LPV Modelling and Control of Burgers’ Equation

Seyed Mahdi Hashemi and Herbert Werner

Institute of Control Systems, Hamburg University of Technology,
Eissendorfer Str. 40, 21073 Hamburg, Germany,
(e-mail: seyed.hashemi@tu-harburg.de, h.werner@tu-harburg.de)

Abstract: Linear parameter-varying (LPV) modelling and control of a nonlinear PDE is presented in this paper. The one-dimensional viscous Burgers’ equation is discretized using a finite difference scheme and the boundary conditions are taken as control inputs. A nonlinear high-order state space model is generated and proper orthogonal decomposition is used for model order reduction and the accuracy of the reduced model is verified. A discrete-time quasi-LPV model that is affine in scheduling parameters is derived based on the reduced model and a polytopic LPV controller is synthesized. A low-order functional observer is designed to estimate the scheduling parameters required for LPV controller. Simulation results demonstrate the high tracking performance and disturbance and measurement noise rejection capabilities of the designed LPV controller comparing with an LQG controller based on a linearized model.

1. INTRODUCTION

In recent years, there has been growing interest in modelling and control of distributed systems and flow control is a motivating application in this field. Due to the complexity of the governing partial differential equation (PDE) i.e. the Navier-Stokes equation, simpler analog models e.g. Burgers’ equation can be used to attack the control problem first and then try to extend the solution to more general cases, see Atwell et al. [2001] and Effe and Ozbay [2004]. Burgers’ equation includes the nonlinear convective term that is challenging to handle in flow-related problems. Besides, supersonic flow about airfoils, shockwaves, some boundary layer problems and traffic flows can be modeled by Burgers’ equation.

Obtaining a finite dimensional approximation of distributed parameter systems via Galerkin projection (Holmes et al. [1996]) has become popular in the literature. Using proper orthogonal decomposition (POD) to obtain optimal basis functions leads to low-order models which can represent the original system reasonably. This technique has been frequently used to obtain a suitable model of Burgers’ equation for controller synthesis e.g. by Atwell et al. [2001], Effe and Ozbay [2004], Lawrence et al. [2005] and Djouadi et al. [2008]. However, linear controllers designed in these reports were based on locally linearized models. Stabilization of a family of stationary solutions of Burgers’ equation using output feedback was reported by Kristic et al. [2009].

Linear parameter-varying (LPV) gain-scheduling techniques have been developed into effective tools to control MIMO nonlinear plants. Their attractiveness lies in the extension of well known linear optimal control methods and the use of linear matrix inequalities (LMIs), to the solution of nonlinear control problems. Many nonlinear systems of practical interest can be represented as quasi-LPV systems, where the scheduling parameters include external signals and measured outputs or states, see e.g. Rugh and Shamma [2000]. However, the application of these techniques have not been extensively explored in control of PDEs. Control of a nonlinear Galerkin model using an adaptation-based LPV model was presented in Kasnakoglu [2010].

Polytopic LPV controllers (Apkarian et al. [1995]) have been proven to be effective and practical due to the simplicity of synthesis and implementation and low off-line computation effort. However, overbounding in the convex scheduling parameter set may lead to conservative controller design, and the number of LMIs to be solved for standard $\mathcal{H}_\infty$ controller synthesis increases exponentially with the number of scheduling parameters. Parameter set mapping proposed by Kwiatkowski and Werner [2008] was successfully used to reduce the number of scheduling parameters of a robotic manipulator LPV model (Hashemi et al. [2009]). The procedure to obtain the reduction mapping in this technique is similar to calculating POD basis functions. In both, the reduction transformations are extracted from experimental data or numerical simulations.

This paper presents the application of POD to obtain a low-order quasi-LPV model of Burgers’ equation together with controller and observer design. Firstly, one-dimensional Burgers’ equation with Dirichlet boundary conditions is discretized applying an explicit finite difference technique, and a discrete-time nonlinear state space system with a high order is obtained where boundary conditions are taken as control inputs.

The model is simulated for a typical input trajectory and method of snapshots (Sirovich [1987]) is used to obtain POD basis functions which form the model reduction transformation. Galerkin projection is then employed to map the original system to a lower-dimensional space. A polytopic quasi-LPV model is derived and the states of the reduced model are chosen as scheduling parameters. Since the derived quasi-LPV model has a small number
of scheduling parameters and vertices, an output-feedback LPV controller with fixed Lyapunov function is designed which has a low on-line computation cost. The LPV controller needs on-line access to the scheduling parameters. Thus, a low-order functional observer is designed using the method proposed by Abbaszadeh and Marquez [2007] to estimate the scheduling parameters.

The simulation results illustrate the high tracking performance and measurement noise rejection capability of the designed LPV controller. A comparison of disturbance rejection by the LPV controller and an LQG controller based on a linearized reduced model indicates that the designed LPV controller rejects the disturbances faster and attenuates the measurement noise significantly better.

The key feature of this paper is to demonstrate the applicability of LPV gain-scheduling techniques for modelling and control of Burgers’ equation which can be a starting point for more general flow control problems. The proposed nonlinear functional observer makes the implementation of the LPV controller possible by only a few measurements.

This paper is organized as follows: Discretization of Burgers’ equation and derivation of the nonlinear state space model is presented in section 2. In Section 3, application of POD and accuracy of the model is discussed. The LPV modelling and controller synthesis is described in Section 4. Nonlinear observer design is discussed in Section 5. Simulation results are given in Section 6 and the last section draws some conclusions.

2. NONLINEAR MODELLING

Consider the one-dimensional nonlinear viscous Burgers’ equation on the physical domain \( S = \{ s | s \in [0, L] \} \) and temporal domain \( T = \{ t | t \in [0, T] \} \)

\[
\frac{\partial w}{\partial t} + \frac{\partial w}{\partial s} + \nu \frac{\partial^2 w}{\partial s^2} = 0,
\]

where \( w(s,t) : S \times T \rightarrow \mathbb{R} \) is the space and time dependent velocity, \( s \) and \( t \) refer to space and time respectively and \( \nu \) is a known constant representing viscosity. The initial condition is given as

\[
w(s,0) = w_0(s),
\]

and Dirichlet boundary conditions are applied at both ends

\[
w(0,t) = u_1(t), \quad w(L,t) = u_2(t),
\]

where \( u_1(t) \) and \( u_2(t) \) are taken as control inputs.

An approximate discrete solution of (1) is represented by

\[
w_i^k = w(\hat{s}_i, \hat{t}_k) : \hat{S} \times \hat{T} \rightarrow \mathbb{W},
\]

with finite discrete sets

\[
\hat{S} = \{ s_1, \ldots, s_G \}, \quad \hat{T} = \{ t_1, \ldots, t_K \},
\]

where \( G \) is the number of grid points and \( K \) is the number of time samples. Using the forward-time central-space (FTCS) method which is an explicit finite difference scheme, (1) is discretized as

\[
w_{i+1}^k = w_i^k - \frac{\Delta t}{\Delta s} \alpha_i \left( \frac{w_{i+1}^k - w_{i-1}^k}{2} + r \left( w_{i+1}^k - 2w_i^k + w_{i-1}^k \right) \right),
\]

where \( \lambda = \frac{\Delta t}{\Delta s}, \quad r = \frac{\nu \Delta t}{(\Delta s)^2} \). \( \Delta s \) is the spatial grid size, \( \Delta t \) is the time-stepping and \( i \) and \( k \) represent the space and time indices respectively. As explained in Thomas [1998], choosing some appropriate values for \( \Delta s \) and \( \Delta t \) leads to convergence of this scheme if \( \nu \) is not very small.

The difference equation (6) is used to obtain the nonlinear discrete-time state space model

\[
x(k + 1) = A_n x(k) + F_n (x(k), u(k))
\]

\[
y(k) = C_n x(k),
\]

where the state vector \( x(k) \in \mathbb{R}^N \), the control input \( u(k) \in \mathbb{R}^{n_u} \) and the control output \( y(k) \in \mathbb{R}^{n_Y} \) are defined as

\[
x(k) = [x_1(k), \ldots, x_N(k)]^T = [w_1^k, \ldots, w_{G-1}^k]^T,
\]

\[
u(k) = [w_1^k, w_{G-1}^k]^T, \quad y(k) = [w_{N_1}^k, w_{N_2}^k]^T,
\]

and \( N = G - 2, n_1 = 2 \) and \( n_0 = 2 \) are numbers of states, inputs and outputs respectively. The matrices \( A_n \) and \( C_n \) and the vector \( F_n \) are given as

\[
A_n = \begin{bmatrix}
1 - 2r & r & 0 & \cdots & \cdots & 0 \\
0 & 1 - 2r & r & 0 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \cdots \\
0 & \cdots & \cdots & 0 & r & 0 \\
0 & \cdots & \cdots & 0 & 1 - 2r & r
\end{bmatrix},
\]

\[
F_n = \begin{bmatrix}
\frac{\lambda}{2} x_2 x_1 + \frac{\lambda}{2} x_1 u_1 + ru_1 \\
\frac{\lambda}{2} x_2 x_1 - \frac{\lambda}{2} x_3 x_2 \\
\frac{\lambda}{2} x_3 x_2 - \frac{\lambda}{2} x_4 x_3 \\
\vdots \\
\frac{\lambda}{2} x_{G-3} x_{G-4} - \frac{\lambda}{2} x_N x_{G-3} \\
\frac{\lambda}{2} x_{G-2} x_{G-3} - \frac{\lambda}{2} x_N x_{G-2} + ru_2
\end{bmatrix},
\]

\[
C_n = \begin{bmatrix}
0_{1 \times (N_1 - 1)} & 1 & 0_{1 \times (N_2 - 1)} \\
0_{1 \times (N_1 - 1)} & 1 & 0_{1 \times (N_2 - 1)}
\end{bmatrix}.
\]

The subscripts \( N_1 \) and \( N_2 \) specify the two grid points whose velocities are assumed to be measurable. The non-linearities in \( F_n \) come from the nonlinear convective term which is common in all flow problems.

The objective is to design a controller that computes the input signals \( u(k) \) such that the control output \( y(k) \) tracks a reference input and rejects process disturbance and measurement noise. The order of the nonlinear model (7) is typically too large to be used for controller synthesis and should be reduced.

3. PROPER ORTHOGONAL DECOMPOSITION

POD is a method of extracting orthonormal basis functions from an ensemble of experimental or simulation data which form a model reduction transformation. A solution of (1) is approximated in terms of a set of basis functions

\[
w(\hat{s}, \hat{t}) \approx \hat{w}(\hat{s}, \hat{t}) = \sum_{j=1}^M \phi_j(\hat{s}) \alpha_j(\hat{t}),
\]
where \( \phi_j \) define the set of orthonormal basis functions and \( \alpha_j \) are time-dependent coefficients.

The method of snapshots Sirovich [1987] is used to obtain the basis functions. The discretized model (6) is simulated for some typical boundary condition trajectories to form the matrix of snapshots \( W_{\text{snap}} \in \mathbb{R}^{N \times K} \)

\[
W_{\text{snap}} = \begin{bmatrix} w_1^1 & \ldots & w_1^K \\ \vdots & \ddots & \vdots \\ w_G^1 & \ldots & w_G^K \end{bmatrix}.
\]  

(11)

Introduce the singular value decomposition of \( W_{\text{snap}} \)

\[
W_{\text{snap}} = \Phi \Sigma \Psi^T = [\Phi_\nu \Phi_n] \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_n \end{bmatrix} \begin{bmatrix} \Psi_n^T \\ \Psi_n \end{bmatrix},
\]  

(12)

where \( \Phi \in \mathbb{R}^{N \times N} \) and \( \Psi \in \mathbb{R}^{K \times K} \) and \( \Phi_n, \Sigma_n \) and \( \Psi_n \) correspond to the \( M \) dominant singular values.

The columns of \( \Phi \) form the set of basis functions \{\phi_1, \ldots, \phi_N\} which can be used to obtain an accurate low-order dynamic model via Galerkin projection. Such a projection captures the most energy and properties of the original system for a given number of modes or basis function Holmes et al. [1996]. The basis functions obtained by this method are also called POD modes.

If \( M = N \), the original system is perfectly expressed in terms of the basis functions. However, a smaller number of dominant basis functions can be chosen according to the percentage of captured energy criterion

\[
E = \frac{\sum_{i=1}^{M} \sigma_i^2}{\sum_{i=1}^{N} \sigma_i^2}.
\]  

(13)

A higher \( E \) means that the reduced model captures the information contained in the snapshots better.

For state space models such as (7), the projection is simplified to multiplying both sides of (7) by truncated orthonormal matrix \( \Phi_s \in \mathbb{R}^{N \times M} \), see Astrid et al. [2002]

\[
\Phi_s^T x(k+1) = \Phi_s^T A_n x(k) + \Phi_s^T F_n (x(k), u(k)).
\]  

(14)

A reduced state vector \( x_r(k) \in \mathbb{R}^M \) is then defined as

\[
\hat{x}(k) = \Phi_s x_r(k)
\]

\[
x_r(k) = \Phi_s^T \hat{x}(k),
\]  

(15)

where \( \hat{x} \in \mathbb{R}^N \) is the approximate state vector in the original dimension

\[
\hat{x} = [\hat{x}_1, \ldots, \hat{x}_N]^T = [\hat{w}_1^1, \ldots, \hat{w}_K^1]^T.
\]  

(16)

Note that each element of \( x_r \) is a linear combination of \( \hat{x} \). Substituting \( x \) by \( \hat{x} \) in (14) and in the output equation of (7) yields

\[
x_r(k+1) = A_r x_r(k) + F_r (\Phi_s x_r(k), u(k)),
\]

\[
y(k) = C_r x_r(k),
\]  

(17)

with

\[
A_r = \Phi_s^T A_n \Phi_s, \quad F_r = \Phi_s^T F_n, \quad C_r = C_n \Phi_s.
\]  

(18)

\( A_r \) and \( C_r \) can be calculated offline, provided that basis functions have been determined. Since \( F_r \) is a function of \( x_r \) and \( u \), it should be calculated on-line if needed.

The procedure of POD is applied to Burgers’ equation (1) with \( \nu = 0.01, \Delta s = 0.02, \Delta T = 0.005, \; L = 1 \) and \( w_0(s) = 0 \). Both boundary conditions for generating the matrix of snapshots are chosen as sinusoidal excitation trajectories covering frequencies up to 50 Hz, similar to the report by Effe and Ozbay [2004]. In the simulation, \( K = 20000 \) and \( G = 51 \) are taken.

The reliability of the resulting reduced-order model (17) strongly depends on the excitation signal; if the operating conditions become different from excitation trajectories, the model accuracy will degrade.

Fig. 1 demonstrates the percentage of captured energy versus different number of chosen dominant POD modes. About 50% of the dynamical information is captured by the first mode and 81% by the first two modes. The validation results for these choices are not satisfactory, while selection of three first modes which captures 93% of the information leads to a reasonably accurate model for controller synthesis purposes. Fig. 2 shows the open-loop simulation results of the full-order and the reduced-order models for some typical boundary conditions.

4. LPV MODELLING AND CONTROL

In this section, a low-order LPV model is derived and controller synthesis is briefly reviewed and applied.

4.1 Polytopic LPV Model

The nonlinear discrete-time reduced-order model (17) can be converted to a polytopic quasi-LPV model

\[
x_r(k+1) = A(\theta(k)) x_r(k) + B(\theta(k)) u(k),
\]

\[
y(k) = C x_r(k),
\]  

(19)

Fig. 1. Percentage of captured energy

Fig. 2. Open-loop simulation for full-order model (upper) and reduced-order model (lower)

Although the resulting model (17) has a low order, it has a nonlinear term \( F_r \) which makes the controller synthesis challenging.
where parameter dependent matrices $A(\theta) \in \mathbb{R}^{M \times M}$ and $B(\theta) \in \mathbb{R}^{M \times n}$, and constant matrix $C \in \mathbb{R}^{n \times M}$ are to be determined and $\theta(k)$ is the scheduling parameter vector. Without loss of generality, assume that $\theta \in \mathbb{R}^M$. The LPV model can be represented by a linear input-output map

$$
P(\theta) = \begin{bmatrix} A(\theta) & B(\theta) \\ C & 0 \end{bmatrix}.
$$

Consider the compact set $\mathcal{P}_\theta \subset \mathbb{R}^M : \theta(k) \in \mathcal{P}_\theta, \forall k > 0$. Here, this set is assumed to be a polytope defined by the convex hull

$$
\mathcal{P}_\theta := \text{Co}\{\theta_{v_1}, \theta_{v_2}, \ldots, \theta_{v_m}\},
$$

where $m = 2^M$ is the number of vertices.

The LPV system is called parameter-affine, if the state space model depends affinely on the parameters

$$
P(\theta) = \sum_{i=0}^M \theta_i P_i = P_0 + \theta_1 P_1 + \cdots + \theta_M P_M.
$$

Since $\theta$ can be expressed as a convex combination of $m$ vertices $\theta_{v_i}$, if (22) holds, it follows that the system can be represented by a linear combination of LTI models at the vertices; this is called a polytopic LPV system

$$
P(\theta) \in \text{Co}\{P(\theta_{v_1}), P(\theta_{v_2}), \ldots, P(\theta_{v_m})\} = \sum_{i=1}^m \alpha_i P(\theta_{v_i}),
$$

where $\sum_{i=1}^m \alpha_i = 1$, and $\alpha_i \geq 0$ are the convex coordinates.

The source of nonlinearity in (17) is the nonlinear convective term which is obvious in (6). By defining the scheduling parameter vector as

$$
\theta(k) = x_c(k),
$$

one can obtain state matrices $A(\theta)$, $B(\theta)$ and $C$ as given in (25) such that the model (19) is affine in $\theta$. Note that these matrices are needed to be calculated only for controller synthesis and not for on-line implementation. The number of states is $M = 3$ and number of vertices is $m = 8$, which leads to low on-line computation and makes it possible to use standard synthesis tools.

### 4.2 LPV Control Synthesis

A discrete-time output-feedback LPV controller with a fixed Lyapunov function is designed for the low-order quasi-LPV model of Burgers’ equation (19) using the $H_\infty$ loop-shaping approach proposed by Apkarian et al. (1995). This method has been proven to be an effective and practical tool for LPV synthesis due to the simplicity of the synthesis and implementation. The block diagram of the LPV controller is shown in Fig. 3. Note that $\theta$ is not available and has to be estimated. Thus, it is replaced by $\hat{\theta}$. The nonlinear Burgers’ system (7) is controlled by an LPV controller $\Omega(\hat{\theta})$, where the output $y(k)$ is corrupted by measurement noise $n(k)$. The design objective considered here is to stabilize (7) in the whole operating range with a high tracking capability and disturbance and measurement noise rejection while taking into consideration a limited bandwidth and amplitude for control signal. The generalized plant for mixed sensitivity design is shown in Fig. 4, where $W_S(z)$ and $W_K(z)$ are the weighting filters to shape sensitivity $S(\theta)$ and control sensitivity $\Omega(\theta)S(\theta)$ functions, respectively.

![Fig. 3. Block diagram of the LPV gain-scheduled controller](image)

For designs based on polytopic LPV models, the model must not have a parameter dependent input matrix $B(\theta)$; but this is not the case in (19). To solve this problem, the plant is augmented by a low-pass filter $W_P(z)$ with a sufficiently large bandwidth (Apkarian et al. 1995).

The mixed-sensitivity criterion to be minimized for controller design is the induced $L_2$ gain of the closed-loop operator between $r$ and $[z_N \ -z]$. A $9$th order discrete-time output-feedback LPV controller $\Omega(\theta)$ is designed with the state space realization

$$
x^{c}(k + 1) = A(\theta(k))x^{c}(k) + B(\theta(k))e(k)
$$

$$
u(k) = C(\theta(k))x^{c}(k) + D(\theta(k))e(k),
$$

(26)
Fig. 4. Generalized plant for LPV controller synthesis where the parameter dependent matrices are determined by the convex coordinates and controller vertices \( A_{i}^{c} \), \( B_{i}^{c} \), \( C_{i}^{c} \), and \( D_{i}^{c} \) with \( i = 1, \ldots, m \). These are computed using a discrete-time version of MATLAB robust control toolbox command \textit{hinfgs}. The frequency response of the sensitivity and the control sensitivity functions of the closed-loop system at the vertices of the LPV model are plotted in Fig. 5, which shows that all frequency domain objectives are achieved.

Fig. 5. Sensitivity (upper) and control sensitivity (lower) plots (solid) and inverse of weighting filters (dashed) since the LPV controller (26) needs access to the scheduling parameters \( \theta \) during implementation and only \( w_{N_1} \) and \( w_{N_2} \) are assumed to be measurable, an observer is designed in the next section to estimate the scheduling parameters. It should be noted that estimation of scheduling parameters is not taken into account in the controller synthesis procedure and it is assumed that knowledge about scheduling parameters is accurate. However, when the designed controller was implemented, it turned out that using estimated scheduling parameters does not affect the closed-loop stability and performance significantly.

5. FUNCTIONAL OBSERVER DESIGN

Functional observers (Luenberger [1971]) can be used to estimate linear functionals of the system states. Recall that the states of the reduced model \( x_r \) are linear combinations or functionals of the original states \( x \) or their approximation \( \hat{x} \). Thus, a functional observer design for the original nonlinear system (7) is equivalent to the design of an observer for the reduced nonlinear model (17). Such an observer has a much lower dynamic order than an observer which estimates the states of the original system.

The recent observer design method for nonlinear discrete-time systems developed by Abbaszadeh and Marquez [2007] is used in this paper. This method is applicable to systems with a nonlinearity \( F_r(\hat{x}, u) \) which is locally Lipschitz with respect to \( \hat{x} \) in a region \( D \), i.e. \( \forall \hat{x}_2(k) \in D : \| F_r(\hat{x}_1, u^*) - F_r(\hat{x}_2, u^*) \| \leq \gamma_d \| \hat{x}_1 - \hat{x}_2 \| \), (27) where \( \| . \| \) is the induced 2-norm, \( u^* \) is any admissible control signal and \( \gamma_d \) is the Lipschitz constant. If the nonlinearity satisfies the Lipschitz continuity condition globally, then the designed observer will be stable globally.

The proposed observer takes the form
\[
\hat{x}_r(k+1) = A_r \hat{x}_r(k) + F_r(\Phi_r \hat{x}_r(k), u(k)) + L(y(k) - C_r \hat{x}_r(k)),
\]
where \( \hat{x}_r \in \mathbb{R}^M \) is an estimate of \( x_r \) and the observer gain \( L \) is designed such that the following estimation error dynamics is asymptotically stable
\[
e_r(k+1) = A_r e_r(k+1) - \hat{x}_r(k+1) = (A_r - LC_r)e_r(k) + F_r(\Phi_r e_r(k), u(k)) - F_r(\Phi_r \hat{x}_r(k), u(k)).
\]

Aabbazadeh and Marquez [2007] proposed a sufficient LMI condition to obtain a \( L \) stabilizing (29), such that \( \gamma_d \) is maximized. Since the quadratic nonlinearities in (9) are Lipschitz, this technique is employed here and the designed observer is used to estimate \( x_r \) or \( \theta \) and schedule the LPV controller. The order of observer is \( M = 3 \) and only one high-dimensional on-line matrix multiplication is needed in (28) to calculate \( F_r \) using (18).

Implementation of the designed LPV controller and the reduced-order observer is presented in the next section.

6. SIMULATION RESULTS

The closed-loop system shown in Fig. 3 is simulated in MATLAB/SIMULINK using the full-order model (7) and the designed LPV controller and functional observer. The simulation sampling frequency is set to 200 Hz. The control outputs \( y_1 \) and \( y_2 \) are corrupted using the band-limited white noise block of SIMULINK with power \( P_n = 10^{-6} \). Control outputs in flow control problems are typically close to the boundaries, thus \( N_1 = 6 \) and \( N_2 = 44 \) are chosen. The reference trajectory \( r(t) \) to be tracked is chosen as a sequence of smoothed step functions.

Figure 6 illustrates the outputs together with reference trajectories. A fast response without any overshoot is achieved. The reference trajectory for \( y_2 \) has been delayed in order to check the cross-coupling rejection. The injected measurement noise is suppressed considerably. The results illustrate that loop-shaping objectives have been met.

To make a comparison, the reduced model (17) is linearized around an equilibrium \( (x_r, u) = (0, 0) \) and an LQG controller similar to the report by Lawrence et al. [2005] is designed and tuned for the best disturbance and measurement noise rejection. In the mentioned report, a noise-free distributed measurement was assumed. Another simulation is done to reject a non-zero initial condition with both controllers and the results are illustrated in Fig. 7. It is obvious that the LPV controller suppresses the measurement noise better, and rejects the disturbance faster than the LQG controller. Faster disturbance rejection is possible by LQG controller, but measurement noise is amplified significantly in that case.
Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Fig. 6. Command trajectory (dotted) and control outputs (dashed) for LPV controller

Fig. 7. Rejection of non-zero initial condition disturbance by LPV (dashed) and LQG (solid) controllers

7. CONCLUSION

In this paper, a framework for LPV modelling and control of a nonlinear PDE is proposed. The one-dimensional viscous Burgers’ equation was discretized using a finite difference scheme and the boundary conditions were taken as control inputs. A discrete-time nonlinear state space model with a high-order was obtained and after generating an ensemble of typical trajectories, proper orthogonal decomposition together with Galerkin projection was used for model order reduction.

The accuracy of the reduced model was verified and a discrete-time polytopic quasi-LPV model that is affine in scheduling parameters was derived, where the scheduling parameters are the same as the reduced states. Since the derived LPV model has a small number of scheduling parameters and vertices, an output-feedback LPV controller was designed which has a low on-line computation cost. A low-order functional observer was designed to estimate the scheduling parameters required for LPV controller. Simulation results demonstrate the high tracking performance and disturbance and measurement noise rejection capabilities of the designed LPV controller when compared with an LQG controller based on a linearized model.

The number of required measurements and actuations in the proposed method is realistic.

The work reported in this paper is a first step towards the application of the LPV framework to PDEs and flow control problems. Although the control loop and scheduling parameter estimation loop are proven to be stable, the stability of the combined loop has not been proven yet. As an alternative, extracting the system states from controller states has been also done and will be reported separately.

REFERENCES


