta_1s+\omega_1^2}$ and $G_2 = \frac{\omega_2^2}{s^2+2\omega_2\zeta_2s+\omega_2^2}$ with $\omega_1 = 1, \omega_2 = 3, \zeta_1 = 0.1 + 0.9\delta$ and $\zeta_2 = 0.1 + 0.9(1 - \delta)$ and $\delta \in [0, 1]$. The system was sampled for 30 δ equidistant in $[0, 1]$ *i.e.* we are given 30 linear models with four states. It is interesting to look at this system because one realization of the system has only linear dependence on δ , in fact only two elements of the system matrix \mathbf{A} are linear in δ and all the other matrix elements in $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are constants. But if we do a balanced realization for every sampled model individually, we still have the same systems, only a different realization, this construction is done to show that the methods work even if we have different state-basis in the different models. Now no elements in any of the three system matrices are linear in δ , instead they are highly nonlinear, see Fig. 1. Because of the highly nonlinear dependence on δ in the elements it is very hard to find a good approximation for the system using element-wise polynomial approximation.

If we use the new methods and try to find a low order

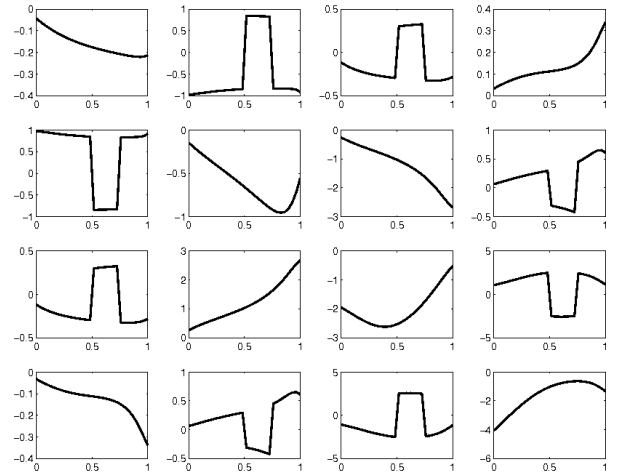


Fig. 1: The elements in the \mathbf{A} -matrix depending on δ .

LPV approximation of the systems, we will find, for the method using the general nonlinear approach, a very good approximation with only linear dependence in δ . But with the method using SDP-optimization, it gets stuck in a local optima, but is, in \mathcal{H}_2 -norm, still better than the element-wise method. To illustrate the correspondence between the sampled models of the true system and the approximated systems their root loci have been plotted in Fig. 2.

	Iter	Time	$\sum_i \ E_i\ _{\mathcal{H}_2}^2$	Degree
Method 1	47	61.0s	$6.27 \cdot 10^{-5}$	1
Method 2	240	1528s	2.18	1
Element-wise	—	0.108s	2.98	9

TABLE I: Numerical results for Example V.1

This example illustrates what was said in section II-A. Even

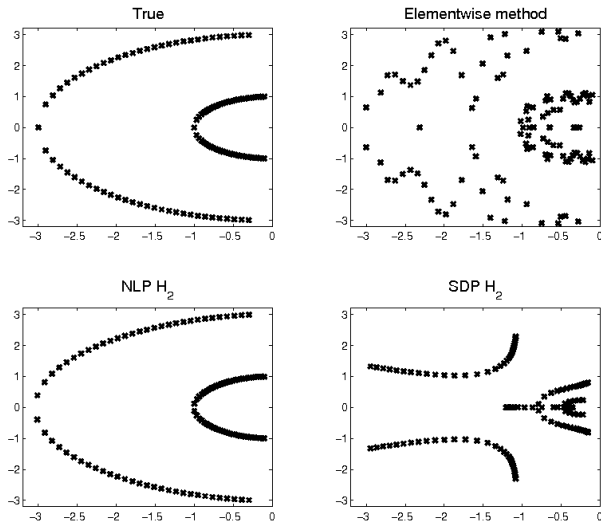


Fig. 2: Root loci for the true system, the systems approximated with the new algorithms using 1st order polynomials and the system approximated with the element-wise method using 9th order polynomials.

though the realization of the system given is non-polynomial in the parameters, we are able to find the underlying system with only linear dependence.

Example V.2

The system we are looking at here has two states and one parameter and is given by

$$G(\delta) = \left[\begin{array}{cc|c} (a_{11} & a_{12}) & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ (a_{21} & a_{22}) & \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \hline (1 & 0) & 0 \end{array} \right]$$

with $a_{11} = 1.45$ and $a_{22} = -0.55$ and $a_{12} = f(\delta) = \frac{\pi}{2} + \arctan(\delta)$ and $a_{21} = g(\delta) = -\frac{1}{f(\delta)}$. This system has its poles in $s = \frac{a_{11}+a_{22}}{2} \pm \sqrt{\frac{(a_{11}-a_{22})^2}{4} - 1} = -0.45$ for all δ . In this example the system matrix $\mathbf{A}(\delta)$ was sampled 30 times equidistant in the interval $\delta \in [-1, 1]$. The important thing to note with this system is that the elements a_{12} and a_{21} are nonlinearly depending on δ , but also coupled. Another important thing to notice is that the poles of this system does not vary with δ , they are constant.

Using the element-wise method and interpolating the elements $a_{12} = f(\delta)$ and $a_{21} = g(\delta)$, the elements in \mathbf{A} that vary, with polynomials of order 8. After doing the element-wise approximation the maximum interpolation error in the two elements are such that the largest deviation is 0.07. With this interpolation we get a root locus where the poles are far from constant (see Fig. 3). If we use our methods, we find a good approximation with only constant matrices (no dependence of δ). The root loci for the systems are in Fig. 3).

This example illustrate the importance to look at the input-output relation and not only the individual elements. We see that when using the element-wise method we do not take

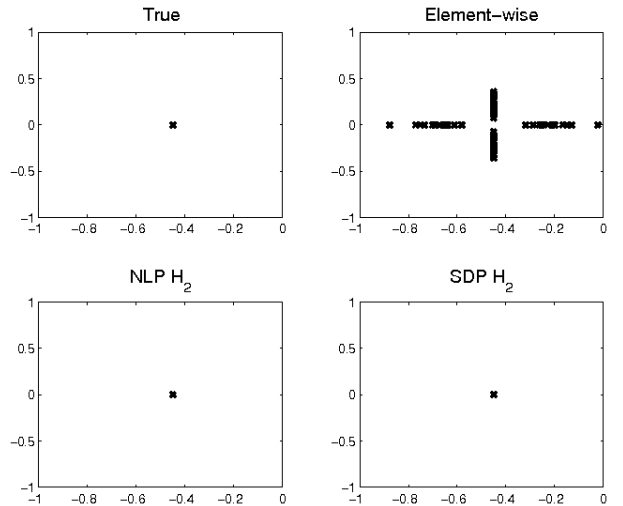


Fig. 3: Root loci for the true system, the systems approximated with the new algorithms using zeroth order polynomials and the system approximated with the element-wise method using 8th order polynomials.

	Iter	Time	$\sum_i \ E_i\ _{\mathcal{H}_2}^2$	Degree
Method 1	65	8.39s	$1.4 \cdot 10^{-13}$	0
Method 2	6	25.2s	$1.0 \cdot 10^{-8}$	0
Element-wise	—	0.031s	$5.4 \cdot 10^4$	8

TABLE II: Numerical results for Example V.2

into account the strong coupling between element a_{12} and a_{21} . But with the new algorithms we find the correct system.

The examples were performed on a Dell Optiplex GX620 with 2GB RAM, Intel P4 640 (3.2 GHz) CPU running under Windows XP SP2 with MATLAB version 7.6 (R2008a). When solving the examples using the general nonlinear approach, the function `fminunc` in MATLAB was used. When solving the examples using the SDP approach, YALMIP version 3 (R20080414) [15] together with SEDUMI version 1.1 [16] were used. Starting points for both the methods were in both examples the system you obtain if you do an element-wise interpolation.

VI. CONCLUSIONS

In this paper we have formulated an optimization problem that tries to approximate a multi-model system with a low order LPV-system. To solve this problem two different methods were developed. In the two examples, we have shown the potential of the new methods and the benefits of using model reduction techniques to capture the input-output behavior to get accurate low order LPV-approximations. As we can see in both table I and II and figure 2 and 3 the element-wise method is fast but gives high order LPV-systems and does not capture the system properties well. The new results shows that though the methods are slower they find better approximations of the systems using lower orders. Method 1 has also shown promising results upon testing it on an flexible aircraft model with 22 states and one parameter, trying to identify an affine LPV-model.

SDP-optimization is frequently used today on related model reduction problems. The problem here is that it introduces auxiliary Lyapunov variables, which make the problem scale bad. Also this problem leads to a nonconvex bilinear SDP-problem which is a very hard problem. That is the reason for developing the other method that uses a more general nonlinear approach. For this problem it was possible to get a closed-form expression for the gradient which makes the method applicable to *e.g.* descent or Newton methods.

REFERENCES

- [1] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [2] A. Varga, G. Looye, D. Moormann, and G. Grübel, "Automated generation of LFT-based parametric uncertainty descriptions from generic aircraft models," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 4, pp. 249–274, 1998.
- [3] M. Steinbuch, R. van de Molengraft, and A. van der Voort, "Experimental modelling and l_{pv} control of a motion system," *American Control Conference, 2003. Proceedings of the 2003*, vol. 2, pp. 1374–1379, 4-6, 2003.
- [4] M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra, "L_{pv} control for a wafer stage: beyond the theoretical solution," *Control Engineering Practice*, vol. 13, no. 2, pp. 231 – 245, 2005.
- [5] X. Wei and L. Del Re, "On persistent excitation for parameter estimation of quasi-LPV systems and its application in modeling of diesel engine torque," in *Proceedings of the 14th IFAC Symposium on System Identification*, 2006, pp. 517–522.
- [6] F. Previdi and M. Lovera, "Identification of a class of non-linear parametrically varying models," *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 1, 2003.
- [7] R. Tóth, "Modeling and Identification of Linear Parameter-Varying Systems—An Orthonormal Basis Function Approach," Ph.D. dissertation, Delft University of Technology, 2008.
- [8] P. Van Dooren, K. Gallivan, and P. Absil, " \mathcal{H}_2 -optimal model reduction of MIMO systems," *Appl. Math. Lett.*, 2008.
- [9] D. Wilson, "Optimum solution of model reduction problem," *Proc. Inst. Elec. Eng.*, vol. 117, pp. 1161 – 1165, 1970.
- [10] W.-Y. Yan and J. Lam, "An approximate approach to \mathcal{H}_2 optimal model reduction," *Automatic Control, IEEE Transactions on*, vol. 44, no. 7, pp. 1341–1358, Jul 1999.
- [11] P. Benner, J. M. Claver, and E. S. Quintana-orti, "Efficient solution of coupled lyapunov equations via matrix sign function iteration," in *Proc. 3 rd Portuguese Conf. on Automatic Control CONTROL'98, Coimbra*, 1998, pp. 205–210.
- [12] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, ser. Studies in Applied Mathematics. SIAM, 1994, vol. 15.
- [13] M. Mesbahi, G. Papavassilopoulos, and M. Safonov, "Matrix cones, complementarity problems, and the bilinear matrix inequality," *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 3, pp. 3102–3107 vol.3, Dec 1995.
- [14] A. Helmersson, "Model reduction using l_{mis}," *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, vol. 4, pp. 3217–3222 vol.4, Dec 1994.
- [15] J. Löfberg, "YALMIP: a toolbox for modeling and optimization in MATLAB," *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pp. 284–289, 2004.
- [16] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11-12, pp. 625–653, 1999.



Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering

Datum
Date

2010-03-02

Språk
Language

- Svenska/Swedish
 Engelska/English

Rapporttyp
Report category

- Licentiatavhandling
 Examensarbete
 C-uppsats
 D-uppsats
 Övrig rapport

ISBN

—

ISRN

—

Serietitel och serienummer

Title of series, numbering

ISSN

1400-3902

URL för elektronisk version

<http://www.control.isy.liu.se>

LiTH-ISY-R-2936

Titel Optimization based LPV-approximation of multi-model systems
Title

Författare Daniel Petersson, Johan Löfberg
Author

Sammanfattning
Abstract

In this paper we have formulated the problem of finding an LPV-approximation to a system as an optimization problem. For this optimization problem we have presented two possible ways to solve this. The problem is posed as a model reduction problem and formulated such that it should try to preserve the input-output behavior of the system. In the two examples in the paper the potential of the new methods are shown. We have also shown the benefits of using model reduction techniques to capture the input-output behavior to obtain accurate low order LPV-approximations.

One method uses SDP-optimization to solve the problem. SDP-optimization has been a hot topic during the last years, but the problem with the SDP method is that it scales badly with the dimension of the problem. Also here it has bilinear constraints which makes the problem really difficult. With the other method we try to use a more general nonlinear approach which seem to be more suitable for this problem. For this method we have also calculated a gradient that can be used to apply a descent or Newton-like method to solve the problem.

Nyckelord
Keywords Optimization, LPV