

Robust Generation of LPV State-Space Models Using a Regularized \mathcal{H}_2 -Cost

Daniel Petersson and Johan Löfberg

Abstract—In this paper we present a regularization of an \mathcal{H}_2 -minimization based LPV-model generation algorithm. Our goal is to take care of uncertainties in the data, and obtain more robust models when we have few data. We give an interpretation of the regularization, which shows that the regularization has connections to robust optimization and worst-case approaches. We present how to effectively calculate the original cost function and its gradient, and extend these ideas to the regularized cost function and its gradient. A few examples, illustrating effects of both uncertain and few data, are finally presented to show the validity of the regularization.

I. INTRODUCTION

The behavior of a linear parameter varying (LPV)-model 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 $x(t)$ are the states, $u(t)$ and $y(t)$ 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 typically mass, position of center of gravity and various aerodynamic coefficients, but can also include state dependent parameters such as altitude and velocity, specifying current flight conditions. In this paper we will study the case when the parameters vary slowly and we will thus not take time dependence of the parameters into account.

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

$$G_i = \left[\begin{array}{c|c} \mathbf{A}_i & \mathbf{B}_i \\ \hline \mathbf{C}_i & \mathbf{D}_i \end{array} \right],$$

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

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

whose state-space realization depends polynomially on p . A frequently used method today is element-wise approximation see *e.g.* [1]. This method interpolates the elements in the system matrices individually with rational or polynomial functions. A possible drawback with this approach is that it fails to take system properties into account. Additionally, a prerequisite for the application of this method is that the number of states is the same in all models and the matrices correspond to the same ordering of states, which

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}

the proposed method in this paper do not need. Other methods that also use interpolation are *e.g.* [2], [3], but they transform the models into canonical state-space forms before performing the interpolation. There are also methods that address the problem of preserving the input-output relation and try to generate an LPV-model using, *e.g.* linear regression [4] or nonlinear optimization [5]. An excellent survey of existing methods can be found in [6].

One important thing to consider is that the given data, $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i$, might be corrupted by noise or there could be only a small amount of data available. This is often addressed by using regularization and robust optimization, but a problem then is that the optimization problem easily can become much more complicated or even intractable [7].

In this paper we extend the idea presented in [8], by adding a problem-specific regularization to the problem. We give a robust optimization interpretation of the regularization and motivate it with some examples. It is shown how to, in an efficient way, calculate the original cost function and its gradient, and similar ideas are applied to the regularized cost function and its gradient.

II. GENERATION OF LPV STATE-SPACE MODELS USING \mathcal{H}_2 -MINIMIZATION

In this section we present the optimization problem introduced in [8]. This problem arises when trying to approximate a multi-model system with an LPV-model. The optimization problem is formulated such that the sought model should capture input-output behavior of the sampled models. The objective is to minimize the error between the true models and the sought LPV-model in the sampled points in the \mathcal{H}_2 -norm, *i.e.*

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

where

$$G_i = \left[\begin{array}{c|c} \mathbf{A}_i & \mathbf{B}_i \\ \hline \mathbf{C}_i & \mathbf{D}_i \end{array} \right]$$

are the sampled (given) 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 LPV-model depending on the parameter p . We will in this paper assume that the system matrices in the LPV-model depend polynomially on the parameters, *i.e.*

$$\hat{\mathbf{A}}(p) = \hat{\mathbf{A}}_{p^0} + \hat{\mathbf{A}}_{p^1}p + \dots + \hat{\mathbf{A}}_{p^{k_A}}p^{k_A}, \quad (2a)$$

$$\hat{\mathbf{B}}(p) = \hat{\mathbf{B}}_{p^0} + \hat{\mathbf{B}}_{p^1}p + \dots + \hat{\mathbf{B}}_{p^{k_B}}p^{k_B}, \quad (2b)$$

$$\hat{\mathbf{C}}(p) = \hat{\mathbf{C}}_{p^0} + \hat{\mathbf{C}}_{p^1}p + \dots + \hat{\mathbf{C}}_{p^{k_C}}p^{k_C}. \quad (2c)$$

Our goal is to find the coefficient matrices $\hat{\mathbf{A}}_{p^i}, \hat{\mathbf{B}}_{p^i}, \hat{\mathbf{C}}_{p^i}$, that $\hat{\mathbf{A}}(p), \hat{\mathbf{B}}(p), \hat{\mathbf{C}}(p)$ are composed of. We start by looking at the models in one sample point and omit the index i . This will later be generalized to the case where we have multiple models. The error systems is defined as

$$E = G - \hat{G}(p).$$

The error system can be realized in state-space form as

$$E = \left[\begin{array}{c|c} \mathbf{A}_e & \mathbf{B}_e \\ \mathbf{C}_e & \mathbf{D}_e \end{array} \right] = \left[\begin{array}{c|c} \left(\begin{array}{c} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{A}} \end{array} \right) & \left(\begin{array}{c} \mathbf{B} \\ \hat{\mathbf{B}} \end{array} \right) \\ \hline (\mathbf{C} & -\hat{\mathbf{C}}) & \mathbf{D} - \hat{\mathbf{D}} \end{array} \right]. \quad (3)$$

This realization of the error system will later prove beneficial in 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.* we need $\mathbf{D} = \hat{\mathbf{D}}$ for all models or that both $\mathbf{D} = \mathbf{0}$ and $\hat{\mathbf{D}} = \mathbf{0}$. We can thus see the problem of finding $\hat{\mathbf{D}}$ as a separate problem which we do not address in this paper.

A. Rewriting the \mathcal{H}_2 -norm

To calculate the cost function we rewrite it to a numerically more suitable form (see [8], [9]).

$$\|E\|_{\mathcal{H}_2}^2 = \text{tr} \mathbf{B}_e^T \mathbf{Q}_e \mathbf{B}_e = \text{tr} \mathbf{C}_e \mathbf{P}_e \mathbf{C}_e^T. \quad (4)$$

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

$$\begin{aligned} \mathbf{A}_e \mathbf{P}_e + \mathbf{P}_e \mathbf{A}_e^T + \mathbf{B}_e \mathbf{B}_e^T &= \mathbf{0}, \\ \mathbf{A}_e^T \mathbf{Q}_e + \mathbf{Q}_e \mathbf{A}_e + \mathbf{C}_e^T \mathbf{C}_e &= \mathbf{0}. \end{aligned}$$

With the realization (3) of E and the following partitioning of the Gramians \mathbf{P}_e and \mathbf{Q}_e

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

we obtain six Sylvester and Lyapunov equations from the equations for the Gramians above

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

$$\mathbf{A}\mathbf{X} + \mathbf{X}\hat{\mathbf{A}}^T + \mathbf{B}\hat{\mathbf{B}}^T = \mathbf{0}, \quad (5b)$$

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

$$\mathbf{A}^T \mathbf{Q} + \mathbf{Q}\mathbf{A} + \mathbf{C}^T \mathbf{C} = \mathbf{0}, \quad (5d)$$

$$\mathbf{A}^T \mathbf{Y} + \mathbf{Y}\hat{\mathbf{A}} - \mathbf{C}^T \hat{\mathbf{C}} = \mathbf{0}, \quad (5e)$$

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

We note that \mathbf{P} and \mathbf{Q} satisfy the Lyapunov equations for the controllability and the observability Gramians for the given system, and $\hat{\mathbf{P}}$ and $\hat{\mathbf{Q}}$ satisfy the Lyapunov equations for the controllability and the observability Gramians for the sought system. With the partitioning of \mathbf{P}_e and \mathbf{Q}_e it is possible to rewrite (4) as

$$\|E\|_{\mathcal{H}_2}^2 = \text{tr} \left(\mathbf{B}^T \mathbf{Q} \mathbf{B} + 2\mathbf{B}^T \mathbf{Y} \hat{\mathbf{B}} + \hat{\mathbf{B}}^T \hat{\mathbf{Q}} \hat{\mathbf{B}} \right) \quad (6a)$$

$$\|E\|_{\mathcal{H}_2}^2 = \text{tr} \left(\mathbf{C} \mathbf{P} \mathbf{C}^T - 2\mathbf{C} \mathbf{X} \hat{\mathbf{C}}^T + \hat{\mathbf{C}} \hat{\mathbf{P}} \hat{\mathbf{C}}^T \right). \quad (6b)$$

Both of these equations can be used to calculate the \mathcal{H}_2 -norm and will serve different purposes in the next section.

III. THE OPTIMIZATION PROBLEM

In this section, we try to solve the optimization problem by addressing it as a general nonlinear optimization problem, taking great care to develop expressions that can be evaluated and differentiated efficiently.

A. Cost Function

The two equations in (6) can both be used to calculate the \mathcal{H}_2 -norm and are both useful to simplify the derivations for the gradients later. But for now we will only use (6a) to calculate the cost function. It is now straightforward to express the cost function for the more general case when we have multiple models, *i.e.* rewrite the cost function V in (1) with the new partitioning, as

$$V = \sum_i \|E_i\|_{\mathcal{H}_2}^2 = \sum_i \text{tr} \left(\mathbf{B}_i^T \mathbf{Q}_i \mathbf{B}_i + 2\mathbf{B}_i^T \mathbf{Y}_i \hat{\mathbf{B}}(p^{(i)}) + \hat{\mathbf{B}}(p^{(i)})^T \hat{\mathbf{Q}}_i \hat{\mathbf{B}}(p^{(i)}) \right) \quad (7)$$

The optimization problem (1) can now be written as

$$\min_{\hat{\mathbf{A}}_{p^k}, \hat{\mathbf{B}}_{p^k}, \hat{\mathbf{C}}_{p^k}} V. \quad (8)$$

Keep in mind the parametrization of the system matrices introduced in (2). Additionally, $\mathbf{P}_i, \mathbf{Q}_i, \hat{\mathbf{P}}_i, \hat{\mathbf{Q}}_i, \mathbf{X}_i$ and \mathbf{Y}_i satisfy the equations

$$\mathbf{A}_i \mathbf{P}_i + \mathbf{P}_i \mathbf{A}_i^T + \mathbf{B}_i \mathbf{B}_i^T = \mathbf{0}, \quad (9a)$$

$$\mathbf{A}_i \mathbf{X}_i + \mathbf{X}_i \hat{\mathbf{A}}(p^{(i)})^T + \mathbf{B}_i \hat{\mathbf{B}}(p^{(i)})^T = \mathbf{0}, \quad (9b)$$

$$\hat{\mathbf{A}}(p^{(i)}) \hat{\mathbf{P}}_i + \hat{\mathbf{P}}_i \hat{\mathbf{A}}(p^{(i)})^T + \hat{\mathbf{B}}(p^{(i)}) \hat{\mathbf{B}}(p^{(i)})^T = \mathbf{0}, \quad (9c)$$

$$\mathbf{A}_i^T \mathbf{Q}_i + \mathbf{Q}_i \mathbf{A}_i + \mathbf{C}_i^T \mathbf{C}_i = \mathbf{0}, \quad (9d)$$

$$\mathbf{A}_i^T \mathbf{Y}_i + \mathbf{Y}_i \hat{\mathbf{A}}(p^{(i)}) - \mathbf{C}_i^T \hat{\mathbf{C}}(p^{(i)}) = \mathbf{0}, \quad (9e)$$

$$\hat{\mathbf{A}}(p^{(i)})^T \hat{\mathbf{Q}}_i + \hat{\mathbf{Q}}_i \hat{\mathbf{A}}(p^{(i)}) + \hat{\mathbf{C}}(p^{(i)})^T \hat{\mathbf{C}}(p^{(i)}) = \mathbf{0}. \quad (9f)$$

The cost function to the optimization problem (8) is now expressed in the sought variables $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}$, 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 solutions to the equations in (9) that can easily be calculated. The Gramians are thus not decision variables, but auxiliary variables used to evaluate the cost function.

B. Gradient

An appealing feature of the proposed nonlinear optimization approach to solve the problem is that the equations in (6) are differentiable in the system matrices, $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ (see [10], [11]).

$$\frac{\partial \|E\|_{\mathcal{H}_2}^2}{\partial \hat{\mathbf{A}}} = 2 \left(\hat{\mathbf{Q}} \hat{\mathbf{P}} + \mathbf{Y}^T \mathbf{X} \right) \quad (10a)$$

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

$$\frac{\partial \|E\|_{\mathcal{H}_2}^2}{\partial \hat{\mathbf{C}}} = 2 \left(\hat{\mathbf{C}} \hat{\mathbf{P}} - \mathbf{C} \mathbf{X} \right) \quad (10c)$$

The closed form expressions obtained are expressed in the given data (\mathbf{A} , \mathbf{B} and \mathbf{C}), the optimization variables ($\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$) and solutions to equations (9), some of them already computed when calculating the cost function. To be more precise, the computational effort of computing the derivative is within a small constant factor from the computational effort required to compute the cost function. See Section IV-C for a more detailed discussion on computational aspects.

The results can easily be extended to the general form where we are given multiple models and the LPV-model has polynomial dependence in the parameters *i.e.* $\hat{\mathbf{A}}(p) = \hat{\mathbf{A}}_{p^0} + \hat{\mathbf{A}}_{p^1}p + \hat{\mathbf{A}}_{p^2}p^2 + \dots + \hat{\mathbf{A}}_{p^k}p^k$. The gradient of (7) with respect to the coefficient matrices $\hat{\mathbf{A}}_{p^j}$, $\hat{\mathbf{B}}_{p^j}$, $\hat{\mathbf{C}}_{p^j}$ becomes

$$\begin{aligned}\frac{\partial V}{\partial \hat{\mathbf{A}}_{p^j}} &= 2 \sum_i \left(p^{(i)} \right)^j \left(\hat{\mathbf{Q}}_i \hat{\mathbf{P}}_i + \mathbf{Y}_i^T \mathbf{X}_i \right) \\ \frac{\partial V}{\partial \hat{\mathbf{B}}_{p^j}} &= 2 \sum_i \left(p^{(i)} \right)^j \left(\hat{\mathbf{Q}}_i \hat{\mathbf{B}}_i + \mathbf{Y}_i^T \mathbf{B}_i \right) \\ \frac{\partial V}{\partial \hat{\mathbf{C}}_{p^j}} &= 2 \sum_i \left(p^{(i)} \right)^j \left(\hat{\mathbf{C}}_i \hat{\mathbf{P}}_i - \mathbf{C}_i \mathbf{X}_i \right).\end{aligned}$$

IV. DEALING WITH UNCERTAINTIES IN DATA

In the previous sections we have assumed that the given data, *i.e.*, the state-space matrices in the different operating points, are exact. In reality, there can be some errors in the data. The question is how to cope with these errors and take them into account. The method we propose is to use a problem-specific regularization, which we will show can be interpreted as a worst-case optimization approach.

A. Regularized Cost Function

To reduce the influence of errors in data, we regularize the cost function by adding three new terms to it. These are the Frobenius norm of the derivative of the cost function with respect to the given data, \mathbf{A} , \mathbf{B} and \mathbf{C} , *i.e.*

$$\begin{aligned}\min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}} & \|E\|_{\mathcal{H}_2}^2 + \epsilon_{\mathbf{A}} \left\| \frac{\partial \|E\|_{\mathcal{H}_2}^2}{\partial \mathbf{A}} \right\|_F + \\ & + \epsilon_{\mathbf{B}} \left\| \frac{\partial \|E\|_{\mathcal{H}_2}^2}{\partial \mathbf{B}} \right\|_F + \epsilon_{\mathbf{C}} \left\| \frac{\partial \|E\|_{\mathcal{H}_2}^2}{\partial \mathbf{C}} \right\|_F.\end{aligned}\quad (11)$$

As was the case for the original cost function, this regularized cost function is also differentiable in the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} . Using the same strategy as the one used to derive (10) we obtain

$$\begin{aligned}\frac{\partial \|E\|_{\mathcal{H}_2}^2}{\partial \mathbf{A}} &= 2 (\mathbf{Q}\mathbf{P} + \mathbf{Y}\mathbf{X}^T) \\ \frac{\partial \|E\|_{\mathcal{H}_2}^2}{\partial \mathbf{B}} &= 2 (\mathbf{Q}\mathbf{B} + \mathbf{Y}\hat{\mathbf{B}}) \\ \frac{\partial \|E\|_{\mathcal{H}_2}^2}{\partial \mathbf{C}} &= 2 (\mathbf{C}\mathbf{P} - \hat{\mathbf{C}}\mathbf{X}^T)\end{aligned}$$

The regularized cost function for the general case is defined analogously.

$$\begin{aligned}V_{reg} &= V + 2 \sum_i \left(\left\| \mathbf{Q}_i \mathbf{P}_i + \mathbf{Y}_i \mathbf{X}_i^T \right\|_F + \right. \\ & \quad \left. + \left\| \mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}(p^{(i)}) \right\|_F + \right. \\ & \quad \left. + \left\| \mathbf{C}_i \mathbf{P}_i - \hat{\mathbf{C}}(p^{(i)}) \mathbf{X}_i^T \right\|_F \right) = V + V_{rob}\end{aligned}\quad (12)$$

1) *Interpretation:* To give an interpretation of the choice of the Frobenius norm in (11), we look at the case where we have an unstructured error in the \mathbf{B} matrix.

$$\begin{aligned}V_{\Delta} &= \text{tr} \left((\mathbf{B} + \Delta)^T \mathbf{Q} (\mathbf{B} + \Delta) + 2(\mathbf{B} + \Delta)^T \mathbf{Y} \hat{\mathbf{B}} + \right. \\ & \quad \left. + \hat{\mathbf{B}}^T \hat{\mathbf{Q}} \hat{\mathbf{B}} \right) = V + \text{tr} \left(2\Delta^T (\mathbf{Q}\mathbf{B} + \mathbf{Y}\hat{\mathbf{B}}) + \Delta^T \mathbf{Q} \Delta \right)\end{aligned}$$

Now maximize this expression with respect to Δ , under the assumption that the error is small in some norm, *e.g.* $\|\Delta\|_F < \epsilon$.

$$\begin{aligned}\max_{\Delta} V_{\Delta} &= V + \max_{\Delta} \text{tr} \left(2\Delta^T (\mathbf{Q}\mathbf{B} + \mathbf{Y}\hat{\mathbf{B}}) + \right. \\ & \quad \left. + \Delta^T \mathbf{Q} \Delta \right) = V + 2\epsilon \left\| \mathbf{Q}\mathbf{B} + \mathbf{Y}\hat{\mathbf{B}} \right\|_F + O(\epsilon^2)\end{aligned}$$

Here we identify the second term as the Frobenius norm of the derivate of the cost function with respect to \mathbf{B} . Analogous calculation can be done when we have a small unstructured error in \mathbf{C} . This shows that the approach has clear connections to recently popularized worst-case approaches, [7]. Regarding \mathbf{A} , the interpretation is not as clear and is currently an open question.

B. Gradient for the Regularized Cost Function

Finding an explicit expression of the derivative of the regularized cost function with the extra regularizing terms inserted can be done using the same methodology as in [8], but the details are omitted for brevity. The gradient of the new part, V_{rob} , is

$$\begin{aligned}\frac{\partial V_{rob}}{\partial \mathbf{A}} &= 8\epsilon_{\mathbf{A}} [(\mathbf{F}_i + \mathbf{G}_i) \mathbf{X}_i + \mathbf{Y}_i^T (\mathbf{M}_i + \mathbf{N}_i)] + \\ & \quad + 8\epsilon_{\mathbf{B}} \mathbf{Y}_i^T [\mathbf{Z}_i + \mathbf{V}_i] + 8\epsilon_{\mathbf{C}} [\mathbf{W}_i + \mathbf{U}_i] \mathbf{X}_i\end{aligned}\quad (13a)$$

$$\begin{aligned}\frac{\partial V_{rob}}{\partial \mathbf{B}} &= 8\epsilon_{\mathbf{A}} [\mathbf{F}_i + \mathbf{G}_i] \mathbf{B}_i + 8\epsilon_{\mathbf{C}} [\mathbf{W}_i + \mathbf{U}_i] \mathbf{B}_i \\ & \quad + 8\epsilon_{\mathbf{B}} \left[\mathbf{Y}_i^T (\mathbf{Q}_i \mathbf{B}_i + \mathbf{Y}_i \hat{\mathbf{B}}(p^{(i)})) \right]\end{aligned}\quad (13b)$$

$$\begin{aligned}\frac{\partial V_{rob}}{\partial \mathbf{C}} &= 8\epsilon_{\mathbf{C}} [\hat{\mathbf{C}}(p^{(i)}) \mathbf{X}_i^T - \mathbf{C}_i \mathbf{P}_i] \mathbf{X}_i - \\ & \quad - 8\epsilon_{\mathbf{A}} \mathbf{C}_i [\mathbf{M}_i + \mathbf{N}_i] - 8\epsilon_{\mathbf{B}} \mathbf{C}_i [\mathbf{Z}_i + \mathbf{V}_i]\end{aligned}\quad (13c)$$

where $\mathbf{M}, \mathbf{N}, \mathbf{F}, \mathbf{G}, \mathbf{W}, \mathbf{U}, \mathbf{Z}, \mathbf{V}$ satisfy eight new Sylvester equations that we need to solve

$$\mathbf{A}_i \mathbf{M}_i + \mathbf{M}_i \hat{\mathbf{A}}(p^{(i)})^T + \mathbf{Q}_i \mathbf{P}_i \mathbf{X}_i = \mathbf{0}, \quad (14a)$$

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

$$\hat{\mathbf{A}}(p^{(i)})^T \mathbf{F}_i + \mathbf{F}_i \mathbf{A}_i + \mathbf{Y}_i^T \mathbf{Q}_i \mathbf{P}_i = \mathbf{0}, \quad (14c)$$

$$\hat{\mathbf{A}}(p^{(i)})^T \mathbf{G}_i + \mathbf{G}_i \mathbf{A}_i + \mathbf{Y}_i^T \mathbf{Y}_i \mathbf{X}_i^T = \mathbf{0}, \quad (14d)$$

$$\hat{\mathbf{A}}(p^{(i)})^T \mathbf{W}_i + \mathbf{W}_i \mathbf{A}_i + \hat{\mathbf{C}}(p^{(i)})^T \hat{\mathbf{C}}(p^{(i)}) \mathbf{X}_i^T = \mathbf{0}, \quad (14e)$$

$$\hat{\mathbf{A}}(p^{(i)})^T \mathbf{U}_i + \mathbf{U}_i \mathbf{A}_i - \hat{\mathbf{C}}(p^{(i)})^T \mathbf{C}_i \mathbf{P}_i = \mathbf{0}, \quad (14f)$$

$$\mathbf{A}_i \mathbf{Z}_i + \mathbf{Z}_i \hat{\mathbf{A}}(p^{(i)})^T + \mathbf{Q}_i \mathbf{B}_i \hat{\mathbf{B}}(p^{(i)})^T = \mathbf{0}, \quad (14g)$$

$$\mathbf{A}_i \mathbf{V}_i + \mathbf{V}_i \hat{\mathbf{A}}(p^{(i)})^T + \mathbf{Y}_i \hat{\mathbf{B}}(p^{(i)}) \hat{\mathbf{B}}(p^{(i)})^T = \mathbf{0}. \quad (14h)$$

C. Computations

We first describe the steps required to calculate the original cost function and its gradient in an efficient way, and then extend this to the regularized cost function.

To calculate the cost function (7), three Lyapunov/Sylvester equations (9d,9e,9f) need to be solved¹ for every i and iteration in an optimization algorithm. First notice that \mathbf{Q}_i in (9d) only depend on given data, and can thus be precomputed before the algorithm starts. Additionally if we look at equations (9), we see that all of the equations have \mathbf{A}_i and/or $\hat{\mathbf{A}}_i$ as factors. This means that we can speed up the computation of the Lyapunov/Sylvester equations by Schur factoring \mathbf{A}_i and $\hat{\mathbf{A}}_i$ before we start solving the Lyapunov/Sylvester equations. The Schur factorization of the matrices \mathbf{A}_i can be precomputed before we start the optimization algorithm. Crucial to notice is that the extra cost to compute the gradient is merely to solve two additional Lyapunov/Sylvester equations (9b,9c), 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 calculate the regularized cost function (11), we see that we need $\mathbf{Q}_i, \mathbf{P}_i, \mathbf{X}_i, \mathbf{Y}_i$. But \mathbf{Q}_i and \mathbf{Y}_i have already been calculated in the original cost function. \mathbf{P}_i , which only depend on given data, can be precomputed and finally \mathbf{Y}_i has the previously Schur factorized \mathbf{A}_i and $\hat{\mathbf{A}}_i$ as factors in the equation.

To calculate the gradient of the regularization terms, V_{rob} , in the extended cost function, (13), we need to solve eight new Sylvester equations, (14). But 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, which means that they can be solved efficiently.

D. A Complete Solver

Once we have efficiently computable expressions for 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 [12]. Hence, to test the efficiency of the proposed algorithm, essentially any

¹A Lyapunov or Sylvester solver is simply speaking based on three major steps of cubic complexity, Schur factorization of the multiplying factors, solution of triangular matrix equation, and some dense matrix multiplications

available commercial or open-source solver can be used. It is thus beyond the scope of this paper to give any details on how a complete solver is implemented.

V. EXAMPLES

In this section we will present three examples, based on the same LPV-model, to see how a low amount of data and uncertainties can be addressed by the new regularized version of the optimization problem.

When solving the problems, 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 with much research left to do, the linear model given in the mid-point of the parameter-space was used, after a balanced realization, to initialize $(\hat{\mathbf{A}}_{p^0}, \hat{\mathbf{B}}_{p^0}, \hat{\mathbf{C}}_{p^0})$. All other parameters were initialized to zero. The initialization part of the problem is of course very important due to the non-convexity in the problem, but this initialization has shown to be sufficient for the models that have been tested by the authors. 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 7.9 (R2009b)

The underlying LPV-model in the examples is $G(p) = G_1(p)G_2(p)$ where $G_1(p) = \frac{1}{s^2 + 2\zeta_1(p)s + 1}$ and $G_2(p) = \frac{9}{s^2 + 6\zeta_2(p)s + 9}$ with $\zeta_1(p) = 0.1 + 0.9p$ and $\zeta_2(p) = 0.1 + 0.9(1 - p)$ and $p \in [0, 1]$.

A. Reference Example

In this example the model was sampled in 30 points equidistantly in $[0, 1]$ *i.e.* we are given 30 linear models with four states. The data is given in a state basis for which all the elements in the system matrices happen to depend nonlinearly on the parameter p , see Fig. 1. In this basis it will undoubtedly be hard to find a good low order approximation with an element-wise approach with polynomial dependence of p . The interesting and obvious property of this example is that there exists a state basis for which the model has linear dependence on p , in fact only two elements of the system matrix \mathbf{A} are linear in p and all other matrix elements in $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are constants.

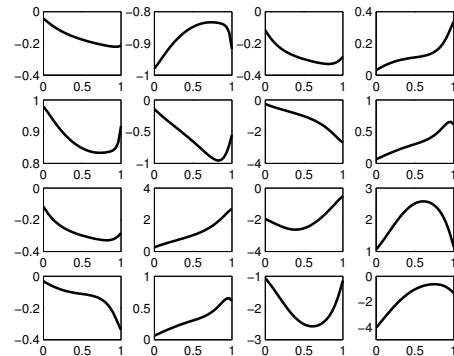


Fig. 1. The elements in the \mathbf{A} -matrix depending on p .

TABLE I
RESULTS FOR THE EXAMPLE V-A

	$\sum_i \ E_i\ _{\mathcal{H}_2}^2$	Degree	Time(s)	Iter
H2-NLP	$5.37 \cdot 10^{-7}$	1	393	251
EW-9	$3.79 \cdot 10^{-5}$	9	0.036	–
EW-1	0.514	1	0.034	–

To validate the result, 15 validation points were generated. From the result in Table I we see that a high accuracy low order (indeed linear) LPV-model of the system can be found. If we try to obtain a model using an element-wise method, interpolating the elements in the matrices independently, with first order polynomials we, of course, obtain a much worse model. Achieving comparable results using an element-wise strategy requires polynomials of order 9. To further illustrate the accuracy in the validation points, the \mathcal{H}_2 -norm for the error model in the 15 validation points is shown in Fig. 2.

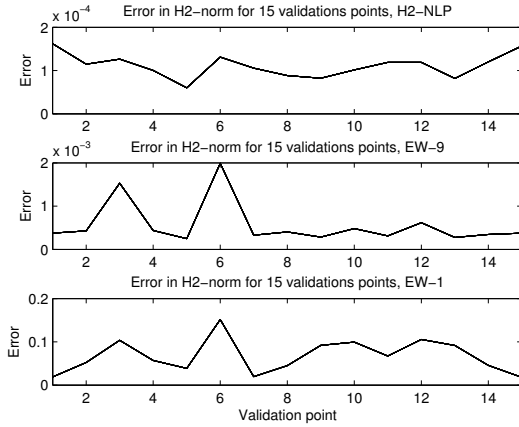


Fig. 2. \mathcal{H}_2 -norm in 15 validation points for the different methods

B. Example with Uncertain Data

Again we use the same model and we are given 15 models, but the data is corrupted by noise.

$$\mathbf{A}_u = \mathbf{A} + e_A, e_A \sim \mathcal{N}(0, 10^{-4}) \quad (15a)$$

$$\mathbf{B}_u = \mathbf{B} + e_B, e_B \sim \mathcal{N}(0, 10^{-4}) \quad (15b)$$

$$\mathbf{C}_u = \mathbf{C} + e_C, e_C \sim \mathcal{N}(0, 10^{-4}) \quad (15c)$$

If we now use the regularized cost function (11) with $\epsilon_A = \epsilon_B = \epsilon_C = 0.01$ and compare with using the original cost function we see in Figure 3 that the regularized cost function finds a better model, *i.e.* takes the noise into account.

C. Example with Uncertain Data and Few Data

Once again we use the same model, but in this case we are only given three models instead of 30, in the points $p = \{-1, 0, 1\}$. The data is corrupted by noise in the same way as in the previous example, (15). If we now use the regularized cost function (11) with $\epsilon_A = \epsilon_B = \epsilon_C = 0.01$ and compare with using the original cost function we see in Figure 4 that we get a better result using the regularized one.

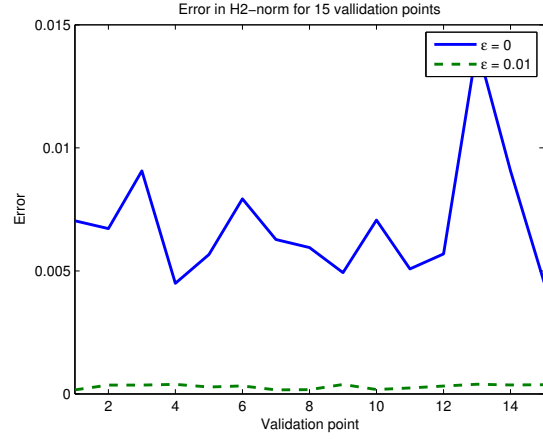


Fig. 3. \mathcal{H}_2 -norm in 15 validation points for different values of ϵ

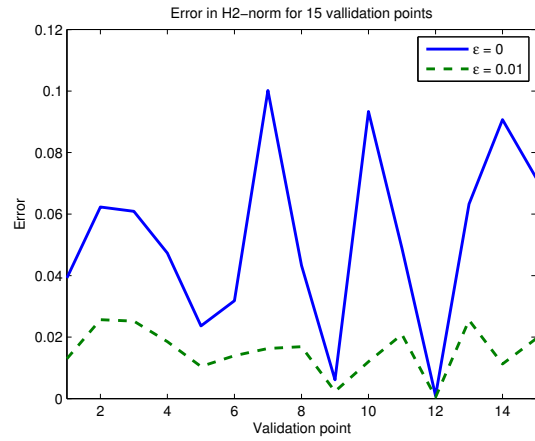


Fig. 4. \mathcal{H}_2 -norm in 15 validation points for different values of ϵ

VI. CONCLUSIONS

In this paper we first looked at the method proposed in [8], which is a new method for generating LPV-models. The core concept in this approach is to preserve input-output relations in the approximation, and not strive to match the actual numbers in the given state space models. The method has shown good properties on both academic examples and more realistic problems. The main contribution in this paper is the proposed regularization of the problem introduced in [8]. The regularization has, through the interpretation presented in the paper, connections to recently popularized worst-case approaches. We have shown how to calculate the original cost function and its gradient in an efficient way and extended these results to the regularized cost function and its gradient.

REFERENCES

- [1] 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.
- [2] M. Steinbuch, R. van de Molengraft, and A. van der Voort, "Experimental modelling and LPV control of a motion system," *American Control Conference, 2003. Proceedings of the 2003*, vol. 2, pp. 1374–1379, 4-6, 2003.

- [3] M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra, "LPV control for a wafer stage: beyond the theoretical solution," *Control Engineering Practice*, vol. 13, no. 2, pp. 231 – 245, 2005.
- [4] 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.
- [5] 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.
- [6] R. Tóth, "Modeling and identification of linear parameter-varying systems, an orthonormal basis function approach," Ph.D. dissertation, Delft University of Technology, 2008.
- [7] A. Ben-Tal and A. Nemirovski, "Robust Optimization - Methodology and Applications," *Mathematical Programming (Series B)*, vol. 92, pp. 453–480, 2002.
- [8] D. Petersson and J. Löfberg, "Optimization based LPV-approximation of multi-model systems," in *Proceedings of the European Control Conference 2009*, Aug. 2009, pp. 3172–3177.
- [9] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [10] P. Van Dooren, K. Gallivan, and P. Absil, " \mathcal{H}_2 -optimal model reduction of MIMO systems," *Appl. Math. Lett.*, 2008.
- [11] D. Wilson, "Optimum solution of model reduction problem," *Proc. Inst. Elec. Eng.*, vol. 117, pp. 1161 – 1165, 1970.
- [12] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer-Verlag, New York, Inc., 1999.