Optimisation-based modelling of LPV systems using an ℓ_1 -objective

Daniel Petersson and Johan Löfberg

Linköping University Post Print

N.B.: When citing this work, cite the original article.

This is an electronic version of an article published in:

Daniel Petersson and Johan Löfberg, Optimisation-based modelling of LPV systems using an ℓ_1 -objective, 2014, International Journal of Control, (87), 8, 1536-1548.

International Journal of Control is available online at informaworldTM:

<http://dx.doi.org/10.1080/00207179.2013.878477>

Copyright: Taylor & Francis: STM, Behavioural Science and Public Health Titles

<http://www.tandf.co.uk/journals/default.asp>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-107437>

RESEARCH ARTICLE

Optimization-Based Modeling of LPV Systems using an $\mathcal{H}_2$ -Objective

Daniel Petersson^{a†} and Johan Löfberg^a

^a*Division of Automatic Control,
Department of Electrical Engineering,
Linköpings universitet, SE-581 83 Sweden*

(Received 00 Month 200x; final version received 00 Month 200x)

A method to identify LPV-models through minimization of an $\mathcal{H}_2$ -norm objective is presented. The method uses a direct nonlinear programming approach to a non-convex problem. The reason to use $\mathcal{H}_2$ -norm is twofold. To begin with, it is a well known and widely used system norm and secondly, the cost functions described in this paper becomes differentiable when using the $\mathcal{H}_2$ -norm. This enables us to have a measure of first order optimality and to use standard quasi-Newton solvers to solve the problem. The specific structure of the problem is utilized in great detail to compute cost functions and gradients efficiently. Additionally, a regularized version of the method, which also has a nice computational structure, is presented. The regularized version is shown to have an interesting interpretation with connections to worst-case approaches.

1 Introduction

In the last decades intensive research has been carried out on linear parameter varying models (LPV-models), see e.g., Rugh and Shamma (2000), Leith and Leithead (2000), Tóth (2008), Lovera et al. (2011) or Mohammadpour and Scherer (2012). An important reason for this interest is that it is a powerful tool for modeling and analysis of nonlinear systems, such as aircrafts (see Marcos and Balas (2004)) or wafer stages (see Wassink et al. (2005)). Some advanced robustness analysis methods, such as IQC-analysis and μ -analysis, see e.g., Megretski and Rantzer (1997), Zhou et al. (1996), require a conversion of the LPV-model into a linear fractional representation (LFR), see e.g., Zhou et al. (1996). For this to be possible it is necessary that the parametric matrices $\mathbf{A}(\mathbf{p})$, $\mathbf{B}(\mathbf{p})$, $\mathbf{C}(\mathbf{p})$ and $\mathbf{D}(\mathbf{p})$ of the LPV-model are rational in $\mathbf{p}$. This requirement is often violated in LPV-models generated directly from a non-fractional model description, either due to presence of non-fractional parametric expressions or tabulated data in the model. In both cases, rational approximations must be used to obtain a suitable model. This motivates a method that both can approximate a nonlinear model with an LPV-model and approximate a complex LPV-model with a less complex one.

LPV-models can be described by linear differential equations whose coefficients depend on scheduling parameters,

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t), \quad (1)$$

$$\mathbf{y}(t) = \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{D}(\mathbf{p}(t))\mathbf{u}(t), \quad (2)$$

where $\mathbf{x}(t)$ is the state, $\mathbf{u}(t)$ and $\mathbf{y}(t)$ are the input and output signals and $\mathbf{p}(t)$ is the vector of scheduling parameters. For example, in flight control applications, the components of $\mathbf{p}(t)$ are typically mass, position of centre of gravity and various aerodynamic coefficients, but can

[†] Corresponding author. Email: petersson@isy.liu.se

also include state dependent parameters such as altitude and velocity, specifying current flight conditions.

Generation of LPV-models can simplistically be divided into two main families of methods, global methods (see e.g., Nemani et al. (1995), Lee and Poolla (1999), Bamieh and Giarre (2002), Felici et al. (2007), Tóth (2008)) and local methods (see e.g., Steinbuch et al. (2003), Wassink et al. (2005), Lovera and Mercere (2007), De Caigny et al. (2011), Pfifer and Hecker (2008), De Caigny et al. (2012)). An excellent survey of existing methods can be found in Tóth (2008).

1.1 Global Methods

In the global approach a global identification experiment is performed by exciting the system, while the scheduling parameters change the dynamics of the system. An advantage with this approach of generating LPV-models is that it is also possible to capture the rate of change of the parameters and how they can vary between different operating points. However, one drawback is that it is sometimes, for example in flight applications, not possible to perform such an experiment.

1.2 Local Methods

In the local approach, the LPV-models are generated by interpolating, or in some other way combining, several local linear time invariant models (LTI-models). The local models can for example have been identified using a set of measurements for every local model, for which there exists several methods, see e.g., Ljung (1999), or by linearizing a nonlinear model in different operating points. In this family of methods it is assumed that the system can operate at different fixed operating points, where the scheduling parameters are “frozen”. There are of course systems where this is not possible and where this family of methods is not applicable, requiring the use of global methods. Another drawback with this family of methods is that it does not take time variations of the scheduling parameters into account, thus limiting local methods to systems where the scheduling parameters vary slowly in time, which is a commonly used assumption in gain scheduling, see Shamma and Athans (1992). An excellent paper that explains the pitfalls of interpolation is Tóth et al. (2007). There are several different variants of methods that use the local approach and the method presented in this paper is one of them. A common drawback of many of the local methods is that they need the local models to be given in the same state space basis, see e.g., Pfifer and Hecker (2008). Some methods have remedies by transforming the systems to a basis that encourage interpolation, usually some canonical form, see e.g., Steinbuch et al. (2003). However, they can suffer from bad numerics in that form. In De Caigny et al. (2012) they solve this problem by fixing one of the given models a reference model and transforming the other models to state space bases that are consistent with the reference model.

In this paper we develop a local method that helps us to model LPV-models. The method uses an approach that tries to preserve the input-output relations from the given model in the resulting LPV-model. This is done by minimizing the sum of the $\mathcal{H}_2$ -norms of the difference between the given models and a parametrized LPV-model. When developing the method we concentrate on the computational efficiency. To handle uncertainty in the data we suggest a problem specific regularization as a proxy for robust optimization.

2 Generation of LPV State Space Models

Our goal with the method proposed in this paper is to try to preserve the input-output relations of the given models. Define $G(\mathbf{p}^{(i)})$ to be a linearized model at $\mathbf{p} = \mathbf{p}^{(i)}$ of a model that we want to approximate with an LPV-model. Ideally the goal would be, given $G(\mathbf{p})$, to find an LPV-model, $\hat{G}(\mathbf{p})$, that is optimal with respect to some global discrepancy measure on the model error, for

instance the following integral

$$\min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}} \int \left\| G(\mathbf{p}) - \hat{G}(\mathbf{p}) \right\| d\mathbf{p}, \quad (3)$$

where

$$\hat{G}(\mathbf{p}) : \begin{cases} \dot{\mathbf{x}}(t) = \hat{\mathbf{A}}(\mathbf{p})\mathbf{x}(t) + \hat{\mathbf{B}}(\mathbf{p})\mathbf{u}(t) \\ \mathbf{y}(t) = \hat{\mathbf{C}}(\mathbf{p})\mathbf{x}(t) + \hat{\mathbf{D}}(\mathbf{p})\mathbf{u}(t) \end{cases} \quad (4)$$

which will be denoted as

$$\hat{G}(\mathbf{p}) : \left[\frac{\hat{\mathbf{A}}(\mathbf{p})}{\hat{\mathbf{C}}(\mathbf{p})} \middle| \frac{\hat{\mathbf{B}}(\mathbf{p})}{\hat{\mathbf{D}}(\mathbf{p})} \right]. \quad (5)$$

This is not always practical or even tractable. In many applications, e.g., flight applications, one often only have a simulation model available or a model that is used for computational fluid dynamic calculations and not an analytical nonlinear model and it is only possible to extract linearized models for discrete values of the scheduling parameters, $\mathbf{p}^{(i)}$. Having this in mind equation (3) is changed into a discrete version

$$\min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}} \sum_i \left\| G_i - \hat{G}_i \right\|, \quad (6)$$

where G_i denotes $G(\mathbf{p}^{(i)})$. A benefit of formulating the problem as proposed is that the optimization will be invariant to state transformations, i.e., the systems G_i do not need to have the same state basis.

Remark 1: Since the method is based on taking the input-output relation into account, the given models do not even have to have the same number of states. In that case it doesn't even exist any base transformation between the models.

The system matrices, $\hat{\mathbf{A}}(\mathbf{p})$, $\hat{\mathbf{B}}(\mathbf{p})$, $\hat{\mathbf{C}}(\mathbf{p})$ and $\hat{\mathbf{D}}(\mathbf{p})$ are taken to be a linear combination of some basis functions $w_k(\mathbf{p})$, e.g., in the polynomial case, monomials. The system matrices in the LPV-model will then depend on $\mathbf{p}$ as

$$\hat{\mathbf{A}}(\mathbf{p}) = \sum_k w_k(\mathbf{p}) \hat{\mathbf{A}}^{(k)} \quad (7a)$$

$$\hat{\mathbf{B}}(\mathbf{p}) = \sum_k w_k(\mathbf{p}) \hat{\mathbf{B}}^{(k)} \quad (7b)$$

$$\hat{\mathbf{C}}(\mathbf{p}) = \sum_k w_k(\mathbf{p}) \hat{\mathbf{C}}^{(k)} \quad (7c)$$

$$\hat{\mathbf{D}}(\mathbf{p}) = \sum_k w_k(\mathbf{p}) \hat{\mathbf{D}}^{(k)}. \quad (7d)$$

Our goal is to find the coefficient matrices $\hat{\mathbf{A}}^{(k)}$, $\hat{\mathbf{B}}^{(k)}$, $\hat{\mathbf{C}}^{(k)}$ and $\hat{\mathbf{D}}^{(k)}$.

The choice of which norm to use has up to this point not been decided. The two most widely used norms in system theory are the $\mathcal{H}_2$ - and $\mathcal{H}_\infty$ -norms, both capturing the input-output relation of the system. The norm that will be used here is the $\mathcal{H}_2$ -norm. The main reason for this choice is the computational efficiency that follows since the the cost function is differentiable with respect to the optimization variables, with readily computed gradients. This will be covered in depth in the following sections.

2.1 LPV-Modeling using the $\mathcal{H}_2$ -Norm

For computational reasons, we work with a squared $\mathcal{H}_2$ -norm, i.e., the optimization problem is formulated as

$$\min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}} \sum_i \left\| G_i - \hat{G}_i \right\|_{\mathcal{H}_2}^2 = \min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}} \sum_i \|E_i\|_{\mathcal{H}_2}^2 = \min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}} V. \quad (8)$$

To study the problem, (8), let us start by looking at the case when we only have one model and later generalize this to the case where we have multiple models.

Define the error systems as

$$E_i = G_i - \hat{G}_i, \quad (9)$$

which can be realized in state space form as

$$E_i : \left[\begin{array}{c|c} \mathbf{A}_{E,i} & \mathbf{B}_{E,i} \\ \hline \mathbf{C}_{E,i} & \mathbf{D}_{E,i} \end{array} \right] = \left[\begin{array}{c|c} \left(\begin{array}{cc} \mathbf{A}_i & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{A}}_i \end{array} \right) & \left(\begin{array}{c} \mathbf{B}_i \\ \hat{\mathbf{B}}_i \end{array} \right) \\ \hline (\mathbf{C}_i - \hat{\mathbf{C}}_i) & \mathbf{D}_i - \hat{\mathbf{D}}_i \end{array} \right]. \quad (10)$$

This realization of the error system will later prove beneficial when rewriting the optimization problem. Notice that for a continuous-time model the $\mathcal{H}_2$ -norm is unbounded if the model is not strictly proper, i.e., $\mathbf{D}_i = \hat{\mathbf{D}}_i$ is needed for all models (which is trivially true in the case when both $\mathbf{D}_i = \mathbf{0}$ and $\hat{\mathbf{D}}_i = \mathbf{0}$). The problem of finding $\hat{\mathbf{D}}(\mathbf{p})$ can thus be seen as a separate problem which is not addressed here. Throughout the paper it is assumed that the models we are given are stable; otherwise the $\mathcal{H}_2$ -norm is not defined.

The $\mathcal{H}_2$ -norm of the error system can be computed as

$$\|E_i\|_{\mathcal{H}_2}^2 = \text{tr } \mathbf{B}_{E,i}^T \mathbf{Q}_{E,i} \mathbf{B}_{E,i} = \text{tr } \mathbf{C}_{E,i} \mathbf{P}_{E,i} \mathbf{C}_{E,i}^T, \quad (11)$$

where $\mathbf{Q}_{E,i}$ and $\mathbf{P}_{E,i}$ are the observability and controllability Gramians respectively, for the error system E_i , satisfying the equations

$$\mathbf{A}_{E,i} \mathbf{P}_{E,i} + \mathbf{P}_{E,i} \mathbf{A}_{E,i}^T + \mathbf{B}_{E,i} \mathbf{B}_{E,i}^T = \mathbf{0}, \quad (12a)$$

$$\mathbf{A}_{E,i}^T \mathbf{Q}_{E,i} + \mathbf{Q}_{E,i} \mathbf{A}_{E,i} + \mathbf{C}_{E,i}^T \mathbf{C}_{E,i} = \mathbf{0}. \quad (12b)$$

With the realization (10) of E_i and the partitioning of the Gramians $\mathbf{P}_{E,i}$ and $\mathbf{Q}_{E,i}$ as

$$\mathbf{P}_{E,i} = \begin{pmatrix} \mathbf{P}_i & \mathbf{X}_i \\ \mathbf{X}_i^T & \hat{\mathbf{P}}_i \end{pmatrix}, \quad \mathbf{Q}_{E,i} = \begin{pmatrix} \mathbf{Q}_i & \mathbf{Y}_i \\ \mathbf{Y}_i^T & \hat{\mathbf{Q}}_i \end{pmatrix}, \quad (13)$$

six smaller Sylvester and Lyapunov equations

$$\mathbf{A}_i \mathbf{P}_i + \mathbf{P}_i \mathbf{A}_i^T + \mathbf{B}_i \mathbf{B}_i^T = \mathbf{0}, \quad (14a)$$

$$\mathbf{A}_i \mathbf{X}_i + \mathbf{X}_i \hat{\mathbf{A}}_i^T + \mathbf{B}_i \hat{\mathbf{B}}_i^T = \mathbf{0}, \quad (14b)$$

$$\hat{\mathbf{A}}_i \hat{\mathbf{P}}_i + \hat{\mathbf{P}}_i \hat{\mathbf{A}}_i^T + \hat{\mathbf{B}}_i \hat{\mathbf{B}}_i^T = \mathbf{0}, \quad (14c)$$

$$\mathbf{A}_i^T \mathbf{Q}_i + \mathbf{Q}_i \mathbf{A}_i + \mathbf{C}_i^T \mathbf{C}_i = \mathbf{0}, \quad (14d)$$

$$\mathbf{A}_i^T \mathbf{Y}_i + \mathbf{Y}_i \hat{\mathbf{A}}_i - \mathbf{C}_i^T \hat{\mathbf{C}}_i = \mathbf{0}, \quad (14e)$$

$$\hat{\mathbf{A}}_i^T \hat{\mathbf{Q}}_i + \hat{\mathbf{Q}}_i \hat{\mathbf{A}}_i + \hat{\mathbf{C}}_i^T \hat{\mathbf{C}}_i = \mathbf{0}, \quad (14f)$$

are obtained.

Note that $\mathbf{P}_i$ and $\mathbf{Q}_i$ satisfy the Lyapunov equations for the controllability and the observability Gramians for the given system, and $\hat{\mathbf{P}}_i$ and $\hat{\mathbf{Q}}_i$ satisfy the Lyapunov equations for the controllability and the observability Gramians for the sought system. With the partitioning (13) of $\mathbf{P}_{E,i}$ and $\mathbf{Q}_{E,i}$ it is possible to rewrite equation (11), as

$$\|E_i\|_{\mathcal{H}_2}^2 = \text{tr} \left(\mathbf{B}_i^T \mathbf{Q}_i \mathbf{B}_i + 2\mathbf{B}_i^T \mathbf{Y}_i \hat{\mathbf{B}}_i + \hat{\mathbf{B}}_i^T \hat{\mathbf{Q}}_i \hat{\mathbf{B}}_i \right), \quad (15a)$$

$$\|E_i\|_{\mathcal{H}_2}^2 = \text{tr} \left(\mathbf{C}_i \mathbf{P}_i \mathbf{C}_i^T - 2\mathbf{C}_i \mathbf{X}_i \hat{\mathbf{C}}_i^T + \hat{\mathbf{C}}_i \hat{\mathbf{P}}_i \hat{\mathbf{C}}_i^T \right). \quad (15b)$$

The equations in (15) are equivalent and can both be used to compute the squared $\mathcal{H}_2$ -norm of the error system.

One major drawback with this approach though, is that the optimization problem (8), in the general case, is non-convex. The non-convexity of the problem is of course problematic and with it the initialization of the problem becomes important, which will be addressed in Section 3. In the examples in Section 4 we however see that the method is able to find good model approximations despite the non-convexity.

It is now straightforward to express the optimization problem (8) in the system matrices and solutions to Lyapunov and Sylvester equations, using (15), as

$$\min_{\hat{\mathbf{A}}^{(k)}, \hat{\mathbf{B}}^{(k)}, \hat{\mathbf{C}}^{(k)}} V, \quad (16)$$

where

$$V = \sum_i \|E_i\|_{\mathcal{H}_2}^2 = \sum_i \text{tr} \left(\mathbf{B}_i^T \mathbf{Q}_i \mathbf{B}_i + 2\mathbf{B}_i^T \mathbf{Y}_i \hat{\mathbf{B}}_i + \hat{\mathbf{B}}_i^T \hat{\mathbf{Q}}_i \hat{\mathbf{B}}_i \right) \quad (17a)$$

$$= \sum_i \text{tr} \left(\mathbf{C}_i \mathbf{P}_i \mathbf{C}_i^T - 2\mathbf{C}_i \mathbf{X}_i \hat{\mathbf{C}}_i^T + \hat{\mathbf{C}}_i \hat{\mathbf{P}}_i \hat{\mathbf{C}}_i^T \right), \quad (17b)$$

the system matrices are parametrized as in (7) and $\mathbf{P}_i, \mathbf{Q}_i, \hat{\mathbf{P}}_i, \hat{\mathbf{Q}}_i, \mathbf{X}_i$ and $\mathbf{Y}_i$ satisfy the equations in (14).

The cost function (17) to the optimization problem (16) is now expressed in the sought variables $\hat{\mathbf{A}}^{(k)}, \hat{\mathbf{B}}^{(k)}, \hat{\mathbf{C}}^{(k)}$ (through affine dependency in $\hat{\mathbf{A}}_i, \hat{\mathbf{B}}_i, \hat{\mathbf{C}}_i$), the given data $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i$ and in the different partitions of the Gramians for the error system, i.e., the solution to the equations in (14) that can easily be computed. Important to stress is that the Gramians are not decision variables, but auxiliary variables used to define and evaluate the cost function.

We next state the necessary conditions for optimality for (16).

Proposition 2.1 Necessary conditions for optimality Assume $\mathbf{A}_i$ and $\hat{\mathbf{A}}(\mathbf{p}_i)$ are Hurwitz for all i , for the $\mathcal{H}_2$ -norm to be defined. Then in order for the matrices $\hat{\mathbf{A}}^{(k)}, \hat{\mathbf{B}}^{(k)}$ and $\hat{\mathbf{C}}^{(k)}$ to be optimal for the problem (16), it is necessary that they satisfies the equations in (14) and

$$\sum_i w_k(\mathbf{p}_i) \left(\hat{\mathbf{Q}}_i \hat{\mathbf{P}}_i + \mathbf{Y}_i^T \mathbf{X}_i \right) = \mathbf{0}, \quad (18a)$$

$$\sum_i w_k(\mathbf{p}_i) \left(\hat{\mathbf{Q}}_i \hat{\mathbf{B}}_i + \mathbf{Y}_i^T \mathbf{B}_i \right) = \mathbf{0}, \quad (18b)$$

$$\sum_i w_k(\mathbf{p}_i) \left(\hat{\mathbf{C}}_i \hat{\mathbf{P}}_i - \mathbf{C}_i \mathbf{X}_i \right) = \mathbf{0}. \quad (18c)$$

Proof If $\mathbf{A}_i$ and $\hat{\mathbf{A}}(\mathbf{p}_i)$ is Hurwitz for all i , then we can solve the equations in (14) which are

needed to compute the cost function. Differentiating the cost function with respect to $\hat{\mathbf{A}}^{(k)}$, $\hat{\mathbf{B}}^{(k)}$ and $\hat{\mathbf{C}}^{(k)}$ entails the expressions (see Petersson (2010) for the calculations) in 18. $\square$

Remark 2: In the proof it is never assumed that all of the parameters in the matrices $\hat{\mathbf{A}}^{(k)}$, $\hat{\mathbf{B}}^{(k)}$ and $\hat{\mathbf{C}}^{(k)}$ should be free. One can choose any element to be either free or constant. With this fact it is possible to impose sparsity structure in the elements of the system matrices, as long as there exists an $\hat{\mathbf{A}}$ that is Hurwitz with that particular structure. Note also that in the proof we have expressions for the gradient of the cost functions which can be used in for example a quasi-Newton algorithm to find a local minimum.

2.2 Generation of Robust LPV State Space Models using Regularization

In the previous section we have tacitly assumed that the given data, (i.e., the state space matrices for different values of the scheduling parameters) are exact. In a more realistic setting we can assume the presence of errors (e.g., modeling, truncation or round-off) in these data. The question is how to cope with these errors and take them into account. This section will be an extension to the method presented in Section 2.1. We propose a problem-specific regularization, which we will show can be interpreted as a worst-case optimization approach (see Ben-Tal and Nemirovski (2002)).

Regularization can, for example, be used to make ill-posed problems well posed or to make a solution to an estimation problem less sensitive when having small amount of data. Commonly used regularization methods are for example l_1 - and l_2 -regularizations, for least squares problem referred to, in the l_1 -case as LASSO and in the l_2 -case Tikhonov regularization or ridge regression, see e.g., Hastie et al. (2001). In these regularizations, an extra term is added to the cost function to penalize the l_1 - or l_2 -norm of the sought variables,

$$V_{\text{regularized}}(x) = V_{\text{original}}(x) + \epsilon \|x\|_{l_1, l_2}. \quad (19)$$

The regularization parameter, here denoted ϵ , is seen as a design parameter.

In the problems discussed in this paper there is, intuitively, no reason for using either l_1 - or l_2 -regularization. We have no *a priori* knowledge about our systems, e.g., that the elements in the system matrices should be small (typically achieved using l_2 -regularization) or that the system matrices should be sparse (typically achieved using l_1 -regularization). Instead we will use the regularization as a proxy for a more difficult problem, namely robust optimization.

2.2.1 The Regularized Optimization Problem

To reduce the influence of errors in data, the unregularized cost function (15) is regularized by adding three new terms. These are the Frobenius norms of the derivatives of the cost function with respect to *the given data*, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, i.e., we want the solution we obtain to be less sensitive to uncertainties in *the data*. The optimization problem with these new terms becomes

$$\min_{\hat{\mathbf{A}}^{(k)}, \hat{\mathbf{B}}^{(k)}, \hat{\mathbf{C}}^{(k)}} \sum_i \|E_i\|_{\mathcal{H}_2}^2 + \epsilon_{\mathbf{A}} \left\| \frac{\partial \|E_i\|_{\mathcal{H}_2}^2}{\partial \mathbf{A}} \right\|_F + \epsilon_{\mathbf{B}} \left\| \frac{\partial \|E_i\|_{\mathcal{H}_2}^2}{\partial \mathbf{B}} \right\|_F + \epsilon_{\mathbf{C}} \left\| \frac{\partial \|E_i\|_{\mathcal{H}_2}^2}{\partial \mathbf{C}} \right\|_F. \quad (20)$$

Differentiating the cost function (15) with respect to $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ (calculations can be found

in Petersson (2010)) yields

$$\frac{\partial \|E_i\|_{\mathcal{H}_2}^2}{\partial \mathbf{A}} = 2 (\mathbf{Q}_i \mathbf{P}_i + \mathbf{Y}_i \mathbf{X}_i^T), \quad (21a)$$

$$\frac{\partial \|E_i\|_{\mathcal{H}_2}^2}{\partial \mathbf{B}} = 2 (\mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}_i), \quad (21b)$$

$$\frac{\partial \|E_i\|_{\mathcal{H}_2}^2}{\partial \mathbf{C}} = 2 (\mathbf{C}_i \mathbf{P}_i - \hat{\mathbf{C}}_i \mathbf{X}_i^T). \quad (21c)$$

Hence, the regularized optimization problem we want to solve is

$$\min_{\hat{\mathbf{A}}^{(k)}, \hat{\mathbf{B}}^{(k)}, \hat{\mathbf{C}}^{(k)}} V_{\text{reg}} = \min_{\hat{\mathbf{A}}^{(k)}, \hat{\mathbf{B}}^{(k)}, \hat{\mathbf{C}}^{(k)}} V + V_{\text{rob}}, \quad (22)$$

where

$$V_{\text{rob}} = 2 \sum_i \left(\epsilon_{\mathbf{A}} \|\mathbf{Q}_i \mathbf{P}_i + \mathbf{Y}_i \mathbf{X}_i^T\|_F + \epsilon_{\mathbf{B}} \|\mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}_i\|_F + \epsilon_{\mathbf{C}} \|\mathbf{C}_i \mathbf{P}_i - \hat{\mathbf{C}}_i \mathbf{X}_i^T\|_F \right), \quad (23)$$

and V is defined as in (17).

Proposition 2.2 Necessary conditions for optimality *In order for the matrices $\hat{\mathbf{A}}^{(k)}$, $\hat{\mathbf{B}}^{(k)}$ and $\hat{\mathbf{C}}^{(k)}$ to be optimal for (22), it is necessary that Proposition 2.1 is satisfied and additionally the following set of equations need to be satisfied for all i :*

$$\sum_i w_k(\mathbf{p}_i) \left(\epsilon_{\mathbf{A}} \frac{\mathbf{W}_{1,i} \mathbf{X}_i + \mathbf{Y}_i^T \mathbf{W}_{2,i}}{\|\mathbf{Q}_i \mathbf{P}_i + \mathbf{Y}_i \mathbf{X}_i^T\|_F} + \epsilon_{\mathbf{B}} \frac{\mathbf{Y}_i^T \mathbf{W}_{3,i}}{\|\mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}_i\|_F} + \epsilon_{\mathbf{C}} \frac{\mathbf{W}_{4,i} \mathbf{X}_i}{\|\mathbf{C}_i \mathbf{P}_i - \hat{\mathbf{C}}_i \mathbf{X}_i^T\|_F} \right) = \mathbf{0}, \quad (24a)$$

$$\sum_i w_k(\mathbf{p}_i) \left(\epsilon_{\mathbf{A}} \frac{\mathbf{W}_{1,i} \mathbf{B}}{\|\mathbf{Q}_i \mathbf{P}_i + \mathbf{Y}_i \mathbf{X}_i^T\|_F} + \epsilon_{\mathbf{B}} \frac{\mathbf{Y}_i^T (\mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}_i)}{\|\mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}_i\|_F} + \epsilon_{\mathbf{C}} \frac{\mathbf{W}_{4,i} \mathbf{B}_i}{\|\mathbf{C}_i \mathbf{P}_i - \hat{\mathbf{C}}_i \mathbf{X}_i^T\|_F} \right) = \mathbf{0}, \quad (24b)$$

$$\sum_i w_k(\mathbf{p}_i) \left(\epsilon_{\mathbf{A}} \frac{\mathbf{C}_i \mathbf{W}_{2,i}}{\|\mathbf{Q}_i \mathbf{P}_i + \mathbf{Y}_i \mathbf{X}_i^T\|_F} + \epsilon_{\mathbf{B}} \frac{\mathbf{C}_i \mathbf{W}_{3,i}}{\|\mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}_i\|_F} + \epsilon_{\mathbf{C}} \frac{(\mathbf{C}_i \mathbf{P}_i - \hat{\mathbf{C}}_i \mathbf{X}_i^T) \mathbf{X}_i}{\|\mathbf{C}_i \mathbf{P}_i - \hat{\mathbf{C}}_i \mathbf{X}_i^T\|_F} \right) = \mathbf{0}, \quad (24c)$$

with

$$\hat{\mathbf{A}}_i^T \mathbf{W}_{1,i} + \mathbf{W}_{1,i} \mathbf{A}_i + \mathbf{Y}_i^T (\mathbf{Q}_i \mathbf{P}_i + \mathbf{Y}_i \mathbf{X}_i^T) = \mathbf{0}, \quad (25a)$$

$$\mathbf{A}_i \mathbf{W}_{2,i} + \mathbf{W}_{2,i} \hat{\mathbf{A}}_i^T + (\mathbf{Q}_i \mathbf{P}_i + \mathbf{Y}_i \mathbf{X}_i^T) \mathbf{X}_i = \mathbf{0}, \quad (25b)$$

$$\mathbf{A}_i \mathbf{W}_{3,i} + \mathbf{W}_{3,i} \hat{\mathbf{A}}_i^T + (\mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}_i) \hat{\mathbf{B}}_i^T = \mathbf{0}, \quad (25c)$$

$$\hat{\mathbf{A}}_i^T \mathbf{W}_{4,i} + \mathbf{W}_{4,i} \mathbf{A}_i - \hat{\mathbf{C}}_i^T (\mathbf{C}_i \mathbf{P}_i - \hat{\mathbf{C}}_i \mathbf{X}_i^T) = \mathbf{0}. \quad (25d)$$

Proof The difference between problem (16) and problem (22) is that the term V_{rob} has been added to the cost function in the latter. This means that, for Proposition 2.2 to hold, Proposition 2.1

needs to hold and additionally, differentiate the cost function with respect to $\hat{\mathbf{A}}^{(k)}$, $\hat{\mathbf{B}}^{(k)}$ and $\hat{\mathbf{C}}^{(k)}$ (calculations can be found in Petersson (2010)) we arrive at the expressions in 24 and 25. $\square$

Remark 3: Just as in the proof for Proposition 2.1 it is never assumed that all of the parameters in the matrices $\hat{\mathbf{A}}^{(k)}$, $\hat{\mathbf{B}}^{(k)}$ and $\hat{\mathbf{C}}^{(k)}$ should be free. One can choose any element to be either free of constant. With this fact it is possible to impose arbitrary structure in the elements of the system matrices, as long as there exists an $\hat{\mathbf{A}}$ that is Hurwitz with that particular structure.

2.2.2 Interpretation of the Regularization Terms

In this section we will give an *additional* interpretation of the presented regularization that will explain how this problem specific regularization can be seen as a proxy to robust optimization. Let us start by studying the case when an unstructured error is added to the $\mathbf{B}$ -matrix, $\mathbf{B}_\Delta = \mathbf{B} + \Delta$,

$$\begin{aligned} V_\Delta = \text{tr} \left((\mathbf{B} + \Delta)^T \mathbf{Q} (\mathbf{B} + \Delta) + 2(\mathbf{B} + \Delta)^T \mathbf{Y} \hat{\mathbf{B}} + \hat{\mathbf{B}}^T \hat{\mathbf{Q}} \hat{\mathbf{B}} \right) \\ = \|E\|_{\mathcal{H}_2}^2 + \text{tr} \left(2\Delta^T (\mathbf{Q}\mathbf{B} + \mathbf{Y}\hat{\mathbf{B}}) + \Delta^T \mathbf{Q} \Delta \right) \end{aligned} \quad (26)$$

Now maximizing this expression with respect to Δ , under the assumption that the error is small in the Frobenius norm,

$$\begin{aligned} \max_{\|\Delta\|_F < \epsilon} V_\Delta = \|E\|_{\mathcal{H}_2}^2 + \max_{\|\Delta\|_F < \epsilon} \text{tr} \left(2\Delta^T (\mathbf{Q}\mathbf{B} + \mathbf{Y}\hat{\mathbf{B}}) + \Delta^T \mathbf{Q} \Delta \right) \\ = \|E\|_{\mathcal{H}_2}^2 + 2\epsilon \left\| \mathbf{Q}\mathbf{B} + \mathbf{Y}\hat{\mathbf{B}} \right\|_F + \mathcal{O}(\epsilon^2). \end{aligned} \quad (27)$$

Here, identify the second term (equation (21b)) as the Frobenius norm of the derivative of the cost function with respect to $\mathbf{B}$. Analogous calculation can be done when having a small unstructured error in $\mathbf{C}$.

3 Computational Aspects

In this section we will explain how the results derived in the previous sections can be used, and how the optimization can be performed efficiently. We will discuss one possible initialization for the LPV-generation method presented in Section 2.1 and give a more extensive explanation on one example of how to utilize the common structure in the Lyapunov equations that we need to solve in the methods to make them more efficient. Once we have efficient means to compute the cost function and its gradient, any quasi-Newton based optimization scheme can be used to actually perform the numerical search for a local optimum, see e.g., Nocedal and Wright (2006). Hence, to test the efficiency of the proposed algorithm, essentially any available commercial or open-source solver can be used. For the examples in this paper `fminunc` in MATLAB[®] has been used, but we have also tried SNOPT[®] (see Gill et al. (2005)) and `minFunc` (see Schmidt (2010)), which both performed similarly.

3.1 Initialization

The optimization problem studied in Section 2.1 is both nonlinear and non-convex, which means that an important part of the problem is to choose a good starting point. Looking at the cost function (17) again we see that the problem is a quadratic problem in $\hat{\mathbf{B}}$ (or $\hat{\mathbf{C}}$) if $\hat{\mathbf{A}}$ and $\hat{\mathbf{C}}$ (or $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$) is fixed. Realizing this, a strategy for finding a starting point is now proposed.

First, to have systems better suited for numeric computations, compute balanced realizations of the given models. Then choose a model from the set of models given, for example the model that is in the middle of the region of interest, let us call it $\tilde{G} : \left[\begin{array}{c|c} \tilde{\mathbf{A}} & \tilde{\mathbf{B}} \\ \hline \tilde{\mathbf{C}} & \mathbf{0} \end{array} \right]$. Now set $\hat{\mathbf{A}}(\mathbf{p}) = \tilde{\mathbf{A}}$, i.e., choose $\hat{\mathbf{A}}$ to be a constant matrix that does not depend on $\mathbf{p}$, and do the same thing for $\hat{\mathbf{C}}$. The problem of finding $\hat{\mathbf{B}}^{(k)}$ is now a quadratic problem which can be solved as explained below.

The cost function (17) was

$$V = \sum_i \text{tr} \left(\mathbf{B}_i^T \mathbf{Q}_i \mathbf{B}_i + 2\mathbf{B}_i^T \mathbf{Y}_i \hat{\mathbf{B}}_i + \hat{\mathbf{B}}_i^T \hat{\mathbf{Q}}_i \hat{\mathbf{B}}_i \right), \quad (28)$$

where we can write $\hat{\mathbf{B}}_i$ as

$$\begin{aligned} \hat{\mathbf{B}}_i &= (\mathbf{I}w_1(\mathbf{p}_i) \mathbf{I}w_2(\mathbf{p}_i) \mathbf{I}w_3(\mathbf{p}_i) \dots \mathbf{I}w_{N_w}(\mathbf{p}_i)) \\ &\quad \times ((\hat{\mathbf{B}}^{(1)})^T (\hat{\mathbf{B}}^{(2)})^T (\hat{\mathbf{B}}^{(3)})^T \dots (\hat{\mathbf{B}}^{(N_w)})^T)^T = \bar{\mathbf{p}}_i \bar{\mathbf{B}}. \end{aligned} \quad (29)$$

Now rewrite

$$\begin{aligned} V &= \sum_i \text{tr} \left(\mathbf{B}_i^T \mathbf{Q}_i \mathbf{B}_i + 2\mathbf{B}_i^T \mathbf{Y}_i \hat{\mathbf{B}}_i + \hat{\mathbf{B}}_i^T \hat{\mathbf{Q}}_i \hat{\mathbf{B}}_i \right) = \sum_i \text{tr} \left(\mathbf{B}_i^T \mathbf{Q}_i \mathbf{B}_i + 2\mathbf{B}_i^T \mathbf{Y}_i \bar{\mathbf{p}}_i \bar{\mathbf{B}} \right. \\ &\quad \left. + \bar{\mathbf{B}}^T \bar{\mathbf{p}}_i^T \hat{\mathbf{Q}}_i \bar{\mathbf{p}}_i \bar{\mathbf{B}} \right) = \text{tr} \left(\sum_i \mathbf{B}_i^T \mathbf{Q}_i \mathbf{B}_i + \left[2 \sum_i \mathbf{B}_i^T \mathbf{Y}_i \bar{\mathbf{p}}_i \right] \bar{\mathbf{B}} + \frac{1}{2} \bar{\mathbf{B}}^T \left[2 \sum_i \bar{\mathbf{p}}_i^T \hat{\mathbf{Q}}_i \bar{\mathbf{p}}_i \right] \bar{\mathbf{B}} \right) \\ &= \text{tr} \left(\mathbf{b}_1 + \mathbf{b}_2 \bar{\mathbf{B}} + \frac{1}{2} \bar{\mathbf{B}}^T \mathbf{b}_3 \bar{\mathbf{B}} \right). \end{aligned} \quad (30)$$

The solution to the problem $\min_{\bar{\mathbf{B}}} V$, which always exists since $\mathbf{b}_3$ is positive semidefinite, is the solution to the linear system of equations

$$\mathbf{b}_3 \bar{\mathbf{B}} = -\mathbf{b}_2^T. \quad (31)$$

Analogous calculations can be used to compute a starting point for $\hat{\mathbf{C}}^{(k)}$; define $\bar{\mathbf{C}} = (\hat{\mathbf{C}}^{(1)}, \hat{\mathbf{C}}^{(2)}, \dots, \hat{\mathbf{C}}^{(N_w)})$, use the same $\hat{\mathbf{A}}$ as for finding $\hat{\mathbf{B}}^{(k)}$ but use the $\hat{\mathbf{B}}$ that was found solving the quadratic problem described above. Now we obtain the equations

$$V = \text{tr} \left(\mathbf{c}_1 + \mathbf{c}_2 \bar{\mathbf{C}}^T + \frac{1}{2} \bar{\mathbf{C}} \mathbf{c}_3 \bar{\mathbf{C}}^T \right) \quad (32)$$

where $\mathbf{c}_1 = \sum_i \mathbf{C}_i \mathbf{P}_i \mathbf{C}_i^T$, $\mathbf{c}_2 = -2 \sum_i \mathbf{C}_i \mathbf{X}_i \bar{\mathbf{p}}_i$ and $\mathbf{c}_3 = 2 \sum_i \bar{\mathbf{p}}_i^T \hat{\mathbf{P}}_i \bar{\mathbf{p}}_i$. The solution to the quadratic problem in this case, which also always exists since $\mathbf{c}_3$ is positive semidefinite, is the solution to the system of linear equations

$$\bar{\mathbf{C}} \mathbf{c}_3 = -\mathbf{c}_2. \quad (33)$$

These are suggestions for finding initial values for $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$. For the moment the initialization that is used for $\hat{\mathbf{A}}$ is to choose a constant $\hat{\mathbf{A}}$ that represents a stable system.

3.2 Exploiting Structure in Gradient Computations

To compute the cost function and its gradient, both in problem (16) and problem (22), Lyapunov and Sylvester equations need to be solved. This is where most of the computational time is spent. These equations (equations (14) and (25)) have some common structure that we will use to speed up the computations. First the main steps (computationally) in solving a Sylvester equation are studied, and from there we show how to exploit the structure in the problem.

A general Sylvester equation can be written as

$$\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{C}, \quad \mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{B} \in \mathbb{R}^{n \times n}, \quad \mathbf{C} \in \mathbb{R}^{n \times n}. \quad (34)$$

The first main step when solving a Sylvester equation is to Schur factorize (see e.g., Golub and Van Loan (1996)) $\mathbf{A}$ and $\mathbf{B}$, which can be done in $\mathcal{O}(n^3)$ operations. Now the equation

$$\mathbf{A}_S \mathbf{X}_S + \mathbf{X}_S \mathbf{B}_S = \mathbf{C}_S \quad (35)$$

needs to be solved, where $\mathbf{A}_S = \mathbf{U}^T \mathbf{A} \mathbf{U}$ and $\mathbf{B}_S^T = \mathbf{V} \mathbf{B}^T \mathbf{V}^T$ are block upper triangular computed using the Schur factorization and $\mathbf{C}_S = \mathbf{U}^T \mathbf{C} \mathbf{V}$ and $\mathbf{X}_S = \mathbf{U}^T \mathbf{X} \mathbf{V}$. Note that a full matrix-matrix multiplication is needed to create $\mathbf{C}_S$, which is an $\mathcal{O}(n^3)$ operation. It is not hard to verify that the new system of linear equations, (35), can be solved in $\mathcal{O}(n^2)$ complexity, and the solution to (34) is computed by a full matrix-matrix multiplication, $\mathbf{X} = \mathbf{U} \mathbf{X}_S \mathbf{V}^T$. We can conclude that if we are going to solve several equations with the same factors $\mathbf{A}$ and $\mathbf{B}$ but different $\mathbf{C}$'s we can gain speed in the computations if we Schur factorize $\mathbf{A}$ and $\mathbf{B}$ before we start to solve the equations.

As an example, to compute the cost function (17), three Lyapunov/Sylvester equations (14d, 14e, 14f) need to be solved for every i and iteration in the optimization algorithm. In (14d), $\mathbf{Q}_i$, depend only on given data, and can thus be precomputed before the algorithm starts. Additionally looking at the equations in (14), all of the equations have $\mathbf{A}_i$ and/or $\hat{\mathbf{A}}_i$ as factors. This means that the solving of the Lyapunov/Sylvester equations can be sped up by Schur factorizing $\mathbf{A}_i$ and $\hat{\mathbf{A}}_i$ before starting to solve the Lyapunov/Sylvester equations. The Schur factorization of the matrices $\mathbf{A}_i$ (the given data) can be precomputed before starting the optimization algorithm. It is crucial to notice that the extra cost to compute the gradient is merely to solve two additional Lyapunov/Sylvester equations (14b, 14c), and that these Lyapunov/Sylvester equations involve the same two matrices $\mathbf{A}_i, \hat{\mathbf{A}}_i$ in all equations for a fixed i .

To compute the regularized cost function in (20), only $\mathbf{Q}_i, \mathbf{P}_i, \mathbf{X}_i, \mathbf{Y}_i$ are needed. However, $\mathbf{Q}_i$ and $\mathbf{Y}_i$ have already been calculated in the original cost function. The matrices $\mathbf{P}_i$, which only depend on given data and can be precomputed, and finally $\mathbf{X}_i$ has the previously Schur factorized $\mathbf{A}_i$ and $\hat{\mathbf{A}}_i$ as factors in the equation. When computing the gradient of the regularization terms, V_{rob} , in the regularized cost function, (24), we additionally need to solve four new Sylvester equations, (25). Once again these equations have the same structure, and only involves $\mathbf{A}_i$ and $\hat{\mathbf{A}}_i$ as factors which we already have Schur factorized, meaning that they can be solved efficiently.

Remark 4: As we mentioned in Section 2 it is possible to impose structure on the system matrices in the resulting LPV-model. If we choose the structure to be block triangular in $\hat{\mathbf{A}}$, i.e., already in a form such that we do not need to Schur factorize the matrix to solve the Lyapunov/Sylvester equation efficiently, then it is possible to speed up the computations even more.

4 Examples

In this section we start by presenting an illustrative example to shed light on some properties of the proposed method, and continue with a more practical example illustrating the applicability

Figure 1. The elements in the $\mathbf{A}$ -matrix as function of p . The system is given in a representation where the elements in the matrix depends non-linearly on p .

of the proposed method to realistically sized problems. Finally we conclude with an example that shows the potential of the regularized version of the problem.

When solving the examples, the function `fminunc` in MATLAB[®] was used as the quasi-Newton solver framework. To generate a starting point for the solver, which is an extremely important problem in need of significant amounts of research, the initialization procedure explained in Section 3.1 was used.

As a comparison in Example 4.1 we will use a method that will be called SMILE. The method is described in detail in De Caigny et al. (2012).

Example 4.1 Small LPV Approximation Example

To show the potential of the LPV approximation and illustrate the importance of addressing system properties, we study a small example.

The system in this example is defined by a connection of two second-order systems with parameter dependent damping,

$$G = G_1 G_2, \quad \text{where } G_1 = \frac{1}{s^2 + 2\zeta_1 s + 1}, \quad G_2 = \frac{9}{s^2 + 6\zeta_2 s + 9}, \quad (36a)$$

$$\zeta_1 = 0.1 + 0.9p, \quad \zeta_2 = 0.1 + 0.9(1 - p), \quad p \in [0, 1]. \quad (36b)$$

The system was sampled by selecting 30 equidistant points in $p \in [0, 1]$, *i.e.*, we are given 30 linear models with four states each.

The data is given in a state basis where all the elements in the system matrices happen to depend non-linearly on the parameter p , see Figure 1. The interesting and obvious property of this example is that there exists a state basis for which the model has affine dependence on p ; in fact only two elements of the system matrix $\mathbf{A}$ are affine in p while all other matrix elements in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are constants. From the results in Table 1 we see that when the proposed method (called NLP-method) is used, a high accuracy low order (indeed affine) LPV-model of the system can be found. If we try to obtain a model using the SMILE method with first order polynomials we obtain a much worse model. Achieving comparable results using the SMILE strategy requires polynomials of order two. To further illustrate the accuracy, 100 validation points were generated from (36) and the $\mathcal{H}_2$ -norm for the error model in these points is shown in Figure 2.

This example illustrates that even though the realization of the model given is non-polynomial

Figure 2. The figure illustrates the $\mathcal{H}_2$ -norm of the error system in 100 validation points for the different methods. Note the different scales and that it takes a polynomial of order two using the SMILE approach to obtain a satisfactory result, as with the proposed method using an affine function.

Table 1. Numerical results for Example 4.1.

Method	$\sum_i \ E_i\ _{\mathcal{H}_2}^2$	Degree	Time (s)
Proposed NLP-method	$1.34 \cdot 10^{-9}$	1	25
SMILE	$1.65 \cdot 10^{-27}$	2	0.72
SMILE	1.38	1	0.73

in the parameters, we are able to find an underlying model with only affine dependence. It thus illustrates the importance to look at the input-output relation.

To show that the method described in Section 2.1 is applicable to real-world examples we will now address larger examples.

Example 4.2 Nonlinear Aircraft Model

In this example we want to create an LPV-model of an aircraft. There are three models of different complexity; all models have 22 states but depend on one, two or three parameters. The models used in this example were developed and used in the COFCLUO project (Clearance Of Flight Control Laws Using Optimization, see <http://cofcluo.isy.liu.se/>). The three models describe, with different complexity, an airplane in closed-loop in the longitudinal direction. These models were used to evaluate different performance criteria for the airplane. The models are given as LFRs, which makes it possible to extract any number of linear models in the parameter region. This is how estimation and validation data is created. In this example we will try to create an affinely dependent LPV-model.

One problem with the models is that they are not strictly proper, i.e., $\mathbf{D} \neq \mathbf{0}$, which is necessary for the LPV-generating method to be applicable. This fact is circumvented by first ignoring the $\mathbf{D}$ -matrix and generate a model for the model without the $\mathbf{D}$ -matrix. After this we do a polynomial interpolation of the $\mathbf{D}$ -matrix, which is a scalar function because it is a single-input-single-output system, and afterwards incorporate this function for the $\mathbf{D}$ -matrix in the LPV-model.

All parameters are normalized and can vary between -1 and 1 . The resulting models are validated using the relative $\mathcal{H}_2$ -norm ($\frac{\sum_i \|G_i - \hat{G}(p_i)\|_{\mathcal{H}_2}}{\sum_i \|G_i\|_{\mathcal{H}_2}}$) for the system without the $\mathbf{D}$ -matrix.

The models are also validated using the relative $\mathcal{H}_\infty$ -norm ($\frac{\sum_i \|G_i - \hat{G}(p_i)\|_{\mathcal{H}_\infty}}{\sum_i \|G_i\|_{\mathcal{H}_\infty}}$) for the complete system, including the $\mathbf{D}$ -matrix.

The results from the LPV-generations of the different models can be seen in Table 2 and Figures 3, 4 and 5. In Figures 3, 4 and 5 we see that the reduced models represents the original models

Table 2. Numerical results for the airplane model in Example 4.2.

No. of parameters	Computational time	No. of models used for estimation
1	14m	10
2	1h 33m 12s	100
3	1h 10m 52s	125

Figure 3. Relative error in $\mathcal{H}_2$ - and $\mathcal{H}_\infty$ -norm at 100 validation points, in the one parameter case.

well.

As can be seen in the examples above the proposed method does a good job generating low order LPV-models. Note that in the last example, when minimizing the $\mathcal{H}_2$ -norm, also the $\mathcal{H}_\infty$ -norm remains small, see Figures 3, 4 and 5.

Example 4.3 Data with Noise

In this example we will show the potential of the regularized method described in Section 2.2. This example uses the model with one parameter in Example 4.2, and will illustrate how uncertainties can be addressed by the regularized version (20) of the optimization problem (8). In this example we are given 20 linear state space models but where the data is corrupted by noise. The poles of the given systems are perturbed such that if $s_{i,j}$ is the j :th pole of system i , then the perturbed system will have the new pole $\hat{s}_{i,j} = s_{i,j} + e$, where $e \sim \mathcal{N}(0, 1 \cdot 10^{-3})$.

To validate the models we calculate the relative error for the two models compared to the true model for 200 different values of $p \in [0, 1]$. In Figure 6 we plot the relative error for the model we obtain using the original cost function (black solid line) and models we obtain using different values on the regularization parameter using the regularized cost function. We see that by using the regularized cost function we take the uncertainty into account and find a better model.

5 Conclusions

We have proposed a new method for computing an LPV-model, given a number of LTI-models, using a nonlinear optimization approach. The proposed method tries to preserve the input-output behavior of the given systems by minimizing the $\mathcal{H}_2$ -norm of the error system. The cost

Figure 4. Relative error in $\mathcal{H}_2$ - and $\mathcal{H}_\infty$ -norm at 1225 validation points, in the two parameter case.

Figure 5. Relative error in $\mathcal{H}_2$ - and $\mathcal{H}_\infty$ -norm at 3375 validation points, in the three parameter case. Since an ordered plot in four dimensions is impossible the result is illustrated using a histogram.

function and its gradient are derived to be computationally efficient. This enables us to have a measure of first order optimality and to efficiently use standard quasi-Newton solvers to solve the problem. The method have shown to work both conceptually, on small examples, and on real-world examples.

We also have proposed a modification of the method, where we add a regularization term, to make the method more robust to errors in the given data. This modification is shown to have a nice interpretation and a clear connection to worst-case approaches.

The clear benefits of the new method

Figure 6. Error in the relative $\mathcal{H}_2$ -norm, compared to the true system, in 200 validation points for different values of ϵ for Example 4.2. The circles on the lines indicates a value for p that also were used for estimation. Note that $\epsilon = 0$ corresponds to the cost function without the regularization terms.

References

- Bamieh, B., and Giarre, L. (2002), “Identification of linear parameter varying models,” *International Journal of Robust and Nonlinear Control*, 12, 841–853.
- Ben-Tal, A., and Nemirovski, A. (2002), “Robust Optimization - Methodology and Applications,” *Mathematical Programming (Series B)*, 92, 453–480.
- De Caigny, J., Camino, J.F., and Swevers, J. (2011), “Interpolation-based modeling of MIMO LPV systems,” 19, 46–63.
- De Caigny, J., Pintelon, R., Camino, J.F., and Swevers, J. (2012), “Interpolated Modeling of LPV Systems Based on Observability and Controllability,” in *Proceedings of the 16th IFAC Symposium on System Identification*, Brussels, Belgium, pp. 1773–1778.
- Felici, F., Van Wingerden, J.W., and Verhaegen, M. (2007), “Subspace identification of MIMO LPV systems using a periodic scheduling sequence,” *Automatica*, 43, 1684–1697.
- Gill, P.E., Murray, W., and Saunders, M.A. (2005), “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Review*, 47, 99–131.
- Golub, G.H., and Van Loan, C.F., *Matrix Computations*, 3rd ed., Johns Hopkins University Press (1996).
- Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer (2001).
- Lee, L.H., and Poolla, K. (1999), “Identification of Linear Parameter-Varying Systems Using Nonlinear Programming,” *Journal of Dynamic Systems, Measurement and Control*, 121, 71–78.
- Leith, D.J., and Leithead, W.E. (2000), “Survey of gain-scheduling analysis and design,” *International Journal of Control*, 73, 1001–1025.
- Ljung, L., *System Identification: Theory for the User*, second ed., Prentice Hall (1999).
- Lovera, M., and Mercere, G. (2007), “Identification for gain-scheduling: a balanced subspace approach,” in *Proceedings of the American Control Conference*, New York, USA, pp. 858–863.
- Lovera, M., Novara, C., dos Santos, P.L., and Rivera, D. (2011), “Guest Editorial Special Issue on Applied LPV Modeling and Identification,” *IEEE Transactions on Control Systems Technology*, 19, 1–4.
- Marcos, A., and Balas, G.J. (2004), “Development of linear-parameter-varying models for aircraft,” *Journal of Guidance, Control and Dynamics*, 27, 218–228.

- Megretski, A., and Rantzer, A. (1997), “System analysis via integral quadratic constraints,” 42, 819–830.
- Mohammadpour, J., and Scherer, C.W. (eds.) *Control of Linear Parameter Varying Systems with Applications*, Springer US (2012).
- Nemani, M., Ravikanth, R., and Bamieh, B.A. (1995), “Identification of linear parametrically varying systems,” in *Proceedings of the 34th IEEE Conference on Decision and Control*, Vol. 3, Seville, Spain, pp. 2990–2995.
- Nocedal, J., and Wright, S.J., *Numerical Optimization*, Springer (2006).
- Petersson, D. (2010), “Nonlinear optimization approaches to $\mathcal{H}_2$ -norm based LPV modelling and control,” Licentiate Thesis no. 1453, Department of Electrical Engineering, Linköping University.
- Pfifer, H., and Hecker, S. (2008), “Generation of optimal linear parametric models for LFT-based robust stability analysis and control design,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, pp. 3866–3871.
- Rugh, W.J., and Shamma, J.S. (2000), “Research on gain scheduling,” *Automatica*, 36, 1401–1425.
- Schmidt, M., <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html> (2010).
- Shamma, J.S., and Athans, M. (1992), “Gain scheduling: potential hazards and possible remedies,” 12, 101–107.
- Steinbuch, M., van de Molengraft, R., and Van Der Voort, A.J. (2003), “Experimental modelling and LPV control of a motion system,” in *Proceedings of the American Control Conference*, Vol. 2, Denver, USA, pp. 1374–1379.
- Tóth, R. (2008), “Modeling and Identification of Linear Parameter-Varying Systems, an Orthonormal Basis Function Approach,” Delft University of Technology.
- Tóth, R., Felici, F., Heuberger, P.S.C., and Van Den Hof, P. (2007), “Discrete time LPV I/O and state-space representations, differences of behavior and pitfalls of interpolation,” in *Proceedings of the European Control Conference*, Kos, Greece, pp. 5418–5425.
- Varga, A., Looye, G., Moormann, D., and Grübel, G. (1998), “Automated generation of LFT-based parametric uncertainty descriptions from generic aircraft models,” *Mathematical and Computer Modelling of Dynamical Systems*, 4, 249–274.
- Wassink, M.G., Van De Wal, M., Scherer, C., and Bosgra, O. (2005), “LPV control for a wafer stage: beyond the theoretical solution,” *Control Engineering Practice*, 13, 231–245.
- Zhou, K., Doyle, J.C., and Glover, K., *Robust and optimal control*, Prentice-Hall, Inc. (1996).