Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

LPV Design of Charge Control for an SI Engine Based on LFT Neural State-Space Models

H. Abbas * **H. Werner** *

* *Institute of Control, Hamburg University of Technology, 21073, Hamburg, Germany (e-mail: hossam.abbas, h.werner @tu-harburg.de).*

Abstract: This paper is one of two joint papers, each presenting and utilizing a different representation of a feedforward neural network for controller design. Here a neural state-space model is transformed into a linear fractional transformation (LFT) representation to obtain a discrete-time quasi-linear parameter-varying (LPV) model of a nonlinear plant, whereas in the joint paper (Abbas and Werner [2008]) a method is proposed to transform the neural state-space into a discrete-time polytopic quasi-LPV model. As a practical application, air charge control of a Spark-Ignition (SI) engine is used in both papers as example to illustrate two different synthesis methods for fixed structure low-order discrete-time LPV controllers.
In this paper, a method that combines modelling using a multilayer perceptron network and controller synthesis using linear matrix inequalities (LMIs) and evolutionary search is proposed. In the first step a neural state-space model is transformed into a linear fractional transformation (LFT) representation to obtain a discrete-time quasi-LPV model of a nonlinear plant from input-output data only. Then a hybrid approach using LMI solvers and genetic algorithm, which is based on the concept of quadratic separators, is used to synthesize a discrete-time LPV controller.

1. INTRODUCTION

The intake manifold of a SI Engine for air charge control has a highly nonlinear nature, and a model that captures the nonlinear dynamics of the plant is required to design a controller. Linear parameter-varying (LPV) control is an efficient way of extending well-known linear controller design techniques to nonlinear plants. In (Kwiatkowski et al. [2006]), a nonlinear model of the intake manifold was presented, based on which an LPV model and the design of an LPV controller was proposed. Constructing an LPV model of a nonlinear plant is however not trivial, and generating it from a physical nonlinear plant model has two drawbacks:

- Obtaining a suitable physical nonlinear model requires considerable effort and insight into the plant dynamics. Moreover, transforming such a model into an LPV model often necessitates approximations, e.g. of nonlinear characteristics by polynomials.
- The physical model is typically continuous-time, and approximating a controller designed in continuous time e.g. using Tustin approximation may lead to performance deterioration when the sampling rate is limited.

An alternative approach that avoids both problems is to identify a LPV model directly from experimental input-output data. Neural networks such as multilayer perceptrons (MLPs) have been shown to be able to model nonlinear systems to an arbitrary degree of accuracy and have been successfully integrated in several control applications, see e.g. (v. d. Boom et al. [2003]) and (Norgaard et al. [2000]).

A method for extracting an LPV model from a multilayer perceptron network and controller design based on this model, was proposed in (Bendtsen and Trangbæk [2002]), where a neural-state space model is transformed into linear fractional transformation (LFT) form in a nonconservative way. The key to that approach is the observation that the LFT formulation allows for a quasi-LPV description of a nonlinear system, in which the scheduling parameters depend on the measured system output. In this case, the extracted nonlinearities define the parameter variation, and it is hence possible to exploit this for controller synthesis. However, the authors of (Bendtsen and Trangbæk [2002]) mention some drawbacks of their approach:

- The proposed controller synthesis is done in continuous time and requires to convert the discrete-time model into a continuous-time one.
- Bounds on the rate of change of the scheduling parameters are not taken in consideration, which leads to a conservative design.
- Only full-order controllers can be designed, which is unattractive for practical purposes.
- Assessment of the quality of the LPV model was only by neural network validation and not in terms of achievable control performance.

All these issues are addressed in this paper.

A method to convert a neural state-space model into a discrete-time polytopic quasi-LPV model in a nonconservative way is proposed in the joint paper to this paper (Abbas and Werner [2008]), that allows to apply linear techniques such as a polytopic LPV controller synthesis (Apkarian et al. [1996]) to a nonlinear plant.

The problem of designing controllers for discrete or continuous time LPV systems can be solved using LMI-based tools discussed in, e.g. (Boyd et al. [1994]) . Most of these approaches rely on a fixed Lyapunov function for the whole range of operation, which leads to conservatism of the design. This conservatism can be reduced by the use of parameter dependent Lyapunov functions when bounds on the rate of parameter change are known, (Apkairan and Adams [2000]) . This approach is however computationally unattractive: the problem is non-convex and involves extensive gridding. A procedure for designing a discrete-time LPV controller that exploits known bounds on the rate of parameter variation was proposed in (Chughtai and Werner [2007]). In this approach the analysis technique of quadratic separators (Iwasaki and Shibata [2001]) has been used, and the synthesis problem is solved using LMI solvers and genetic algorithms.

In this paper an extended version of the idea in (Bendtsen and Trangbæk [2002]) is used to obtain an LFT representation for the neural state-space model. The method proposed in (Chughtai and Werner [2007]) is then used to design a discrete time low-order LPV controller for the model. The modeling and the controller synthesis steps are combined in one algorithm in order to assess different models in terms of achievable control performance, to detect the best model and find a corresponding controller. To illustrate this approach it is applied to the charge control of a SI engine.

The paper is organized as follows. Section 2 presents a method for transforming a neural state-space model into a quasi-LPV model in the form of an LFT representation. Section 3 gives an outline of the discrete-time LPV controller synthesis technique as well as the proposed design algorithm. To illustrate the algorithm, it is applied to charge control of a SI engine in Section 4; this section includes a brief description of the nonlinear system and how it is identified using a neural network. Finally, conclusions are drawn in Section 5.

2. DERIVATION OF A QUASI-LPV MODEL FROM A NEURAL NETWORK MODEL

Consider a discrete-time nonlinear model

$$x_{k+1} = g(x_k, u_k).$$

The problem considered in this section is to represent this model in the form of a discrete-time quasi-LPV model in LFT representation

$$\begin{aligned} x_{k+1} &= Ax_k + B_\Delta w_k^\Delta + Bu_k \\ z_k^\Delta &= C_\Delta x_x + D_\Delta u_k \\ y_k &= Cx_k \\ w_k^\Delta &= \Delta_k z_k^\Delta \end{aligned}$$

The idea in (Bendtsen and Trangbæk [2002]) will be extended to construct the above quasi-LPV model from a given multilayer perceptron (MLP) feedforward neural network with a single hidden layer, l neurons and hyperbolic tangent activation function. This can be done by transforming a neural state-space model into a LFT representation. Assume that the nonlinear system dynamics are described in the following neural state-space form with zero bias in the output layer

$$\tilde{x}_{k+1} = W_o \tanh(W_x \tilde{x}_k + W_u \tilde{u}_k + \tilde{W}_b) \quad (1a)$$

$$\tilde{y}_k = C\tilde{x}_k \quad (1b)$$

where $\tilde{x}_k = [\tilde{y}_k^T \quad \tilde{y}_{k-1}^T \quad ... \quad \tilde{y}_{k-ny}^T]^T \in \mathbb{R}^n$ is the state vector, $\tilde{u}_k \in \mathbb{R}^m$ is a control signal, $\tilde{y}_k \in \mathbb{R}^p$ is the output vector of the system, $W_o \in \mathbb{R}^{n\times l}$ and $W_x \in \mathbb{R}^{l\times n}$, $W_u \in \mathbb{R}^{l\times m}$ contain the output and hidden layer weights, respectively. $\tilde{W}_b \in \mathbb{R}^l$ contains a set of biases in the hidden layer. During the completion of the neural network model training phase, it should be trained long enough on a sufficiently rich training signal, and choices have to be made for the indices l and n_y, n_u, the past output and input, respectively, such that a good estimation of the parameter matrices W_o, W_x, W_u and vector $\tilde{W}_b$ is obtained (Norgaard et al. [2000]).

In order to remove the bias $\tilde{W}_b$ from (1a), it is assumed that there exists an equilibrium point, $(\tilde{x}_k, \tilde{u}_k) = (\tilde{x}_k^o, \tilde{u}_k^o)$, such that

$$0 = W_o \tanh(W_x \tilde{x}_k^o + W_u \tilde{u}_k^o + \tilde{W}_b)$$

and this equilibrium can be determined as follows:

$$\begin{bmatrix} \tilde{x}_k^o \\ \tilde{u}_k^o \end{bmatrix} = -W^+ \tilde{W}_b \quad (2)$$

where $W = [W_x \quad W_u]$ is assumed to have full column rank and W^+ is a right inverse. Then the network coordinates can be changed such that the new coordinates are $x_k = \tilde{x}_k - \tilde{x}_k^o$, $u_k = \tilde{u}_k - \tilde{u}_k^o$. As a result, (1a) can be written as

$$x_{k+1} = W_o \tanh(W_x x_k + W_u u_k) \quad (3)$$

Let the input argument to the neuron functions be given as

$$\xi_k = W_x x_k + W_u u_k \quad (4)$$

The effective range of ξ can be calculated as

$$\xi^{j,max} = \sup_{0\geq k\geq T_f} |W_x^j x_k + W_u^j u_k|$$

for $1 \leq j \leq l$ where $k \in [0 \quad T_f]$ is the time interval in which the training data have been acquired and W_x^j, W_u^j denote the jth rows in the hidden layer weight matrices. We have the following bounds on the active input range of the jth neuron

$$\xi_k^j = W_x^j x_k + W_u^j u_k \quad \in [-\xi^{j,max} \quad \xi^{j,max}]$$

Hence the neuron function response to the active input range belongs to the sector $[k^{j,min} \quad k^{j,max}]$ where

$$k^{j,min} = \inf_{\xi_k^j \in [-\xi^{j,max} \ \xi^{j,max}]\setminus\{0\}} \left\{ \frac{\tanh(\xi^j)}{\xi^j} \right\} \quad (5a)$$

$$k^{j,max} = \sup_{\xi_k^j \in [-\xi^{j,max} \ \xi^{j,max}]\setminus\{0\}} \left\{ \frac{\tanh(\xi^j)}{\xi^j} \right\} \quad (5b)$$

The sector bounds must be found for each neuron function individually. As shown in Fig. 1, it is immediately concluded that

$$k^{j,min} = \tanh(\xi^{j,max})/\xi^{j,max}, \quad k^{j,max} = 1$$

Once the sector bounds are found, a nonlinear function $\omega(.) : R^{n+m} \to R^n$ can be defined as

$$\omega(\xi_k) = \tanh(\xi_k) - \frac{1}{2}(K^{min} + K^{max})\xi_k \quad (6)$$

where $K^{min} = diag\{k^{j,min}\}$ and $K^{max} = diag\{k^{j,max}\}$, $1 \leq j \leq l$. It is observed that $\omega(.)$ belongs to the sector

$(-1/2(K^{max}-K^{min}), 1/2(K^{max}-K^{min}))$. Now (3) can be written as

$$\begin{aligned} x_{k+1} &= W_o(\omega(\xi_k) + \frac{1}{2}(K^{max}+K^{min})\xi_k) \\ &= Ax_k + Bu_k + B_1\Omega(\xi_k) \end{aligned} \tag{7}$$

in which A, B, B_1 and Ω are given by

$$\begin{aligned} A &= \frac{1}{2}W_o(K^{min}+K^{max})W_x \\ B &= \frac{1}{2}W_o(K^{min}+K^{max})W_u \\ B_1 &= \frac{1}{2}W_o(K^{max}-K^{min}) \\ \Omega(\xi_k) &= 2(K^{max}-K^{min})^{-1}\omega(\xi_k) \end{aligned}$$

Note that the diagonal scaling by $1/2(K^{max}-K^{min})$ is included in order to make the diagonal static nonlinearity Ω belong to the sector $[-1\ 1]$.

Note also that the nonlinear mapping $\Omega(.)$ is diagonal. The gains in $\Omega(\xi_k)$ are considered here to be time-varying residual gains Bendtsen and Trangbæk [2002], i.e.

$$\Delta_k\xi_k = \Omega(\xi_k) \tag{8}$$

The rate of change of the gains of the residual function can be taken into account in the controller synthesis procedure described below to reduce the conservatism.

Using (2), (4), (7) and (8), a discrete-time quasi-LPV model can be determined from the neural state space model. The method here reduces the conservatism by only considering the neuron functions in the range where information is supplied via the training set. Since the training set represents the only information available about the process, there is no straightforward way of introducing robustness to unmodelled uncertainties. This, of course, only makes it more important that the training set is indeed sufficiently rich to provide information about the behavior in the entire operating range of interest and the frequency range at which the controller will operate.

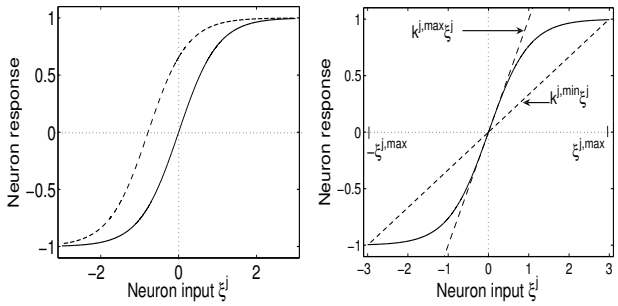

Fig. 1. Shifting the original hyperbolic tangent neuron function (dashed) with bias $\tilde{W}_b$ to the origin (solid) and defining its sector bounds.

3. DISCRETE-TIME CONTROLLER DESIGN

This section describes LPV controller synthesis based on the model obtained in the previous section. The idea of quadratic separators has been used - employing a technique proposed in (Chughtai and Werner [2007]) for discrete-time LPV systems - the existence of which is equivalent to the existence of a parameter-dependent Lyapunov function (Iwasaki and Shibata [2001]). This allows to take bounds on the rate of parameter variation into account. The method in (Chughtai and Werner [2007]) uses a hybrid evolutionary-algebraic approach for solving the non-convex problems of low-order and fixed structure controller design that was proposed in (Farag and Werner [2004]). The idea is to decompose the non-convex problem into a convex subproblem with a large number of variables, solved by Riccati or LMI solvers, and a non-convex subproblem which is solved by evolutionary search methods. The evolutionary search can be carried out for a discrete time LPV controller that minimizes the induced $\mathcal{L}_2$ norm of the closed-loop system from w^p to z^p as illustrated in Fig. 2, while LMI solvers are used to find a quadratic separator.

The generalized plant P, as shown in Fig. 2, includes the model of the plant and the weighting filters for performance. The generalized plant and the controller depend on the parameter matrix Δ in an LFT manner as given in (4), (7), (8) for the plant and for the controller $K(\Delta)$ as follows:

$$K(\Delta) = \begin{bmatrix} A^c & B^c \\ C^c & D^c \end{bmatrix} + \begin{bmatrix} B^c_\Delta \\ D^c_{1,\Delta} \end{bmatrix} \Delta \begin{bmatrix} C^c_\Delta & D^c_{\Delta,1} \end{bmatrix} \tag{9}$$

The closed-loop system can be described by the following LFT structure

$$x_{k+1} = Ax_k + B_\Delta w^\Delta_k + B_p w^p_k \tag{10a}$$
$$z^\Delta_k = C_\Delta x_k + D_\Delta w^\Delta_k \tag{10b}$$
$$z^p_k = C_p x_k + D_p w^p_k \tag{10c}$$
$$w^\Delta_k = \Delta_k z^\Delta_k \tag{10d}$$

where $w^\Delta_k,\ z^\Delta_k \in \mathbb{R}^l$, $w^p_k \in \mathbb{R}^d$, $z^p_k \in \mathbb{R}^v$, and $\Delta_k = diag(\Delta_k, \Delta_k)$ where each $\Delta_k \in \mathbb{R}^{l\times l}$. It should be noted here that the dimension of the uncertainty block Δ_k is equal to the number of neurons in the hidden layer of the neural network model. The condition on the induced $\mathcal{L}_2$-norm, derived in (Chughtai and Werner [2007]), is repeated here for completeness. Define a set of matrices and scalings according to

$$\mathcal{A} = \begin{bmatrix} A & B_\Delta \\ 0 & 0 \end{bmatrix}, \qquad \mathcal{B} = \begin{bmatrix} B_p & 0 & 0 \\ 0 & I & 0 \end{bmatrix},$$
$$\mathcal{C} = \begin{bmatrix} C_\Delta & D_\Delta \\ C_\Delta A & C_\Delta B_\Delta \\ C_\Delta A & C_\Delta B_\Delta \\ 0 & I \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 0 & 0 & 0 \\ C_\Delta B_p & D_\Delta & 0 \\ C_\Delta B_p & D_\Delta & 0 \\ 0 & 0 & 0 \\ 0 & I & -I \\ 0 & 0 & I \end{bmatrix}, \tag{11}$$
$$\mathcal{E} = \begin{bmatrix} C_p & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathcal{F} = \begin{bmatrix} D_p & 0 & 0 \\ I & 0 & 0 \end{bmatrix}$$

$$\mathcal{S}_D := \{D : D\nabla = \nabla D, D = D^T > 0\} \tag{12a}$$
$$\mathcal{S}_G := \{G : G\nabla = \nabla G, G + G^T = 0\} \tag{12b}$$

where

$$\nabla := diag(\Delta_k, \Delta_k, \Delta^\delta_k), \tag{13a}$$
$$\Delta_k = diag\{q_{1,k}I_{r_1}, ..., q_{l,k}I_{r_l}\}, \tag{13b}$$
$$\Delta^\delta_k = diag\{\delta q_{1,k}I_{r_1}, ..., \delta q_{l,k}I_{r_l}\}, \tag{13c}$$
$$= \Delta_{k+1} - \Delta_k \tag{13d}$$

and all varying parameters $q_{i,k}$ and their changes $\delta q_{i,k}$ are bounded such that $|q_{i,k}| < \phi_i$ and $|\delta q_{i,k}| < \rho_i$, $\forall\ i = 1, ..., l$, where the index r_i represents the multiplicity of the i^{th} uncertainty. The following Theorem gives a performance

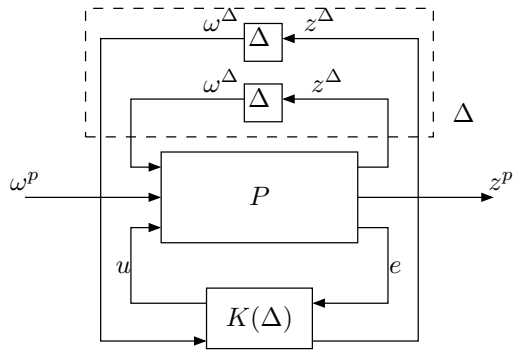

Fig. 2. Closed-loop system with model uncertainty Δ

measure for the closed loop (Chughtai and Werner [2007]).

Theorem 1. The LPV system described by (10a) - (10d) is stable and has worst case induced $\mathcal{L}_2$-performance less than γ if there exist real symmetric matrices $P > 0$, S and R such that with $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$ as defined in (11) the following conditions hold

$$\begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \\ I & 0 \\ \mathcal{E} & \mathcal{F} \end{bmatrix}^T \begin{bmatrix} P & 0 & 0 & 0 \\ 0 & \Theta & 0 & 0 \\ 0 & 0 & -P & 0 \\ 0 & 0 & 0 & \Gamma_\gamma \end{bmatrix} \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \\ I & 0 \\ \mathcal{E} & \mathcal{F} \end{bmatrix} < 0 \tag{14a}$$

$$\begin{bmatrix} I \\ \nabla \end{bmatrix}^T \Theta \begin{bmatrix} I \\ \nabla \end{bmatrix} > 0, \quad \Gamma_\gamma = \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \tag{14b}$$

where

$$\Theta := \left\{ \begin{bmatrix} \Upsilon R \Upsilon & S \\ S^T & -R \end{bmatrix} : R \in \mathcal{S}_D, S \in \mathcal{S}_G \right\} \tag{15a}$$

$$\Upsilon := diag(\Upsilon_\phi, \Upsilon_\phi, \Upsilon_\rho) \tag{15b}$$

$$\Upsilon_\phi := diag(\phi_1 I_{r_1}, ..., \phi_l I_{r_l}) \tag{15c}$$

$$\Upsilon_\rho := diag(\rho_1 I_{r_1}, ..., \rho_l I_{r_l}) \tag{15d}$$

Proof. see (Chughtai and Werner [2007])

The conditions in (14) and (15) - for a given controller are linear in the variables γ, P, S, and R. To determine the optimal performance measure, one needs to find the solution of the problem

$$\min_{P,S,R} \gamma \quad \text{subject to equations (14) and (15)} \tag{16}$$

which can be computed with the help of LMI solvers. Using the above analysis result, the nonconvex controller synthesis problem can be solved using a hybrid evolutionary-algebraic approach proposed in (Chughtai and Werner [2007]), which is briefly reviewed here. Let K_i denote individual controllers, which can have a simple structure and low order, and A^i the corresponding closed-loop system matrix in (10a). The synthesis procedure can then be summarized as follows.

- **Generate** an initial random population of controllers $\{K_1, ..., K_\mu\}$.
- **Evaluate** the objective function

$$f(K_i) = \begin{cases} \hat{\gamma} & \text{if } A^i \text{ is stable,} \\ \kappa(A^i) + \beta_u & \text{if } A^i \text{ is unstable,} \\ \beta_s & \text{if } A^i \text{ stable, (16) infeasible} \end{cases}$$

  where $\hat{\gamma}$ is the solution to the minimization problem (16). The term $\kappa(A^i)$ denotes the maximum real part of the eigenvalues of A^i, and $\beta_u \gg \beta_s$ represents a penalty for destabilizing controllers and infeasible inequalities, respectively.
- **Evolve** the current population, i.e. use ranking to evaluate the fitness and apply evolutionary operators (mutation and crossover).
- **Repeat** the steps *Evaluate* and *Evolve* until a stopping criterion is met.

3.1 Modelling for Controller Synthesis

The choice of the structure (n_y, n_u, l) of the neural state-space model is not trivial. Moreover, validation of the neural network model alone is not enough to say that it is the best representation of the plant. It is proposed here to combine the controller synthesis step with the modeling step in order to find the best model in the sense of achievable control performance. This can be done by training a set of neural networks in state-space form with different structures, possibly using different training signals. The modelling and synthesis procedure described above is then applied to all models in the set, and the achieved control performance is compared.

4. APPLICATION TO CHARGE CONTROL

In the following, the design procedure described in the previous section will be applied to a nonlinear model of the intake manifold of a SI engine (Kwiatkowski et al. [2006]) for air charge control. The nonlinear dynamics of the system can be represented as a discrete time quasi-LPV model based on the neural network model from the input-output data only. Since the engine is controlled by a digital controller and the model is in discrete-time form, a direct discrete-time design of an LPV controller is possible.

4.1 The Intake Manifold of a SI Engine

The intake manifold is not an isolated system but it is part of the overall system of the vehicle, Fig. 3. The nonlinear engine block generates the torque T_e from the normalized air charge m_{nac} and the motor speed N, and the vehicle model generates the motor speed from the torque and some fixed environmental conditions. The opening of the throttle valve in the intake manifold α_{lim} is used to control the amount of the normalized air charge. It should be noted here that the speed of the engine influences the internal dynamics of the intake manifold and the engine.

The vehicle model as shown in Fig. 3 has integral behavior; for this reason the loop is closed between the engine speed and α_{lim} through a proportional gain K_p as shown in Fig. 3 in order to stabilize the engine speed during the input-output data collection for system identification.

4.2 Nonlinear System Identification with Neural Networks

Generating an input signal to excite the different dynamics of the nonlinear system is a crucial step in system identification. The required operating ranges, the sampling frequency and the bandwidth of the system are important information to design a rich training signal. As shown in Fig. 3, the intake manifold system has two inputs α_{lim}

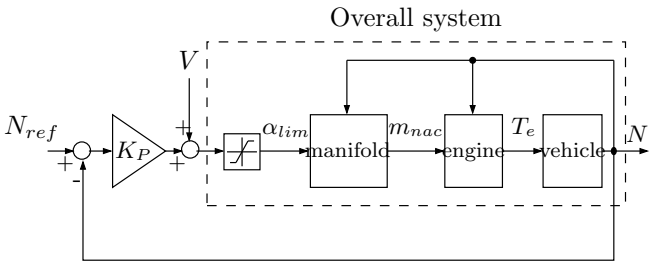

Fig. 3. Overall system in the closed-loop identification

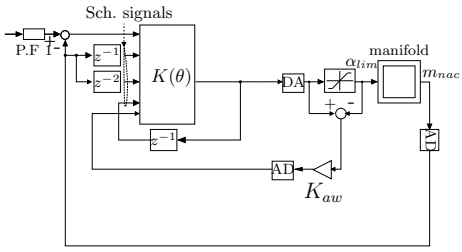

Fig. 4. Block diagram of the closed loop system

and N, and a single output m_{nac}. The aim here is to construct a single-input single-output black box model based on the neural network that is robust against the variation of the engine speed. This can be achieved if the training signal excites the input-output operating ranges at different levels of the engine speed. For this purpose, a random signal, N_{ref}, in the low frequency range is designed to cover the whole range of the engine speed and another pseudo-random multilevel sequence V in the high frequency range (Norgaard et al. [2000]), up to the Nyquist frequency, is designed to excite the input-output dynamics of the system (from α_{lim} to m_{nac}). The way the signals enter the loop is shown in Fig. 3. The input output data are collected and divided into training and validation sets. The required operating ranges for the different variables to design the input signal are as follows

$$m_{nac} \in [10\ 80]\%,\ \alpha_{lim} \in [0\ 100]\%,\ N \in [760\ 6250]rpm$$

Here an NARX structure is used as the neural network model structure with different values of n_y, n_u, l. Several feedforward networks with different structure were separately trained using the Levenberg-Marquardt optimization algorithm (Norgaard [2000]), and it was validated that each model had learned the behavior of the intake manifold system and was robust against the variations of the engine speed through k-step ahead prediction of a separately generated validation set. In this way a set of neural state-space models was determined.

4.3 Controller Implementation and Simulation Results

The first step in the proposed design approach is to convert the neural network models into LFT form using the method in Section 2 to obtain discrete-time quasi-LPV models. Next, controllers are designed for these models. The design objectives considered here are

- Rise time $t_r < 0.3s$
- Zero steady state error
- Small overshoot
- Constraint on actuator usage

To meet these objectives, two weighting filters were designed: W_S for sensitivity and W_{KS} for control sensitivity. These were chosen as:

$$W_S = \frac{0.4z^{-1}}{1 - 0.7z^{-1}}, \quad W_{KS} = 0.01 \tag{17}$$

For implementation in the electronic control unit of the car, a low-order controller is required. Thus, a LPV-PID controller was designed using the approach presented in Section 3. The controller has the structure

$$K(\Delta) = K_P(\Delta) + K_I(\Delta)\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}} + \frac{K_D(\Delta)}{1 + \frac{\alpha T}{2}\frac{1+z^{-1}}{1-z^{-1}}} \tag{18}$$

where T is the sampling time and α ia a suitable constant to make the PID controller realizable. The controller K_Δ is converted into LFT form (9). Next the generalized plant, as shown in Fig. 2, is constructed with the weighting filters W_S and W_{KS} included in the performance channels. The synthesis procedure is then applied to find and tune the coefficients of $K(\Delta)$. Thus a LPV-PID controller has been designed for each model (provided that (16) is feasible). The controllers are then tested on the nonlinear model of the intake manifold. To avoid windup effects after actuator saturation, an anti-wind up is applied to the integral part of the LPV-PID controller as proposed in (Kwiatkowski et al. [2006]). By comparing the achieved performance (in terms of the induced $\mathcal{L}_2$ gain), the best controller and the corresponding model are determined. Here the control algorithm was applied to twenty neural state-space models with different structure, and the best model has $l = 3$, $nu = 1$ and $ny = 2$ and the smallest induced $\mathcal{L}_2$ norm achieved is $\gamma \approx 2$. The discrete-time controller is connected to the continuous-time nonlinear model through digital-to-analog and analog-to-digital converters as shown in Fig. 4, together with the anti-windup and a pre-filter, which is used to shape the step response. The closed-loop tracking performance is illustrated in Fig. 5 and Fig. 6. The controller follows the reference trajectory in a satisfactory manner, with a rise time between $t_r = 0.1s$ and $t_r = 0.25s$, without overshoot, and the maximum undershoot is less than 2.3%. The simulation results also show that the integral gain K_I is adjusted more heavily than the other gains.

It is worth to mention that the achieved performance here is nearly the same as that achieved by the controllers which were designed based on a physical model given in (Kwiatkowski et al. [2006]). On the other hand, a slight improvement in the achieved performance here compared to the one achieved with the method in (Abbas and Werner [2008]) and designed for the same system is because in (Abbas and Werner [2008]) the controller design was based on a fixed Lyapunov function for the whole operating region, which results in more conservatism in the controller design. The approach based on a parameter-dependent Lyapunov function presented in this paper reduces that conservatism and hence improves the performance.

Remark 1: The synthesis method used here for the discrete-time LPV controller assumes that there is no feedforward connection between the input performance channel w_k^p and the input channel to the uncertainty block z^Δ in the closed loop system, see (10b), i.e. the D matrix from w_k^p to z^{Δ_k} should be zero. Unfortunately, in the example here, when the generalized plant, Fig. 2 was constructed with the weighting filters included in the

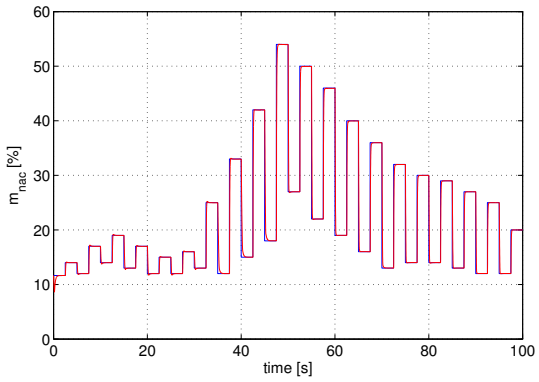

Fig. 5. Tracking of a reference trajectory for m_{nac}

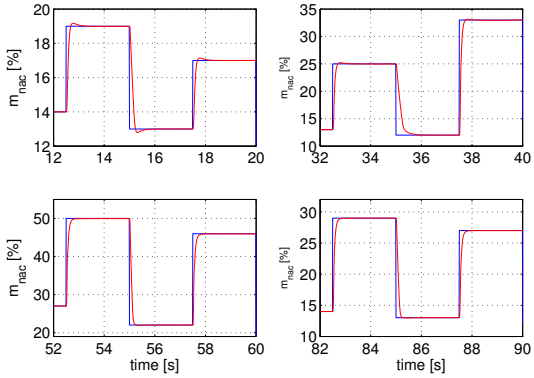

Fig. 6. Zoomed plots of different parts of the trajectory

performance channels, the D matrix turned out to be non-zero. This problem was dealt with by simply adding a one-step delay z^{-1} to the input performance channel w_k^p.

Remark 2: It should be pointed out that the dimension of the Δ block (8) depends on the number of neurons in the hidden layer of the neural state-space model, and the efficiency of the synthesis approach is improved as the dimension of this block is reduced, because the number of the design variables is decreased. This should be taken in consideration when the number of the hidden neurons is chosen during the modelling stage.

Remark 3: Integral action explicitly appears in the PID controller, which makes the use of anti-wind up more straightforward to deal with the saturation problem in comparison with a full order controller, which has no explicit integrator.

Remark 4: Figure 4 shows that the scheduling signals of the LPV-PID controller depend only on the input and the output signals of the nonlinear system, thus there is no need to measure any additional signals. This is a useful feature of the proposed black box modelling method.

5. CONCLUSION

This paper presents the design of a discrete-time low-order LPV controller for the air charge control of a SI engine, which is modelled as neural state space model. The design procedure integrates modelling and controller synthesis. A number of discrete-time quasi-LPV models in LFT representation are derived from neural state-space models in the modelling step. A hybrid evolutionary-algebraic synthesis procedure is used to design low-order discrete-time LPV controllers for these models that guarantee stability and performance for a wide range of operation. The inherent conservatism in the synthesis step is reduced by using the concept of quadratic separator, which takes into account bounds on the rate of change of the parameters. The proposed method resolves all issues that were left open in (Bendtsen and Trangbæk [2002]). Different neural state-space models are assessed according to the performance of the corresponding controller on the nonlinear system. The proposed method was successfully applied to the charge control problem and resulted in a discrete-time LPV-PID controller, that achieved the same performance as a previously reported LPV-PID design for this problem that was based on a nonlinear physical model.

REFERENCES

H. Abbas and H. Werner. Polytopic quasi-LPV model based on neural state-space models and application to air charge control of a SI engine. *To appear in Proc. IFAC World Congress 2008.*

P. Apkairan and R. Adams. *Chapter 11: Advances in Linear matrix Inequality Methods in Control.* Advances in Design and Control. SIAM, Philadelphia, PA, USA, 2000.

P. Apkarian, G. Becker, P. Gahinet, and H. Kajiwara. LMI techniques in control engineering from theory to practice. In *Workshop notes CDC*, Kobe, Japan, 1996. IEEE.

J. Bendtsen and K. Trangbæk. Roubst quasi-LPV control based on neural state space models. *IEEE Transactions on Neural Networks*, 13(2):355–368, 2002.

S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia, 1994. ISBN 0-89871-334-X.

S. S. Chughtai and H. Werner. Synthesis of low-order controllers for discrete LPV systems using LMIs and evolutionary search. In ., Proceedings of ECC07, 2007.

A. Farag and H. Werner. A Riccati - genetic algorithms approach to fixed-structure controller synthesis. In *Proc. of the American Control Conference*, 2004.

T. Iwasaki and G. Shibata. LPV system analysis via quadratic separator for uncertain implicit systems. *IEEE Transactions on Automatic Control*, 46(8):1195–1208, 2001.

A. Kwiatkowski, J. Blath, H. Werner, and M. Schultalbers. Application of LPV gain scheduling to charge control of a SI engine. In *Proc. of IEEE International Symposium on Computer-Aided Control Systems Design*, Munich, Germany, 2006.

M. Norgaard. Neural network based control system identification toolbox. Technical Report 00-E-891, Department of Automation, Technical University of Denmark, 2000.

M. Norgaard, O. Ravn, N. Poulsen, and L. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems.* Springer-Verlag, London, UK, 2000.

T. v. d. Boom, M. A. Botto, and J. S. d. Costa. Robust control of dynamical systems using neural networks with input-output feedback linearization. *International Journal of Control*, 76(18):1783–1789, 2003.