Preprints of the 17th IFAC Symposium on System Identification
Beijing International Convention Center
October 19-21, 2015. Beijing, China

IFT-LPV: Data-Based Tuning of Fixed Structure Controllers for LPV Systems

Sachin T. Navalkar * **Tom Oomen** ** **Jan-Willem van Wingerden** *

* *Delft Center for Systems and Control*
Delft University of Technology, 2628 CD Delft, the Netherlands
(e-mail for correspondence: S.T.Navalkar@tudelft.nl)
** *Control Systems Technology Group*
Eindhoven University of Technology
5600 MB Eindhoven, the Netherlands

Abstract: Fixed structure controllers are widely used, however the tuning thereof can be cumbersome and gives no guarantee of optimality, especially when the system is Linear Parameter-Varying (LPV). Iterative Feedback Tuning (IFT) is a technique for the optimisation of a parameterised controller based on closed-loop experiments. This paper extends the applicability of IFT to LPV systems for the case where the LPV scheduling parameters are measurable but cannot be controlled. The closed-loop LPV system matrices are factorised such that the effect of the scheduling parameter on the IFT gradient estimates can be compensated. A sufficient number of IFT experiments are performed to estimate the cost gradient and tune the parameters. The method is validated successfully via a simulation study for a special case with an LPV system.

Keywords: Data-based controller tuning, Adaptive Control, Iterative Feedback Tuning, LPV systems, Fixed structure control design.

1. INTRODUCTION

A number of industrial systems that demand closed-loop control can be characterised as Linear-Parameter-Varying (LPV). Significant attention has been focussed both on their identification (Lovera et al., 2013) and control (Emedi and Karimi, 2013) recently. Controller design without considering the LPV nature of the system can lead to severe impairment of performance. For instance, the aeroelastic flutter system model, as identified in van Wingerden et al. (2010) shows poles that vary from stable to unstable over the operating range. If a Linear Time Invariant (LTI) controller were to be used, the controller could cause poor performance or even instability.

In industrial applications, the challenges posed by an LPV plant are often dealt with by gain-scheduling the controller, (Leith and Leithead, 2000), (Shamma, 2012). In this method, a number of LTI controllers are designed at different system operating points, and the actual control input is interpolated based on the current operating conditions. For optimal classical control design (Wu, 1995), it may be necessary to have available a good description of system dynamics, uncertainties and disturbance characteristics, arriving at which may not be trivial in practical situations. For instance, LPV system identification literature, (van Wingerden et al., 2010), (Lovera et al., 2013), describes the so-called curse of dimensionality: as the complexity of an LPV system increase, there occurs a combinatorial explosion in the amount of data required to obtain a low-uncertainty model from system identification.

An alternative to conventional data-based system identification and controller synthesis is provided by Iterative Feedback Tuning (IFT) algorithms, Hjalmarsson (2002). Here, the parameters of a stabilising controller are tuned by optimising a cost function with gradient estimates obtained from input-output data. The practical applicability of IFT has been amply demonstrated in literature, e.g. Gevers (2002).

Although IFT has been applied successfully for LTI systems, it cannot be directly applied to tune controllers for LPV systems since the cost function gradient would be contaminated by the effect of the LPV scheduling variable. An alternative would be to bin together similar values of the scheduling variable and perform LTI IFT experiments to obtain a gain scheduling; however, this would require discretisation of the parameter space and a full exploration thereof at every iteration. Another alternative could be to describe the parameter-varying part of the system dynamics by a bounded uncertainty, and synthesise a robustly stabilising LTI controller using IFT, van der Velden et al. (2014). A robust switching controller tuned using IFT has been devised in Koumboulis et al. (2007) for adapting to varying system dynamics. However, current IFT techniques cannot be directly applied for tuning an LPV controller in closed loop with a fully LPV system with an arbitrarily varying scheduling sequence.

The key contribution of this paper is twofold: first, the IFT methodology is cast into a state-space form so that the extension to LPV systems is more accessible. Then, a methodology is proposed for using IFT for LPV systems.

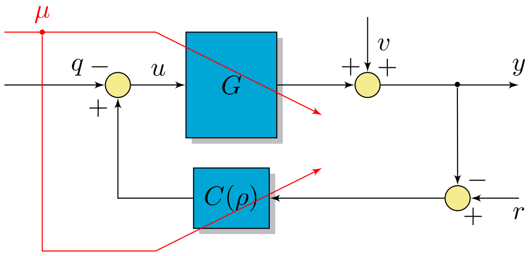

Fig. 1. Iterative Feedback Tuning experiment layout for an LPV controller in closed loop with an LPV plant.

The closed-loop matrices of the LPV system are decomposed into:

- A factor that depends only on the scheduling variable and is known but time-varying,
- A factor that depends only on the plant dynamics that is unknown but constant.

Sufficient experiments are then conducted so that the effect of the LPV scheduling variable on the IFT gradient estimates can be compensated for. Using these IFT cost gradients, the controller parameters will be updated to optimise their values. Thus, a full LPV extension to the IFT methodology is laid down and validated.

This section motivates the need for IFT for LPV systems. Section II describes the theoretical framework, which is validated in Section III. Finally, conclusions and directions for future work will be delineated in Section IV.

2. THEORETICAL FRAMEWORK

The objective of this section is to describe IFT methodology for using data to tune an LPV controller for an LPV system. The experiment layout is depicted in Fig. 1. The plant G is LPV and its dynamics depend on the scheduling variables μ. The output of the system y is perturbed by noise v and fed into the controller $C(\rho)$, which is also LPV. The controller is (non-linearly) parameterised such that ρ are the parameters to be tuned by IFT for optimal performance in tracking the reference r. As IFT uses a gradient-based optimisation routine, an additional control input q is used for gradient estimates; its use will be detailed in the subsections below.

2.1 Preliminaries and Notation

This subsection describes the notation and sets up the problem description that will be addressed by IFT for LPV systems. The plant denoted by G in the figure is taken to be LPV and to admit the following discrete-time description:

$$x_{k+1} = A_k x_k + B_k u_k, \tag{1}$$

$$y_k = C x_k + v_k, \tag{2}$$

where $x_k \in \mathcal{R}^{n_x}$ is the state at time k, $u_k \in \mathcal{R}^{n_u}$ is the control input, $y_k \in \mathcal{R}^{n_y}$ is the measured output and $v_k \in \mathcal{R}^{n_y}$ is the disturbance. The state transition matrix A_k and the input matrix B_k are taken as linear parameter-varying (LPV) and depend on the scheduling variables $\mu_k \in \mathcal{R}^{n_\mu}$ in the following manner:

$$A_k = A^{[0]} + \sum_{j_\mu=1}^{n_\mu} \mu_k^{[j_\mu]} A^{[j_\mu]}, \tag{3}$$

$$B_k = B^{[0]} + \sum_{j_\mu=1}^{n_\mu} \mu_k^{[j_\mu]} B^{[j_\mu]}. \tag{4}$$

The variable $\mu_k = [\mu_k^{[1]}, ..., \mu_k^{[n_\mu]}]^T$ varies arbitrarily and is assumed to be measurable. The output matrix C is treated as constant, and the feedthrough is taken to be zero. The extension to the case of LPV C and D matrices is straightforward, under the condition that either the plant or the controller is strictly proper.

It is assumed that there is a nominally stabilising controller in the loop, which is a function of a set of parameters $\rho_{j_\rho}, j_\rho = 1, ..., n_\rho$. Thus, the state-space realisation of the controller is as follows:

$$\xi_{k+1} = A_{c,k}(\rho)\xi_k + B_{c,k}(\rho)e_k, \tag{5}$$

$$u_k = C_{c,k}(\rho)\xi_k + D_{c,k}(\rho)e_k - q_k, \tag{6}$$

where $q_k \in \mathcal{R}$ is an auxiliary input that will be used for gradient estimation. The controller state-space matrices are affine functions of the scheduling variable μ_k:

$$A_{c,k}(\rho) = A_c^{[0]}(\rho) + \sum_{j_\mu=1}^{n_\mu} \mu_k^{[j_\mu]} A_c^{[j_\mu]}(\rho),$$

and similarly for $B_{c,k}(\rho)$, $C_{c,k}(\rho)$ and $D_{c,k}(\rho)$. Here, $\xi \in \mathcal{R}^{n_c}$ is the state vector of the controller, of length n_c. The input to the controller is the error $e_k \in \mathcal{R}^{n_y}$, defined as the deviation of the output from the reference r_k, thus $e_k = r_k - y_k$. The controller matrices are functions of the parameters ρ_{j_ρ}. The control objective is to tune these parameters to minimise the cost J which is taken as a weighted combination of the error and the control effort over one experiment,

$$J = \frac{1}{2N}(e^T e + \lambda u^T u). \tag{7}$$

Here, N is the length of the experiment and λ is a positive scalar that weights the control effort. The stacked error e is:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}, \tag{8}$$

with the stacked input u, the stacked output y, the stacked disturbance signal v and the stacked reference signal r defined in the same way. Since the algorithm is intended to minimise J over the components of ρ, it is required to be able to estimate the gradient of the cost with respect to the parameters, thus the quantity $\frac{\partial J}{\partial \rho}$. From equation (7),

$$\frac{\partial J}{\partial \rho} = \frac{1}{N}(\frac{\partial e^T}{\partial \rho} e + \lambda \frac{\partial u^T}{\partial \rho} u). \tag{9}$$

The quantities $\frac{\partial e^T}{\partial \rho}$ and $\frac{\partial u^T}{\partial \rho}$ can thus be estimated to optimise the controller parameters. For simplifying notation, the relations between stacked signals will be described using Toeplitz matrices. A Toeplitz matrix $\mathcal{T}$ for an LPV state-space system with realisation $(\mathcal{A}_k, \mathcal{B}_k, \mathcal{C}_k, \mathcal{D}_k)$ is defined as:

$$\mathcal{T}(\mathcal{A}_k, \mathcal{B}_k, \mathcal{C}_k, \mathcal{D}_k) = \begin{bmatrix} \mathcal{D}_1 & 0 & \cdots & 0 \\ \mathcal{C}_2\mathcal{B}_1 & \mathcal{D}_2 & \cdots & 0 \\ \mathcal{C}_3\mathcal{A}_2\mathcal{B}_1 & \mathcal{C}_3\mathcal{B}_2 & \cdots & 0 \\ \mathcal{C}_4\mathcal{A}_3\mathcal{A}_2\mathcal{B}_1 & \mathcal{C}_4\mathcal{A}_3\mathcal{B}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}_N\mathcal{A}_{N-1}...\mathcal{A}_2\mathcal{B}_1 & \mathcal{C}_N\mathcal{A}_{N-1}...\mathcal{A}_3\mathcal{B}_2 & \cdots & \mathcal{D}_N \end{bmatrix}.$$

The Toeplitz matrix of the plant and controller are:

$$T = \mathcal{T}(A_k, B_k, C, 0), \tag{10}$$

$$T_c = \mathcal{T}(A_{c,k}(\rho), B_{c,k}(\rho), C_{c,k}(\rho), D_{c,k}(\rho)). \tag{11}$$

Toeplitz matrices are also constructed for the closed loop transfer functions. Since all the open-loop Toeplitz matrices are lower triangular, Oomen et al. (2009), the sensitivity T_s, the complementary sensitivity T_{cs} and the process sensitivity T_{ps} Toeplitz matrices are:

$$T_s = (I + TT_c)^{-1}, \tag{12}$$

$$T_{cs} = (I + TT_c)^{-1}TT_c, \tag{13}$$

$$T_{ps} = (I + TT_c)^{-1}T. \tag{14}$$

Here, $I \in \mathcal{R}^{Nn_y \times Nn_y}$ is the identity matrix. If the plant and controller matrices are functions of μ_k as above, the closed-loop matrices $(\bar{A}_k, \bar{B}_k, \bar{C}, \bar{D})$ will be affine functions of μ_k and its cross-products over time (Chen and Francis, 1995). So, the number of affine terms in the expansion of $\bar{A}_k$ and $\bar{B}_k$ is $\bar{n}_\mu = 2^{n_\mu}$:

$$\bar{A}_k = \bar{A}^{[0]} + \sum_{j_\mu=1}^{\bar{n}_\mu} \bar{\mu}_k^{[j_\mu]} \bar{A}^{[j_\mu]}. \tag{15}$$

Here $\bar{\mu}$ is a scheduling variable that can be described as a combination of the original scheduling variable μ. A similar expression can be derived for $\bar{B}_k$ and for the other closed-loop transfer functions T_{cs} and T_{ps}.

The closed-loop Toeplitz matrices are thus affine functions of the (new) scheduling variables. The matrices will now be decomposed into a matrix that depends only on the scheduling variables and a matrix that only contains constant system parameters.

2.2 LPV Toeplitz Matrix Decomposition

The factorisation of the Toeplitz sensitivity matrix T_s is:

$$T_s = \underbrace{[M_{k,1}\ M_{k,2}\ \cdots\ M_{k,p}]}_{M(\mu)} \underbrace{\begin{bmatrix} \mathcal{H} & 0 & \cdots & 0 \\ 0 & \mathcal{H} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{H} \end{bmatrix}}_{\tilde{T}_s}. \tag{16}$$

The terms in the equation are explained below. Note that $\tilde{T}_s$ is still a function of ρ, but this dependency has not been explicitly mentioned for brevity. Since the Toeplitz matrices T_{cs} and T_{ps} show the same structure, they can also be factorised in the same way:

$$T_{cs} = M(\mu_k)\tilde{T}_{cs}, \tag{17}$$

$$T_{ps} = M(\mu_k)\tilde{T}_{ps}. \tag{18}$$

Thus, the closed-loop Toeplitz matrices are decomposed into one factor $M(\mu_k)$ that depends only on the known scheduling sequence and varies per experiment, and a factor $\tilde{T}_*$ ($*$ stands for cs, ps and s), which is constant across experiments but depends on the unknown system dynamics. The matrices $\mathcal{F}_{j_f}$ are defined such that:

$$\mathcal{F}_1 = \left[(\bar{D}^{[0]})^T \ \cdots \ (\bar{D}^{[\bar{n}_\mu]})^T\right]^T, \tag{19}$$

and the j_fth block row of $\mathcal{F}_{j_f} \in \mathcal{R}^{n_y \bar{n}_\mu^{j_f} \times n_u}$ is given by

$$\mathcal{F}_{j_f} = \bar{C}^{[i_1]} \bar{A}^{[i_2]} ... \bar{A}^{[i_{j_f-1}]} \bar{B}^{[i_{j_f}]}. \tag{20}$$

Here, $i_1, ... i_{j_f} \in \{1, ..., \bar{n}_\mu\}$. For each row z, the order is such that the following holds: $\sigma_{z+1} > \sigma_z$ where

$$\sigma_z = \left[i_1^z \ \cdots i_{j_f}^z\right] \begin{bmatrix} \bar{n}_\mu^{j_f-1} \\ \vdots \\ \bar{n}_\mu \\ 1 \end{bmatrix}. \tag{21}$$

The cross-product system matrices are stacked on top of each other to form the block column matrix $\mathcal{H} \in \mathcal{R}^{n_z \times n_u}$:

$$\mathcal{H} = \begin{bmatrix} \mathcal{F}_1 \\ \vdots \\ \mathcal{F}_p \end{bmatrix}. \tag{22}$$

The column size of $\mathcal{H}$ is $n_z = \sum_{i_z=1}^{N} n_y \bar{n}_\mu^{i_z}$. The known scheduling sequences with which these matrices multiply are collated in the following manner:

$$M_{k,1} = \begin{bmatrix} \mu_k^T & 0 & \cdots & 0 \\ 0 & \mu_k^T \otimes \mu_{k+1}^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_k^T \otimes ... \otimes \mu_{k+N-1}^T \end{bmatrix} \otimes I_{n_y},$$

$$M_{k,2} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \mu_k^T & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_k^T \otimes ... \otimes \mu_{k+N-2}^T & 0 \end{bmatrix} \otimes I_{n_y},$$

et cetera. It can be seen that the column size of $M(\mu_k) \in \mathcal{R}^{n_y \times n_z N}$ grows exponentially with N for a fully LPV system. For practical applications, it may become necessary to exploit structure for a factorisation of reduced size. Next, an IFT procedure is developed to tune the parameters ρ for an LPV system.

2.3 Experiment I

In the first experiment, the auxiliary input q_I is set to zero. Here, the subscripts I, II and III refer to the three sets of experiments required for IFT-LPV. The relation between the input and output in the controlled system is given as follows, assuming zero initial conditions:

$$y_I = T_I u_I + v_I \tag{23}$$

$$u_I = T_{c,I}(r - y_I). \tag{24}$$

Combining these two equations, we have:

$$y_I = T_I T_{c,I}(r - y_I) + v_I. \tag{25}$$

It is required to find the gradient of the error and the control input with respect to the controller parameters. Both of these gradients can be directly related to the gradient of the plant output with respect to the controller parameters, using (9). So, from the above equation, the gradient of the plant output with respect to one arbitrary controller parameter $\rho_{j_\rho}, j_\rho \in \{1, ..., n_\rho\}$ is given by

$$\frac{\partial y_I}{\partial \rho_{j_\rho}} = T_I \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I) - T_I T_{c,I} \frac{\partial y_I}{\partial \rho_{j_\rho}}, \quad (26)$$

$$\frac{\partial y_I}{\partial \rho_{j_\rho}} = (I + T_I T_{c,I})^{-1} T_I \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I). \quad (27)$$

Using (12)-(14) in (25) and (27):

$$\frac{\partial y_I}{\partial \rho_{j_\rho}} = T_{ps,I} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I), \quad (28)$$

$$y_I = T_{cs,I} r + T_{s,I} v_I. \quad (29)$$

In order to remove the dependence of the Toeplitz matrices on the scheduling variable μ_I, the Toeplitz matrices are factorised as in equation (16).

$$\frac{\partial y_I}{\partial \rho_{j_\rho}} = M(\mu_I) \tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I), \quad (30)$$

$$y_I = M(\mu_I) \tilde{T}_{cs} r + M(\mu_I) \tilde{T}_s v_I. \quad (31)$$

2.4 Experiment Set II

In this set of experiments, the quantity $\frac{\partial y_I}{\partial \rho_{j_\rho}}$ will be estimated. For this, the auxiliary input q_{II} in Fig. 1 is made use of, and, as in Hjalmarsson (2002), it is set to be:

$$q_{II} = \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I). \quad (32)$$

With the introduction of this auxiliary input, the closed-loop system equations become:

$$y_{II} = T_{cs,II} r - T_{ps,II} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I) + T_{s,II} v_{II}. \quad (33)$$

As before, a factorisation is done to remove the dependence of the closed-loop Toeplitz matrices on μ_{II}:

$$y_{II} = M(\mu_{II}) \tilde{T}_{cs} r - M(\mu_{II}) \tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I) + M(\mu_{II}) \tilde{T}_s v_{II}. \quad (34)$$

It can be seen that to obtain the desired term $\tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I)$ from the data, it would be necessary to have the matrix $M(\mu_{II})$ available and to compute a left-inverse. However if a single experiment is done, this matrix may be rank-deficient. So, a number of additional experiments are required to be performed to obtain an overdetermined set of equations and estimate this gradient.

$$y_{II}^{(\hat{j})} = M(\mu_{II}^{(\hat{j})}) \tilde{T}_{cs} r - M(\mu_{II}^{(\hat{j})}) \tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I) \quad (35)$$

$$+ M(\mu_{II}^{(\hat{j})}) \tilde{T}_s v_{II}^{(\hat{j})}, \quad (36)$$

where $\hat{j}$ is the experiment number in this set. To remove its effect, the disturbance term is rearranged as follows:

$$M(\mu_I I^{(\hat{j})}) \tilde{T}_s v_I I^{(\hat{j})} = \tilde{\mathcal{T}}_s w_{II}^{(\hat{j})}. \quad (37)$$

Here, the new disturbance vector given by

$$w_{II}^{(\hat{j})} = \sum_{\ell=1}^{N} v_{II,\ell}^{(\hat{j})} M_{k,\ell}(\mu_{II}^{(\hat{j})}), \quad (38)$$

is a random variable assuming that μ_k is distributed randomly and $\tilde{\mathcal{T}}_s$ is constant. Thus, equation (36) becomes

$$y_{II}^{(\hat{j})} = M(\mu_{II}^{(\hat{j})}) \tilde{T}_{cs} r - M(\mu_{II}^{(\hat{j})}) \tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I) + \tilde{\mathcal{T}}_s w_{II}^{(\hat{j})}. \quad (39)$$

To remove the influence of the external disturbance, the estimate of the difference of the output, $\Delta y_{II}^{(\hat{j})}$ is used:

$$\Delta y_{II}^{(\hat{j})} = \text{est}\{2 y_{II}^{(\hat{j})} - y_{II}^{(1)}\}, \quad (40)$$

$$\Delta y_{II}^{(\hat{j})} = (2M(\mu_{II}^{(\hat{j})}) - M(\mu_{II}^{(1)})) \tilde{T}_{cs} r \quad (41)$$

$$- (2M(\mu_{II}^{(\hat{j})}) - M(\mu_{II}^{(1)})) \tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I) \quad (42)$$

The output from all experiments is now stacked as Y_{II}:

$$Y_{II} = \begin{bmatrix} \Delta y_{II}^{(2)} \\ \Delta y_{II}^{(3)} \\ \vdots \end{bmatrix}. \quad (43)$$

This vector can be obtained from equation (42) as:

$$Y_{II} = \tilde{M}_{II} (\tilde{T}_{cs} r - \tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I)), \quad (44)$$

where $\tilde{M}_{II}$ is built by stacking matrices $(2M(\mu_{II}^{(\hat{j})}) - M(\mu_{II}^{(1)}))$. To compensate for the influence of the scheduling variables $\mu_{II}^{(\hat{j})}$, it is required to premultiply the equation above with the left-inverse of this matrix, which exists and is well-conditioned only for sufficiently large number of experiments. We now have:

$$\tilde{M}_{II}^{\dagger} Y_{II} = \tilde{T}_{cs} r - \tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I) \quad (45)$$

where $*^{\dagger}$ is the left-(pseudo-)inverse of a matrix. Finally, a third set of experiments is necessary to eliminate the influence of the reference signal on the gradient.

2.5 Experiment Set III

The experiments above are repeated, taking q_{III} to be zero so that the output is influenced only by the reference signal. We obtain the following result:

$$\tilde{M}_{III}^{\dagger} Y_{III} = \tilde{T}_{cs} r. \quad (46)$$

Subtracting equation (45) from the equation above:

$$\tilde{M}_{III}^{\dagger} Y_{III} - \tilde{M}_{II}^{\dagger} Y_{II} = \tilde{T}_{ps} \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I). \quad (47)$$

Substituting this in equation (28), the gradient of the output with respect to ρ_{j_ρ} is given as:

$$\frac{\partial y_I}{\partial \rho_{j_\rho}} = M(\mu_I)(\tilde{M}_{III}^{\dagger} Y_{III} - \tilde{M}_{II}^{\dagger} Y_{II}). \quad (48)$$

Now, the required gradients $\frac{\partial e_I}{\partial \rho_{j_\rho}}$ and $\frac{\partial u_I}{\partial \rho_{j_\rho}}$ are obtained:

$$\frac{\partial e_I}{\partial \rho_{j_\rho}} = -\frac{\partial y_I}{\partial \rho_{j_\rho}}, \quad (49)$$

$$\frac{\partial u_I}{\partial \rho_{j_\rho}} = \frac{\partial T_{c,I}}{\partial \rho_{j_\rho}}(r - y_I) - T_{c,I} \frac{\partial y_I}{\partial \rho_{j_\rho}}. \quad (50)$$

So, the (unbiased) gradient of the cost function is:

$$\frac{\partial J}{\partial \rho_{j_\rho}} = \frac{1}{N}\left(\frac{\partial e_I^T}{\partial \rho_{j_\rho}} e_I + \lambda \frac{\partial u_I^T}{\partial \rho_{j_\rho}} u_I\right). \tag{51}$$

Thus, the gradient with respect to each parameter can be estimated. With these gradients, a standard optimisation technique, such as steepest descent, could be used to optimise the controller parameters. As the cost function will typically be non-convex, IFT-LPV will find a local minimum. To test the IFT-LPV, we will make use of a special case, as described next.

2.6 Factorisation for a Special Case

The key step in the IFT experiments above is the construction of the $M(\mu)$ matrix. As seen, the column size of $M(\mu)$ grows exponentially with the complexity of the LPV plant and the length of the experiment, possibly leading to an exponential increase in the number of experiments required to estimate the IFT gradients uniquely. For faster learning, it becomes imperative to exploit structural knowledge of either the LPV plant or the scheduling signal.

A factorisation that is more compact and practically tractable is set up for the special case of a switched system whose dynamics change from $(A^{[0]}, B^{[0]}, C, 0)$ to $(A^{[0]} + A^{[1]}, B^{[0]} + B^{[1]}, C, 0)$ at an arbitrary instant of time p during each experiment. In other words, μ_k is:

$$\mu_k = 0, \qquad k \leq p, \tag{52}$$

$$\mu_k = 1, \qquad k > p, \tag{53}$$

where $p \in \{1, ..., N\}$. A much simpler alternative factorisation can substitute equation (16). Let us consider an LPV controller in the loop. The sensitivity function has the state-space realisation $(\bar{A}^{[0]}, \bar{B}^{[0]}, \bar{C}, \bar{D})$ at and before time instant p. After time instant p, the new state transition becomes $\bar{A}' = \bar{A}^{[0]} + \bar{A}^{[1]}$, with a similar relation holding for the new input matrix $\bar{B}'$. The Toeplitz matrix of the closed-loop sensitivity can thus be written as:

$$T_s(\mu_k) = \begin{bmatrix} \bar{D} & 0 & \cdots & 0 \\ \bar{C}\bar{B}^{[0]} & \bar{D} & \cdots & 0 \\ \bar{C}\bar{A}^{[0]}\bar{B}^{[0]} & \bar{C}\bar{B}^{[0]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}(\bar{A}^{[0]})^{p-1}\bar{B}^{[0]} & \vdots & \cdots & 0 \\ \bar{C}(\bar{A}')(\bar{A}^{[0]})^{p-1}(\bar{B}^{[0]}) & \vdots & \cdots & 0 \\ \bar{C}(\bar{A}')^2(\bar{A}^{[0]})^{p-1}(\bar{B}^{[0]}) & \vdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}(\bar{A}')^{(N-p)}(\bar{A}^{[0]})^{p-1}(\bar{B}^{[0]}) & \cdots & \cdots & \bar{D} \end{bmatrix}. \tag{54}$$

The Toeplitz is approximately factorised as the sum of the initial Toeplitz matrix $T_{0,s}$ and a modified Toeplitz matrix $T_{1,s}$ pre-multiplied by $Q_{N,p}$, which depends on p:

$$T_s(\mu_k) \approx \underbrace{[I_N \ Q_{N,p}]}_{M^*(\mu_k)} \underbrace{\begin{bmatrix} T_{0,s} \\ T_{1,s} \end{bmatrix}}_{\tilde{T}_s^*}, \tag{55}$$

$$Q_{N,p} = \text{diag}(\underbrace{0, 0, ..., 0}_{\text{p terms}}, \underbrace{1, 1, ..., 1}_{\text{(N-p terms)}}). \tag{56}$$

and the unknown system matrices are:

$$T_{0,s} = \begin{bmatrix} \bar{D} & 0 & \cdots & 0 \\ \bar{C}\bar{B}^{[0]} & \bar{D} & \cdots & 0 \\ \bar{C}\bar{A}^{[0]}\bar{B}^{[0]} & \bar{C}\bar{B}^{[0]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}(\bar{A}^{[0]})^{N-1}\bar{B}^{[0]} & \cdots & \cdots & \bar{D} \end{bmatrix} \tag{57}$$

$$T_{1,s} = \begin{bmatrix} \bar{D} & 0 & \cdots & 0 \\ \bar{C}\bar{B}^{[1]} & \bar{D} & \cdots & 0 \\ \bar{C}\bar{A}^{[1]}\bar{B}^{[0]} & \bar{C}\bar{B}^{[1]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}(\bar{A}^{[1]})^{N-\bar{p}}(\bar{A}^{[0]})^{\bar{p}-1}\bar{B}^{[0]} & \cdots & \cdots & \bar{D} \end{bmatrix} \tag{58}$$

where $\bar{p}$ is the mean of all p. Here, $M^*(\mu_k)$ depends on p and is known but different for each experiment, while $\tilde{T}_s^*$ depends on the unknown system parameters but remains. This special case will be simulated next.

3. SIMULATION STUDY

A SISO morphing aerofoil model (Lee and Singh, 2007), with state dimension 4, is used as a test case. An actuated flap at the trailing edge induces changes in displacement. The dynamics depend strongly on wind speed V $(= \mu)$.

An operating wind speed of 6 m/s is chosen with a sudden change of 2 m/s occuring at an arbitrary measurable instant of time. The factorisation (55) will be used for IFT tuning. Each experiment lasts 100 samples and the reference signal is given in Fig. 3.

A PI controller was used in closed loop with the system, initialised at an arbitrarily poor value of proportional and integral gains K_p and K_i. Both K_p and K_i are taken to be affine in the scheduling variable wind speed V_k:

$$K_p = K_{p0} + V_k K_{p1}, \qquad K_i = K_{i0} + V_k K_{i1}. \tag{59}$$

Thus, four parameters were required to be optimised using the IFT-LPV technique. The total number of experiments done per iteration was 11. In Fig. 2, it can be seen that the method is able to tune all four parameters simultaneously. The simulation was done with an external disturbance with an SNR of 20. While the convergence and variance of the method increases with the additional unmeasured disturbance, the method is able to deliver an unbiased estimate of the optimal controller parameters. There is a direct trade-off between the two quantities: increasing the rate of convergence increases the variance of the optimal parameter estimate.

The improvement in performance can be clearly observed in Fig. 3. Since the initial controller is arbitrary, its performance is inadequate, but IFT-LPV is able to converge to an optimal controller.

The additional advantage of tuning an LPV controller for use with an LPV system can be seen in Fig. 4. The LTI controller cannot track changes in system dynamics and gives poor performance for a high wind speed, although it settles at the correct reference. However, the response of the system controlled by an LPV controller varies minimally with the operating point. Thus, IFT-LPV can tune an LPV closed loop, without any knowledge of true system dynamics, as opposed to conventional LPV control techniques that require a system model as a starting point.

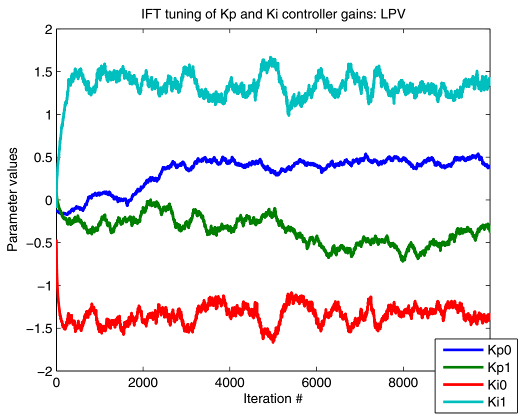

Fig. 2. IFT tuning of Kp and Ki parameters, SNR 20.

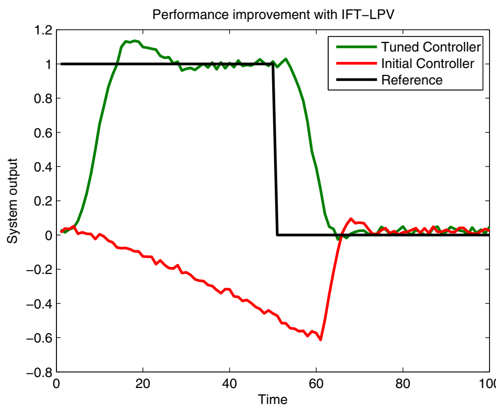

Fig. 3. Comparison of controller performance before and after data-based controller tuning with IFT.

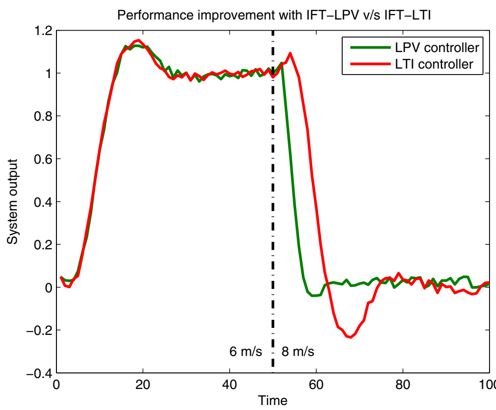

Fig. 4. Comparison of LTI and LPV controllers.

4. CONCLUSION

The data-based fixed structure controller tuning method IFT-LPV is proposed in this paper for LPV systems. Since the dynamics of LPV systems depend on the instantaneous value of the arbitrarily varying scheduling variable, traditional IFT experiments cannot be used to tune controller parameters, so a factorisation is developed with which IFT for LPV systems was made possible. A simulation study validated the theory. IFT-LPV suffers the same curse of dimensionality as is observed in the system identification of LPV systems. For practical applications, it would be necessary to exploit the structure in the LPV system to make the problem tractable, by achieving a reduced-order factorisation of the LPV matrices, and thereby, a reduced number of experiments for IFT tuning. For instance, Navalkar and van Wingerden (2015) exploits the feedforward nature of the problem to apply IFT for tuning an output-LPV controller for the load control of a wind turbine, approximated as an output-LPV system. This paper demonstrates the concept of a fully data-based LPV controller tuning algorithm that is capable of optimising performance for an LPV system with no a priori knowledge of system dynamics or scheduling.

REFERENCES

Chen, T. and Francis, B.A. (1995). Optimal sampled-data control systems. *Springer.*

Emedi, Z. and Karimi, A. (2013). Fixed-order LPV controller design for LPV systems by convex optimisation. *5th IFAC Symposium SSC*, Grenoble, France.

Gevers, M. (2002). A decade of progress in iterative process control design: From theory to practice. *Journal of Process Control*, 12(4), 519–531.

Hjalmarsson, H. (2002). Iterative feedback tuning - an overview. *Journal of Adaptive Control and Signal Processing*, 16(5), 373–395.

Koumboulis, F.N., Tzamtzi, M.P., and Chamilothoris, G.E. (2007). Multivariable iterative feedback tuning - a step-wise safe switching approach. *IEEE Int. Conf. ETFA*, 1354–1363.

Lee, K.W. and Singh, S.N. (2007). Global robust control of an aeroelastic system using output feedback. *Journal of Guidance, Control and Dynamics*, 30(1), 271–275.

Leith, D.J. and Leithead, W.E. (2000). Survey of gain-scheduling analysis and design. *International Journal of Control*, 73, 1001–1025.

Lovera, M., Bergamasco, M., and Casella, F. (2013). LPV modelling and identification: An overview. *Lecture Notes in Control and Information Sciences*, 437, 3–24.

Navalkar, S.T. and van Wingerden, J.W. (2015). Iterative feedback tuning of an LPV feedforward controller for wind turbine load alleviation. *IFAC Workshop on Linear Parameter Varying Systems, Grenoble, France.*

Oomen, T., van de Wijdeven, J., and Bosgra, O. (2009). Suppressing intersample behaviour in iterative learning control. *Automatica*, 45, 981–988.

Shamma, J.S. (2012). An overview of LPV systems. *Control of linear parameter varying systems with applications*, Springer US, 3–26.

van der Velden, B.J.C.H., Oomen, T., and Heertjes, M.F. (2014). Constrained iterative feedback tuning for robust high-precision motion control. *IFAC World Congress.*

van Wingerden, J.W., Gebraad, P., and Verhaegen, M. (2010). LPV identification of an aeroelastic flutter model. *Proceedings of the IEEE Conference on Decision and Control*, Atlanta, USA.

Wu, F. (1995). *Control of Linear Parameter Varying Systems.* University of California at Berkeley, PhD Dissertation.