Nonlinear Predictive Control of Transients in Automotive Variable Cam Timing Engine Using Nonlinear Parametric Approximation

Dimitry Gorinevsky
Honeywell Laboratories,
47102 Mission Falls Court,
Freemont, CA 94539
e-mail: gorinevsky@ieee.org

Jeffrey Cook
Ford Research Laboratory,
Ford Motor Company,
Dearborn, MI 48121-2053

George Vukovich
Canadian Space Agency,
6767 route de l'Aéroport,
Saint-Hubert, Quebec J3Y 8Y9, Canada

The paper considers design of a predictive Linear Time Varying model-based controller with nonlinear feedforward for regulation of transient processes caused by setpoint step changes in a nonlinear plant. An optimal feedforward control sequence is computed based on an empirical Finite Impulse Response model of the process. Though the control techniques developed in this paper are meant to have more general industrial applicability, a specific automotive engine control application—control of Variable Cam Timing automotive engine—is pursued. An advantage of the proposed controller design in this problem is that no first principle models are required. Instead, nonlinear parametric approximations of a neural network type are being used to describe and identify static nonlinear mappings encountered in the problem. A number of simplifying assumptions and approximations are made to make practical implementation of the proposed scheme possible. Validity of the designed controller is verified by simulation. The proposed "model-free" design can potentially increase flexibility and save labor in development and deployment of such controllers for industrial systems.
[DOI: 10.1115/1.1589029]

1 Introduction

This paper considers an approach to designing a model predictive controller for disturbance compensation in a nonlinear plant. Model predictive control (MPC) is a well-established industrial technology used in many process plants and some other applications, e.g., see [1–4]. In a majority of applications, MPC design and implementation assumes linear time-invariant (LTI) models of a process. These models are usually formulated as FIR (Finite Impulse Response) models and can be conventiently identified from the input–output data collected in experiments with the process. One of the reasons for practical success of MPC is because these FIR models can be easily understood and identified. Tuning and troubleshooting of MPC is conceptually straightforward and directly linked to the model.

According to a recent survey [5], MPC is advantageous in constrained control problems as well as in unconstrained nonlinear plants. Much of the technical MPC literature, including [5], deals with the former types of problems. This paper is focused on the latter types of problems. Related work includes the papers [6–8]. These papers use linearization around a steady-state regime and nonlinear parametric approximation for gain scheduling. Using some of the related concepts, this paper differs in scope by being focused on the rejection of large measurable disturbances.

In this paper, a nonlinear process is linearized in a vicinity of a given reference trajectory. An extension of the MPC for linear time-varying (LTV) plants is considered. In order to formulate and implement an MPC controller for an LTV linearization of a nonlinear process, the following issues need to be addressed: (i) Defining a reference trajectory and control for the linearization; (ii) model identification for the linearized system. A contribution of this paper is in demonstrating a specific nontrivial and novel approach to resolving the above two issues. The approach applies on-line training of the controller and use of nonlinear parametric approximation towards the problem of compensating external measured disturbances.

The reference trajectory about which the system of interest is linearized is affected by disturbance inputs. Here, we consider a practically important class of systems in which disturbances are assumed to be either slow quasistatic changes or instantaneous step changes to respective variables. The step disturbances can be parameterized by initial setpoints and an amplitude of the step. The dependencies of the reference trajectory and reference control on the disturbance parameters are approximated using a nonlinear parametric approximation approach similar to neural networks. Specifically, a form of a multivariable Chebyshev polynomial expansion is used for the approximation.

Though the control technology presented in this paper is meant to have more general industrial applicability, the paper is focused on a specific automotive engine control application—control of a Variable Cam Timing (VCT) internal combustion engine. The use of VCT technology reduces engine emissions but might adversely affect torque response of the engine. An optimal tradeoff between emission and torque performance can be achieved by electronic control of the engine. In our earlier paper [9], a linear predictive VCT controller for small-signal fixed setpoint tracking was designed in a form suitable for embedded engine controller implementation. This paper extends the linear predictive controller developed in [9] toward a nonlinear predictive controller using techniques from [10–12].

The validity of the proposed approach is partially supported by existing theoretical justifications of predictive control in LTV systems [13,14]. Validity of the approximation based model of the nonlinear system in question is verified by simulation. Another important issue is finding an optimal control for the reference transient response to the disturbance. This transient regime cannot be computed from the linearized model and requires control opti-

Contributed by the Dynamic Systems, Measurement, and Control Division of THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS for publication in the ASME JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received by the ASME Dynamic Systems and Control Division, March 8, 2000; final revision, June 26, 2002. Associate Editor: P. Voulgaris.

mization for the full nonlinear system. In the proposed approach, this is achieved by on-line iterative updating in experiments with the system. The update is similar to parametric approximation algorithms studied in [10,12], where convergence proofs are presented. Such updates were earlier applied in [11] to be control of transients in an automotive engine, though not in an MPC setting.

When using MPC in process industry applications, high computational capabilities of the hardware and long sampling intervals can allow for extensive optimization computations in the feedback loop. In this paper a nonlinear MPC controller is developed using a parametric approximation of certain static mappings. Different parametric approximation schemes for static mappings such as neural networks, Radial Basis Functions, or polynomial networks are presently well understood, both theoretically and practically. At the same time, the developed combination of the MPC and neural network technologies represents a promising novel practical approach.

Previous approaches that have been applied to automotive engine control problems similar to the one considered here include traditional linear feedback control designs based on a linearized first-principle nonlinear model of the engine [15,16] as well as a nonlinear recurrent neural network controller [17,18]. The former approach is problem-specific and requires labor-intensive development to achieve satisfactory control performance for nonlinear regimes. The latter approach is more generic and is based on the on-line update of the controller input–output characteristics. There is, however, little analysis available to support engineering methods for design of such controllers. The novel control approach based on the nonlinear MPC methodology proposed in this paper has the advantages of the two above approaches.

2 Process Model Overview

The goal of the VCT engine control, as considered in this project, is to minimize engine emissions in response to rapid changes of throttle (TA), engine speed (RPM), and camshaft timing (CAM). The three disturbance variables—RPM, TA, and CAM—are assumed to change in an uncontrollable way. The controller can modify the Fuel input in response to this change and is required to maintain the Air/Fuel ratio (A/F) at the stochiometric value of A/F=14.64 at all times. The controller can use additional measured variables, such as Mass Air Flow (MAF) and Manifold Pressure (P_m), for computations. This paper uses the model developed in [15,16,19,20]. The VCT engine model was embedded in the simulation software and was used as a black-box for the purpose of the controller development.

The controller design and performance requirements depend on the dynamical characteristics of the disturbances acting in the control loop. In this project it is assumed that TA and CAM can change in a stepwise manner within a certain range and that RPM can change with a bounded increment within some limits. The limits and ranges for the disturbance variable change can, in general, be described as follows:

$$\mathrm{TA}(t) \in [7.6,\ 25], \tag{1}$$

$$\mathrm{RPM}(t) \in [750,\ 2000], \quad |\mathrm{RPM}(t) - \mathrm{RPM}(t-1)| \leqslant 2, \tag{2}$$

$$\mathrm{CAM}(t) \in [0,\ 35]. \tag{3}$$

The controller performance measure is taken to be a mean square deviation of the A/F output from the stoichiometry.

2.1 Model Structure. The first step in controller development is definition and identification of the process model. The model formulated in this section is a nonlinear extension of the linear FIR model considered in [9]. Figure 1 illustrates the input and output variables for the VCT engine control problem. The Manipulated Variable (MV) is the Fuel Input increment, which is added to the fuel estimator cascade loop. The Disturbance Variables (DV) are Throttle Angle (TA), Cam Phase Shift (CAM), and the Engine Speed (RPM). The Controlled Variable (CV) is the Air/Fuel Ratio (A/F) deviation from the stoichiometry, which should be maintained as close to zero as possible at all times and despite the effect of the DV changes.

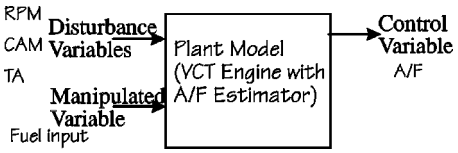

Fig. 1 Input and output variables used for the predictive controller design

The nonlinear predictive controller design uses the following assumptions about the problem:

A1 Variations in the DV and the MV are small around a given regime which is either a steady state-regime or a step change in DV with known amplitude.

A2 Given the condition A1, the plant can be linearized around a given steady-state regime or around a nominal transient regime corresponding to a step change in DV or MV.

A3 Step changes in the DVs result in Finite Step Response (FSR) transients in the CV.

A4 A pulse change in the MV causes a Finite Pulse Response (FIR) change in the CV.

The main practical consideration behind A3 and A4 is that the controlled plant is stable. For a stable plant, a time-limited change in the input variables will always cause an asymtotically decaying transient. For any practical purposes the transient duration can be considered finite, hence, the FIR or FSR models are adequate. The assumptions A1 and A2 are practically reasonable for DV inputs that are either slowly varying or undergo a step change.

2.2 Nominal Transients. Let us consider a single transient process caused by a step change in one or more of the DV inputs. Assume that the step change has occurred at time t_* and prior to this time the system was in the following steady-state regime

$$u(t) = 0, \quad \text{for } t < t_*, \tag{4}$$

$$y(t) = 0, \quad \text{for } t < t_*, \tag{5}$$

$$v(t) = v_0 \equiv [\mathrm{TA}_0\ \ \mathrm{RPM}_0\ \ \mathrm{CAM}_0]^T \quad \text{for } t < t_*, \tag{6}$$

where $u(t)$ is the MV (Fuel) input, $y(t)$ is the CV (A/F deviation from the stoichiometry) output, and $v(t) \in \Re^3$ is the DV input vector for the VCT engine system. At time $t = t_*$ the DV change instantaneously to a new steady state

$$v(t) = v_1 \equiv [\mathrm{TA}_1\ \ \mathrm{RPM}_0\ \ \mathrm{CAM}_1]^T \quad \text{for } t \geqslant t_*. \tag{7}$$

In (7) it is assumed that unlike TA and CAM, the engine RPM never changes instantaneously. This is a realistic assumption since there is inertia in the system associated with the rotating engine parts and the moving vehicle. For automated transmission vehicles, this inertia prevents rapid jumps in RPM for the limited torque generated by the engine. Slower, not instantaneous, variations of the DVs including RPM are modeled and controlled in the further parts of this paper by considering them quasi-static. This essentially means performing gain scheduling in these variables.

After the step change of DV defined by (6) and (7) has occurred, the engine controller reacts by changing the fuel input according to the control agorithm. Since the control algorithm provides for a stable closed-loop operation, it is assumed that after a transient process in MV and CV the system returns to a steady state. Even if the variables might decay asymptotically in the transient process, for all practical purposes the transient can be counted on to have a finite duration, which will be denoted by N_*. The change of the DV and MV in the transient can be mathematically described as

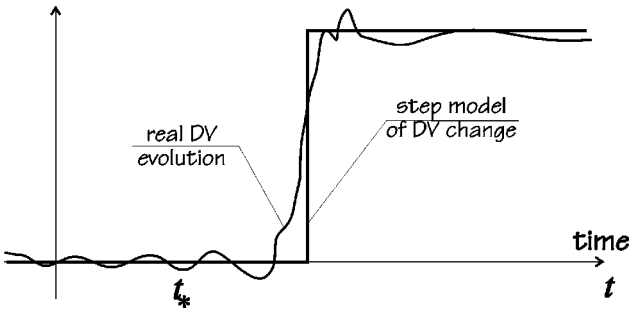

Fig. 2 Identification of the step change in the disturbance history buffer

$$u(t)=\begin{cases} u_*(t-t_*), & \text{for } t_*\leqslant t<t_*+N_* \\ 0, & \text{for } t\geqslant t_*+N_* \end{cases}, \tag{8}$$

$$y(t)=\begin{cases} y_*(t-t_*), & \text{for } t_*\leqslant t<t_*+N_* \\ 0, & \text{for } t\geqslant t_*+N_* \end{cases}. \tag{9}$$

In the predictive control formulation to follow, it will be convenient to describe the transient process with the following two lifted vectors

$$Y_*=\begin{bmatrix} y_*(N_*-1) \\ \vdots \\ y_*(1) \\ y_*(0) \end{bmatrix}, \quad U_*=\begin{bmatrix} u_*(N_*-1) \\ \vdots \\ u_*(1) \\ u_*(0) \end{bmatrix}. \tag{10}$$

The control command sequence U_* in the transient process generated by the control algorithm as well as the respective output transient Y_* are completely defined by the DV values (6), (7) before and after the step change. These values can be collected into a single transient parameter vector

$$p=[\mathrm{RPM}_0 \;\; \mathrm{TA}_0 \;\; \mathrm{CAM}_0 \;\; \mathrm{TA}_1 \;\; \mathrm{CAM}_1]^T. \tag{11}$$

The vector p (11) has only 5 components since in (6) and (7) RPM does not undergo any step change.

The transient regime vectors (10) are defined by the transient parameter vector (11) and these dependencies can be represented as two vector fields of the form

$$U_*(p), \quad Y_*(p). \tag{12}$$

In what follows, the vector fields (12) are approximated by parametric functions of the vector p and the approximations are adaptively updated based on the system closed-loop performance results.

The transient process description (8)–(9) is only valid for step changes in DV or MV that are more than N_* samples apart from each other. It can, however, be extended towards a more general class of disturbances where the DV sequence is either slowly changing or can be approximated by a step. More specifically, consider a DV sequence of the form

$$v(t)=v_*(t)+\delta v(t), \tag{13}$$

$$v_*(t)=\begin{cases} v_0, & \text{for } t_*\leqslant t<t_*+N_*-1 \\ v_1, & \text{for } t\geqslant t_*+N_*-1 \end{cases}, \tag{14}$$

where v_0, v_1 are as defined in (6), (7) and $\delta v(t)$ is assumed to be a small deviation from the nominal step input $v_*(t)$. In practice, the algorithms described in this paper would work even if $\delta v(t)$ is not very small.

In a practical controller, the representation (13), (14) can be obtained on-line based on the available DV data sequence $v(t)$. Consider a DV history with a memory horizon N_* as illustrated in Fig. 2. For this sequence, v_0, v_1, and t_* in (13), (14) are approximation parameters, while $\delta v(t)$, is an approximation error. One feasible method to find the approximation is to do a direct search for all values of t_* within the history time horizon. For each value of t_*, the approximation function $v_*(t)$ (14) is linear in parameters v_0 and v_1. It is straightforward to determine these by minimizing the mean quadratic error of the approximation. Finding a value of t_* corresponding to the minimum quadratic error of the approximation, and taking respective values of the parameters v_0 and v_1 solves the approximation problem.

2.3 Linear Predictive Model in a Vicinity of a Steady-State Regime. In accordance with the Assumptions A3 and A4, the dynamical evolution of the system can be defined through its input and output variables collected over some history horizon N_f. A *linear* dynamical model of the VCT engine in a vicinity of a steady-state regime was considered in [9]. This model has the form

$$y(t)=\sum_{k=1}^{N_f} h(k)u(t-k+1)+\sum_{j=1}^{n_V}\sum_{k=1}^{N_f} h_{V,j}(k)\Delta v_j(t-k+1), \tag{15}$$

where $\Delta v(t)=v(t)-v(t-1)\in\Re^{n_V}$ is a change in DV at time t, $n_V=3$, $h(k)\in\Re$ is an element of the FIR pulse responses for MV, and $h_{V,j}(k)$ $(j=1,2,3;\ k=1,\ldots,N_f)$ describes the FSR element for jth component of the DV vector $v(t)$. The components of the pulse responses in (15) can be joined into vectors of the FIR model parameters in the following way:

$$\overline{h}=[h_F^T \;\; h_{\mathrm{TA}}^T \;\; h_{\mathrm{RPM}}^T \;\; h_{\mathrm{CAM}}^T]^T\in\Re^{4N_f}. \tag{16}$$

Similar to [9] let us introduce the lifted vectors describing historic input (MV, DV) and output (CV) sequences

$$Y_f=\begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-N_f+1) \end{bmatrix}, \quad U_t=\begin{bmatrix} u(t) \\ u(t-1) \\ \vdots \\ u(t-N_f+1) \end{bmatrix},$$

$$\Delta V_t=\mathrm{vec}\left\{\begin{bmatrix} \Delta v(t) \\ \Delta v(t-1) \\ \vdots \\ \Delta v(t-N_f+1) \end{bmatrix}\right\}, \tag{17}$$

where $\Delta v(t)=[\Delta\mathrm{TA}(t) \;\; \Delta\mathrm{RPM}(t) \;\; \Delta\mathrm{CAM}(t)]$ is a 1×3 matrix; $Y_t\in\Re^{N_f}$; by $\mathrm{vec}(A)$, where A is a matrix, we denote a vector obtained by stacking all entries of A into a single vector, column by column; and $\Delta V_t\in\Re^{3N_f}$.

For the controller design, it is convenient to re-write the model (15) in a predictive form. Let us introduce the predicted future histories of the MV, DV, and CV over a prediction horizon N_h

$$\hat{Y}_t=\begin{bmatrix} y(t+N_h) \\ \vdots \\ y(t+2) \\ y(t+1) \end{bmatrix}, \quad \hat{U}_t=\begin{bmatrix} u(t+N_h) \\ \vdots \\ u(t+2) \\ u(t+1) \end{bmatrix}, \quad \Delta\hat{V}_t=0\in\Re^{3N_h}, \tag{18}$$

where as commonly done in predictive control and similar to [9] it is assumed that at any point in time nothing is known about the future disturbances $\Delta v(\tau)$, $\tau>t$, and therefore, they are assumed to be zero.

Assume for a moment that $u(\tau)\equiv 0$, $\Delta V(\tau)\equiv 0$ for $\tau<t-N_f$. In [9], a predictive Linear Time-Invariant model corresponding to the steady-state linearization (15) was considered. This model has the form

$$\begin{bmatrix} \hat{Y}_t \\ Y_t \end{bmatrix}=G^{N_f+N_h}\begin{bmatrix} \hat{U}_t \\ U_t \end{bmatrix}+G_V^{N_f+N_h}\begin{bmatrix} \Delta\hat{V}_t \\ \Delta V_t \end{bmatrix}, \tag{19}$$

where $G^{N_f+N_h}\in\Re^{N_f+N_h,N_f+N_h}$ and $G_V^{N_f+N_h}\in\Re^{N_f+N_h,3(N_f+N_h)}$ are Toeplitz and block-Toeplitz matrices with the entries consisting of

the pulse response elements $h(k)$ and $h_{V,j}(k)$ (15), respectively. In accordance with (15), vector $\hat{Y}_t$ (18) in (19) does not depend on $u(\tau)$ and $\Delta v(\tau)$ for $\tau < N_f$.

2.4 Predictive Model Linearized in a Vicinity of a Transient Process. The LTI model (19) can be generalized to describe a linearization of the VCT engine plant around a transient regime caused by a step change of the DV of the forms (13) and (14). As mentioned above the transient responses are assumed to have a maximum length N_*. Thus, after a single step change of DV occuring at t_*, for $t > t_* + N_*$, the plant dynamics are described by the steady-state model (15). The same model holds for $t < t_*$ because the transient response is causal.

Let us now assume that the step time t_* is such that $t - N_* < t_* \leqslant t$. In accordance with (10), (12), the transient responses (8)–(9) of the MV and CV at time t can be written in the form $u_*(t-t_*;p)$ and $y_*(t-t_*;p)$. Here, the dependence of the transient response on the transient parameter vector p (11) is explicitly emphasized. The assumed dynamical model linearized in the vicinity of a transient response can be mathematically presented in the form

$$y(t) = \sum_{k=1}^{N_*} h(k-t_*;p)[u(t-k+1) - u_*(t-k+1+t_*;p)] + \sum_{j=1}^{n_V} \sum_{k=1}^{N_f} h_{V,j}(k-t_*;p)(k)[\Delta v_j(t-k+1+t_*;p) - \Delta v_{*,j}(t-k+1+t_*;p)], \quad (20)$$

where the nominal and current DV increments $\Delta v_j(t;p)$ and $\Delta v_{*,j}(t;p)$ are defined in accordance with (6), (7), (11), (13), and (14). The model (20) is an extension of the steady-state linearized model (15) and, similar to, (15), $\Delta v(t) = v(t) - v(t-1) \in \Re^{n_V}$ is a change in DV at time t, $n_V = 3$, $h(k) \in \Re$ is an element of the FIR for MV, and $h_{V,j}(k)$ $(j = 1,2,3;\ k = 1, \ldots, N_f)$ describes the FSR element for jth component of the DV vector $v(t)$.

In accordance with (8), (9) the vectors (12) describe the transient process from its beginning at $t = 0$ to its end at $t = N_* - 1$. In the predictive control formulation, the same transient is described by the vectors (17), (18). These vectors have lengths N_h and N_f, respectively. The vectors (12) have length N_*. The vectors (12), (17), (18) describing the same transient are related by the time $\tau = t - t_*$ elapsed after the DV step that caused the transient. The past history and predicted history vectors of the forms (17) and (18) in the nominal transient process can be obtained from (12) in the form.

$$Y_*(\tau,p) \in \Re^{N_h}; \quad \hat{Y}_*(\tau,p) \in \Re^{N_f}, \quad (21)$$

$$U_*(\tau,p) \in \Re^{N_h}; \quad \hat{U}_*(\tau,p) \in \Re^{N_f}. \quad (22)$$

The vectors (21) and (22) are obtained by padding components of the vectors (12) with zeros on either side. In case if the vector p in (21) and (22) corresponds to a zero step, if $\tau > N_f$, or if $\tau \leqslant 0$, the vectors (21) and (22) are zero vectors and the dynamic model (19) can be used.

Similarly, consider the disturbance history vectors corresponding to the nominal step disturbance (13) and (14)

$$\Delta V_*(\tau;p) \in \Re^{3N_h}; \quad \Delta \hat{V}_*(\tau;p) = 0 \in \Re^{3N_f}. \quad (23)$$

For $0 < \tau \leqslant N_f$, all components of $\Delta V_*(\tau;p)$ are zero except for $\Delta V_{*,\tau}(\tau;p) = p(4) - p(2)$, and $\Delta V_{*,2N_h+\tau}(\tau;p) = p(5) - p(3)$.

The predictive dynamics model (19) corresponding to (15) should be replaced by a predictive model corresponding to the plant linearization (20) in the vicinity of the transient dynamics. In accordance with (20), (21)–(23) this predictive model has the form

$$\begin{bmatrix} \hat{Y}_t - \hat{Y}_*(t-t_*;p) \\ Y_t - Y_*(t-t_*;p) \end{bmatrix} = G^{N_f+N_h}(t-t_*;p) \begin{bmatrix} \hat{U}_t - \hat{U}_*(t-t_*;p) \\ U_t - U_*(t-t_*;p) \end{bmatrix} + G_V^{N_f+N_h}(t-t_*;p) \times \begin{bmatrix} \Delta \hat{V}_t - \Delta \hat{V}_*(t-t_*;p) \\ \Delta V_t - \Delta V_*(t-t_*;p) \end{bmatrix}, \quad (24)$$

where $G^{N_f+N_h}(t-t_*;p) \in \Re^{N_f+N_h, N_f+N_h}$ and $G_V^{N_f+N_h}(t-t_*;p) \in \Re^{N_f+N_h, 3(N_f+N_h)}$ are input/output mapping Jacobian matrices. The predictive model (24) shows that for the DV input (6)–(7), (11) and the MV input (4), (8), (22), the CV output should have the form (5), (9), (21).

In this work, the Jacobian matrices $G^{N_f+N_h}(t-t_*;p)$ and $G_V^{N_f+N_h}(t-t_*;p)$ are defined off line as discussed below. The nominal transients described by (12) are determined as a result of an on-line update. This on-line update can be considered as ''training'' of the controller. Once the training is completed and approximate models of the nominal transients are identified, the controller can operate on-line with the transient update disabled. Such operation without the on-line adaptation is preferred in industrial practice.

3 Process Model Identification

The previous section has described the general structure of the nonlinear predictive model used herein for the controller design. This section describes the model in more detail and explains how the model parameters can be identified from the experimental data.

3.1 Model Identification Issues. A steady-state regime of the VCT engine operation can be defined by the following parameter vector:

$$q = [\text{TA RPM CAM}]^T \equiv v, \quad (25)$$

where TA, RPM, and CAM are the average steady-state values of the respective disturbance variables. As a first step in building a nonlinear model, let us approximate dependencies on the vector q (25) for the steady-state plant model (15). The pulse responses (16) define a linearization of the original nonlinear plant in the vicinity of a steady-state regime. The plant model depends smoothly on the parameters (25). Consider the following four smooth maps $\Re^3 \mapsto \Re^{N_f}$

$$h_F(q), \quad h_{\text{TA}}(q), \quad h_{\text{RPM}}(q), \quad h_{\text{CAM}}(q). \quad (26)$$

In this work, the nonlinear mappings are approximated by a regression linear in parameters and with nonlinear regressor vector. For a general mapping $g(q)\Re^3 \mapsto \Re^{N_f}$, where g can represent any of the four vector fields (26), the regression has the form

$$g(q) \approx \sum_{j=1}^{N_a} \gamma^{(j)} \psi_j(q) = \Gamma \Psi(q), \quad (27)$$

$$\Gamma = [\gamma^{(1)} \ldots \gamma^{(N_a)}] \quad (28)$$

$$\Psi(q) = [\psi_1(q) \ldots \psi_{N_a}(q)]^T, \quad (29)$$

where $\gamma^{(j)} \in \Re^{N_f}$ are parameters (weights) of the regression, and $\psi_j(p) \in \Re$ are scalar-valued nonlinear regression functions. Regressions of the form (27) are widely used in Applied Approximation Theory as well as in engineering practice. The regression functions can be polynomials, B-splines, harmonic functions, Radial Basis Functions, wavelets, or other. In this work, Chebyshev polynomials are used as the regression function.

The nonlinear regression model (27) was built as follows. First the FIR responses $g^{(k)}$ for steady-state regimes were identified in the nodes $q^{(k)}$ of a regular grid in a domain of the steady-state regime parameter vector q (25). For each parameter $q^{(k)}$ this was

done by using an input–output experiment with the black box simulation model of the engine as described in [9].

The regression functions $\psi_j(q)$ were chosen semiempirically. These regression functions are designed to perform Chebyshev polynomial approximation of the target nonlinear mappings. The Chebyshev polynomials $C_i(x)$ are a system of polynomials orthogonal on $[-1, 1]$. These polynomials are defined by the following recursive formulas:

$$C_1(x)=1; \quad C_2(x)=x; \quad C_i(x)=2xC_{i-1}(x)-C_{i-2}(x), \quad \text{for } i\geqslant 3. \tag{30}$$

The nonlinear dependences of the pulse response vectors (26) on the parameters (25) are such that the strongest nonlinearity is associated with change of TA. The dependences on RPM and CAM are very close to being affine in TA and CAM. To reflect this, the nonlinear regression function $\Psi(q)$ in (27) was chosen in the form

$$\Psi(q)=\begin{bmatrix} X_1(\mathrm{TA}) \\ X_1(\mathrm{TA})(0.0025\mathrm{RPM}-5) \\ X_1(\mathrm{TA})(0.05\mathrm{CAM}-1) \end{bmatrix}, \tag{31}$$

$$X_1(\mathrm{TA})=[C_1(0.05\mathrm{TA}-1)\ \ldots\ C_{N_{Ch}}(0.05\mathrm{TA}-1)]^T, \tag{32}$$

where $C_j(x)$ are the Chebyshev polynomials. The variables TA, RPM, and CAM in (31)–(32) are scaled such that the range of the scaled variables is close to being $[-1, 1]$.

The weights of the regression models (27) and (32) can be identified by solving the following linear least-square fit problem

$$\Gamma=\arg\min\|\Gamma[\Psi(q^{(1)})\ \ldots\ \Psi(q^{(N_d)})]-[g(q^{(1)})\ \ldots\ g(q^{(N_d)})]\|_F^2, \tag{33}$$

where $\|\cdot\|_F$ is a Frobenius norm of the matrix and N_d denotes number of the nodes in the data grid. The Moore–Penrose pseudoinverse solution Γ to (33), (27)–(29) can be easily computed.

The accuracy of the nonlinear approximation (27), (28), (30)–(33), depends on the maximal degree N_{Ch} of the approximating Chebyshev polynomials in (31), (32). By approximating the pulse responses (26) computed on a grid of the parameter values, it was found that $N_{\mathrm{Ch}}=7$ provides an optimal tradeoff between the approximation accuracy and the number N_a of the regression (shape) functions $\psi_j(q)$ used in the expansion (27). For $N_{\mathrm{Ch}}=7$, the regressor vector $\Psi(q)$ (31), (32) has $N_a=21$ components. The relative accuracy of such approximation was checked on regular grid of the $11\times4\times4=176$ nodes covering the domain of the transient parameters [TA RPM CAM](26) The 176 responses corresponding to the grid nodels were identified from the simulated experiments and had the approximation error better than 3%.

3.2 Approximation Model of the Transient Regime Jacobian. Now consider a nonlinear transient regime corresponding to a step change of DV as defined by (6) and (7) and described by the transient parameter vector (11). If the DV step change occurs at time t_*, the transient process will take a finite duration of time between t_* and t_*+N_*, where N_* describes the transient duration. For $t<t_*$ and $t\geqslant t_*+N_*$ the system can be considered in a steady state and a steady-state linearized model of the process pulse response can be used. This model can be computed through the parametric approximation as described in the previous subsection. Let $g(q_0(p))\in\Re^{N_f}$ and $g(q_1(p))\in\Re^{N_f}$ be such approximations of the pulse responses, where $q_0(p)=v_0$ and $q_1(p)=v_1$ are related to the parameter vector p (11) in accordance with (6) and (7).

The predictive model (24) requires defining the Jacobian matrices $G^{N_f+N_h}(t-t_*;p)$ and $G_V^{N_f+N_h}(t-t_*;p)$. Columns of these matrices are pulse responses of the process linearization around the nominal transient regime. In this work, the transient pulse responses are approximated by assuming linear interpolation (gain scheduling) of the response shape during the transient processes. The pulse responses are taken to be constant and corresponding to the respective steady-state outside the time interval $[t_*, t_*+N_*]$. The pulse response approximation on the interval $[t_*, t_*+N_*]$ has the following form:

$$g(i;\tau;p)=(1-a(i-\tau))g(i;q_0(p))+a(i-\tau)g(i;q_1(p)), \tag{34}$$

$$a(j)=\begin{cases} 0, & j\leqslant 0 \\ j/d_h, & 0<j\leqslant d_h, \\ 1, & j>d_h \end{cases} \tag{35}$$

$$q_0(p)=[\mathrm{TA}_0 \ \ \mathrm{RPM}_0 \ \ \mathrm{CAM}_0]^T, \tag{36}$$

$$q_1(p)=[\mathrm{TA}_1 \ \ \mathrm{RPM}_0 \ \ \mathrm{CAM}_1]^T, \tag{37}$$

where i in (34) indexes the pulse response element, τ is the input pulse application time, and d_h is the model parameter introduced to describe the transient duration. In (34), $g(i;\tau;p)$ corresponds to any of the pulse responses (26).

For the fuel input response, the Jacobian matrix has the form

$$G_{i,j}^{N_f+N_h}(\tau;p)=\begin{cases} h_F(i-j+1;\tau;p), & i\geqslant j \\ 0, & i<j \end{cases}. \tag{38}$$

Expressions for $G_V^{N_f+N_h}$ are defined similarly to (38). The approximations of the Jacobians $G^{N_f+N_h}(t-t_*;p)$ and $G_V^{N_f+N_h}(t-t_*;p)$ are defined solely based on the pulse responses (26) identified in steady-state. The identification procedure described in more detail in [9] is straightforward. A potential downside of such a simplified easy-to-build model is that it might be potentially inaccurate because somewhat arbitrary assumptions were made. The simulation results presented below, however, show that such a model is adequate for the design of the proposed controller and achieves a good disturbance rejection. The performance is achieved because the control scheme developed herein tunes the feedforward parameters in an on-line adaptation process. An error in the Jacobians might slow down convergence on the adaptive update without having a major influence on the performance that is eventually achieved.

3.3 Nominal Transient Regime. In the affine predictive model (24) the Jacobian matrices $G^{N_f+N_h}(t-t_*;p)$ and $G_V^{N_f+N_h}(t-t_*;p)$ defined in accordance (38) characterize the input output dependency slope. The intercept of this dependency is defined by the nominal transient regime vectors (21) and (22). As mentioned above, the latter are obtained by padding components of the vectors $U_*(p)$ and $Y_*(p)$ (12) with zeros. In what follows, the mappings $U_*(p)$ and $Y_*(p)$ defining the nominal transient will be approximated by a Chebyshev network. This approximation cannot be determined from the identification of the steady state regime behavior of the plant as done above for the Jacobian matrices. Instead, the approximations for the mappings $U_*(p)$ and $Y_*(p)$ will be determined as a result of a closed-loop update process with the operating engine. This closed-loop update is the major new algorithm developed in this work and it is described in some detail in the next section. The overall structure of the control loop with the described update of the nominal transient process is illustrated in Fig. 3.

4 Predictive Controller Design

The previous sections have defined the predictive model (24) relating the DV input (6)–(7), (11) and the MV output (4), (8), (10), (22), to the CV output (5), (9), (10), (21) of the process. This section uses this model to build a nonlinear receding horizon controller.

4.1 Optimization Problem and the Control Law. To derive the predictive control optimization problem, let us partition

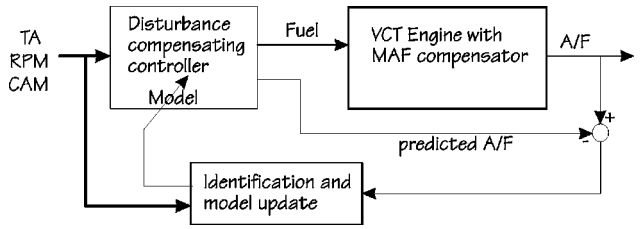

Fig. 3 Transient regime model computation in the designed controller

the matrices $G^{N_f+N_h}(t-t_*;p)$ and $G_V^{N_f+N_h}(t-t_*;p)$ in the model (24) in accordance with the vector partitioning as

$$G^{N_f+N_h}(t-t_*;p)=\begin{bmatrix} G(t-t_*;p) & H(t-t_*;p) \\ \cdots & \cdots \end{bmatrix} \in \Re^{N_f+N_h,N_f+N_h}, \tag{39}$$

$$G_V^{N_f+N_h}(t-t_*;p)=\begin{bmatrix} \cdots & H_V(t-t_*;p) \\ \cdots & \cdots \end{bmatrix} \in \Re^{N_f+N_h,3(N_f+N_h)}, \tag{40}$$

where ... denotes the terms irrelevant for subsequent calculations.

Following the model (24) and notation (18), the predicted future values of the process output can be represented in the form.

$$\begin{aligned}\hat{Y}_t=&Y_*(t-t_*;p)+G(t-t_*;p)(\hat{U}_t-\hat{U}_*(t-t_*;p))\\&+H(t-t_*;p)(U_t-U_*(t-t_*;p))\\&+H_V(t-t_*;p)(\Delta V_t-\Delta V_*(t-t_*;p)).\end{aligned} \tag{41}$$

In accordance with the standard receding-horizon control approaches, [2,4] at each step the following optimization problem will be solved

$$J=\|\hat{Y}_t\|^2+\rho\|\hat{U}_t\|^2\rightarrow\min, \tag{42}$$

where ρ is a scalar controller design parameter defining a tradeoff between performance and robustness. By substituting (41) into (42) and finding an optimum with respect to $\hat{U}_t$, we obtain the optimal predicted control sequence. In the receding horizon algorithm, only the first value of the computed optimal control sequence is applied. The optimal control is then re-computed at the next step. This control law can be presented in the form

$$u(t+1)=-h_C^T(t)U_t-h_D^T(t)\Delta V_t+f(t), \tag{43}$$

$$h_C^T(t)=H(t-t_*;p)^TG(t-t_*;p)R_t\overline{n}, \tag{44}$$

$$h_D^T(t)=H_V(t-t_*;p)^TG(t-t_*;p)R_t\overline{n}, \tag{45}$$

$$\overline{n}=[0\quad\ldots\quad 0\quad 1]^T, \tag{46}$$

$$\begin{aligned}\hat{Z}_t\equiv&-Y_*(t-t_*;p)+G(t-t_*;p)\hat{U}_*(t-t_*;p)\\&+H(t-t_*;p)U_*(t-t_*;p)\\&+H_V(t-t_*;p)\Delta V_*(t-t_*;p),\end{aligned} \tag{47}$$

$$R_t\equiv(G(t-t_*;p)^TG(t-t_*;p)+\rho I)^{-1}, \tag{48}$$

$$f(t)=-\hat{Z}_t^TG(t-t_*;p)R_t\overline{n}, \tag{49}$$

where $G(t-t_*;p)$, $H(t-t_*;p)$ are defined by (39), $H_V(t-t_*;p)$ is defined by (40), R_t is defined by (48), and $\hat{Z}_t$ is defined by (47). Note that the control law (43) has the form similar to that of the predictive control law derived in [9] for the disturbance rejection in the vicinity of a steady-state regime. There are two main features in (43) that reflect the fact that it is derived for control of nonlinear transients. First, in (43) there is a feedforward term $f(t)$ (47), (49) that is aimed at optimization of the nominal transient. Second, (43) is a time-variant control law because it is obtained by linearization of a nonlinear process around a transient regime. This is reflected in the FIR windows $h_C(t)$ and $h_D(t)$ (46) being time-varyng.

4.2 Controller Structure. The overall structure of the designed controller (43)–(49) is illustrated in Fig. 4. The three DVs—TA, RPM, and CAM—are the main inputs of this nonlinear controller; the Fuel MV is the main output. The DVs are fed to the input of a Disturbance Analyzer block that approximates the past DV data sequence as a sequence of step changes and steady-state constant setpoint periods. Note that steady-state constant setpoints can be considered as a special case of a step: one with zero increment. The output of the disturbance analyzer is an approximation of the DV signal coded through the step parameter vector p (11) and the step time t_*. These parameters enter the control computations as described in the previous subsection. As shown in Fig. 4, the nominal (approximated) disturbance is subtracted from the overall disturbance signal to give the approximation residual. This residual is then fed into a linear time-variant predictive controller described by the first two terms in (43). Note that the FIR windows in this linear controller are computed based on the step parameters p and t_* as described in Section 2. The output of the linear controller, which provides compensation for the approximation residual DVs input, is then added to the nominal regime feedforward that provides an optimal compensation for the nominal step disturbance.

The most important transient regime computations, shown as a single block in Fig. 4, are illustrated in more detail in Fig. 5. The step parameter vector p (11) and the step time t_* in (13), (14) allow the computation of the linearized transient regime model. This affine input–output model (41) is described by its gains and offsets. As shown in Fig. 5, these gains and offsets define the nominal disturbance (23) and nominal feedforward (49) to be used in the control scheme illustrated in Fig. 4. The gains and offsets of the model are also used for computing the linear controller gains (46). Further, the linearized model allows the prediction of the A/F output, as described in the following subsection. This prediction is needed in the on-line update scheme presented below.

Consider now in more detail how the transient regime model computations work. This is illustrated in Fig. 6. Based on the step parameter vector p (11) the four pulse responses (26) are approximated for the steady states $q=v_0$ (6) and $q=v_1$ (7) before and after the transient respectively, in accordance with (27), (28), (30), (31), (32), and (33). Based on these FIR approximations, and depending on the DV step time t_* the approximations for the Jaco-

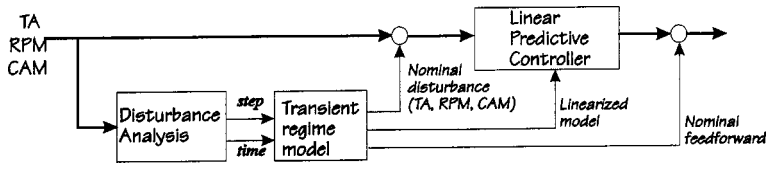

Fig. 4 The overall structure of the designed controller

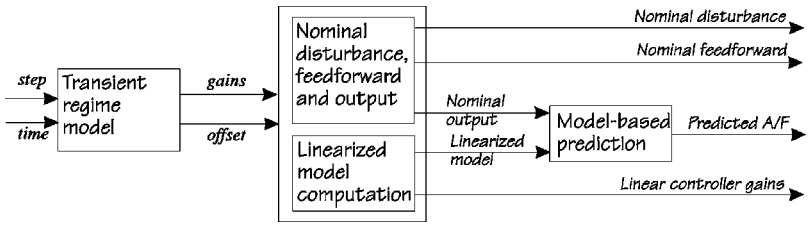

Fig. 5 Transient regime computations in the designed controller

bian matrices $G^{N_f+N_h}(t-t_*;p)$ and $G_V^{N_f+N_h}(t-t_*;p)$ (38) in the affine predictive model (24) are computed as described in previous section.

In addition to this, the step parameter vector p (11) is used to compute an approximation to the nominal transient description vectors (12). This computation is described in more detail in the next subsection. From the vectors (12) and the step time t_*, the transient regime vectors (21) and (22) are computed.

The above described controller structure assumes that the parametric vector field mappings (12) describing the nominal regime dependence on the transient regime parameter vector p are known and can be readily approximated. In fact, these mappings are unknown and their estimates are dynamically updated on-line based on the controller operation data. The next subsection describes the update in more detail.

4.3 Transient Regime Approximation. Similar to the approximations (27), (28), (30)–(33), the vector fields (12) are approximated by expansions of parametric functions of the vector p (11). When presented as linear parametric regressions these expansions have the form

$$U_*(p)=\overline{U}\Phi(p), \quad \overline{U}\in\Re^{N_*,N_a}, \quad \Phi(p)\in\Re^{N_a}, \tag{50}$$

$$Y_*(p)=\overline{Y}\Phi(p), \quad \overline{Y}\in\Re^{N_*,N_a}, \tag{51}$$

where $\overline{U}$ and $\overline{Y}$ are weights of the regression and $\Phi(p)$ is the nonlinear regressor vector containing shape functions for the approximation. In what follows, the form of the regressor vector $\Phi(p)$ is chosen once in the beginning of the controller design process. Conversely, the weight matrices $\overline{U}$ and $\overline{Y}$ are updated on-line based on the controller operation observations.

In order to describe the nonlinear regressor vector $\Phi(p)$, consider the following variables that can be computed from the components of the vector p (11)

$$\mathrm{TA}=(p_2+p_4)/2, \quad \Delta\mathrm{TA}=(p_4-p_2)/2, \tag{52}$$

$$\mathrm{RPM}=p_1, \tag{53}$$

$$\mathrm{CAM}=p_3+p_5, \quad \Delta\mathrm{CAM}=p_5-p_3. \tag{54}$$

The nonlinear regressor vector $\Phi(p)$ is chosen semiempirically in the form of a Chebyshev polynomial network, similar to (30)–(33). Several different forms of the regression vector have been tried in the simulations. The form that was eventually used can be described through the following partial regressor vectors:

$$X_{TA}(\mathrm{TA})=[C_1(0.05\mathrm{TA}-1)\ldots C_{N_{TA}}(0.05\mathrm{TA}-1)]^T, \tag{55}$$

$$X_D(\Delta\mathrm{TA})=[0.1\Delta\mathrm{TA}C_1(0.05\mathrm{TA}-1)\ldots 0.1\Delta\mathrm{TA}C_{N_D}(0.05\mathrm{TA}-1)\ \ 0.05\Delta\mathrm{CAM}]^T, \tag{56}$$

$$X=X_D(\Delta\mathrm{TA})\otimes X_{TA}(\mathrm{TA}), \tag{57}$$

where $\otimes$ denotes a Kronecker (direct) product of matrices and $C_j(x)$ are the Chebyshev polynomials (30). Note that the variables in (55)–(57) are scaled such that the domain of the parameter values corresponds to each of the five scaled variables (52)–(54) changing on the interval $[-1\ 1]$.

The regressor vector $\Phi(p)$ in (50)–(51) is defined with help of (55)–(57) as

$$\phi(p)=\begin{bmatrix} X(\mathrm{TA}) \\ X(\mathrm{TA})(0.0025\mathrm{RPM}-5) \\ X(\mathrm{TA})(0.05\mathrm{CAM}-1) \end{bmatrix}. \tag{58}$$

Note that the regression approximation (50)–(51), (55)–(58) is affine in RPM, CAM, and ΔCAM and defines a strongly nonlinear dependence on TA and ΔTA.

4.4 Transient Regime Control Update. Assume that correct values of the approximation weights $\overline{U}$ and $\overline{Y}$ in the regression approximation (50)–(51) are not exactly known. These weights can be updated on-line by observing the controller operation data.

Consider the following predictive model of the process that can be obtained by considering one-step ahead prediction only (41).

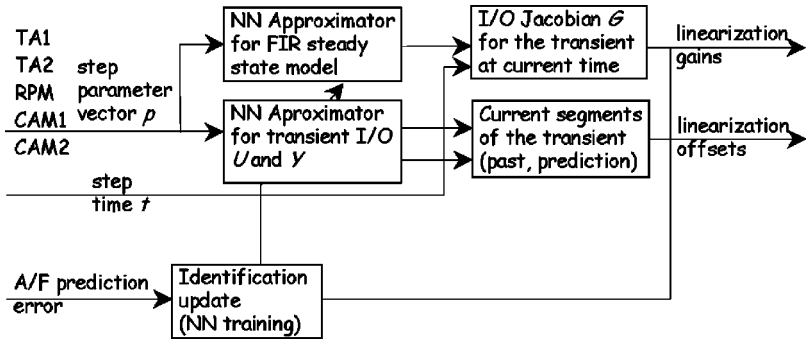

Fig. 6 Transient regime model computation in the designed controller

$$\hat{y}(t+1)=y_*(t-t_*;p)+\overline{n}^T G(t-t_*;p)(\hat{U}_t-\hat{U}_*(t-t_*;p))$$
$$+\overline{n}^T H(t-t_*;p)(U_t-U_*(t-t_*;p))$$
$$+\overline{n}^T H_V(t-t_*;p)(\Delta V_t-\Delta V_*(t-t_*;p)), \quad (59)$$

where $\hat{y}(t+1)$ is a model-based prediction of the process output and $y_*(t-t_*;p)$ is an instantaneous value of the current estimate of the optimal transient obtained as a respective component of the vector $Y_*(p)$ (10).

Define the model prediction error as a difference between the predicted (59) and measured value of the system output

$$e(t)=y(t)-\hat{y}(t). \quad (60)$$

To eliminate the model mismatch error (60), there is a need to update the transient response output in the transient model (10) for the current value of the transient parameter p by

$$\Delta y_*(\tau)=e(\tau+t_*+1). \quad (61)$$

By accumulating the component updates $\Delta y_*(t)$ we obtain the vector update ΔY_*. Different algorithms for updating Y_* can be used in principle, e.g., update at every time sample or once per every transient. In this project the update of U_* and Y_* was implemented once per each transient. In any case, the approximation error can be eliminated by updating the regression weights in the approximation model (51) using the projection update

$$\overline{Y}_+=\overline{Y}+\Delta Y_*\Phi^T(p)/\|\Phi(p)\|^2, \quad (62)$$

where $\overline{Y}_+$ denotes the regression weight matrix (51) after the update.

An error in the model of the transient process (50)–(51) would lead to a suboptimal transient feedforward $U_*(p)$. The optimality of the transient feedforward can be determined directly from the input-output data in the transient. The regression model can then be updated by feeding back the optimality condition error. To derive the optimality condition and the update scheme, assume a linearized model of the relationship between U_* and Y_* in (10) around the nominal transient. For the derivation it will be assumed that the transient duration is the same as the prediction horizon, $N_*=N_h$. In the vicinity of the transient regime, the following affine model relating the transient input estimate $\hat{U}_*$ and the transient output estimate $\hat{Y}_*$ can be assumed

$$\hat{Y}_*=Y_*(p)+G(0;p)(\hat{U}_*-U_*(p)). \quad (63)$$

The optimal transient process defined by the vectors U_* and Y_* should satisfy an optimality condition of the form (42). For the transient process parameters this optimality condition takes the form

$$J=\|\hat{Y}_*\|^2+\rho\|\hat{U}_*\|\rightarrow\min. \quad (64)$$

By substituting (63) into (64) and finding an optimum with respect to $\hat{U}_*$ we obtain the optimal transient control sequence. This sequence corresponds to updating the current estimate of U_* with the update step

$$\Delta U_*=-(G(0;p)^T G(0;p)+\rho I)^{-1}(G(0;p)^T Y_*+\rho U_*). \quad (65)$$

The update step (65) is the update of U_* for current value of the parameter vector p. To achieve this update, the regression weight matrix $\overline{U}$ in the approximation model (50) can be updated by using the projection algorithm as

$$\overline{U}_+=\overline{U}+\Delta U_*\Phi^T(p)/\|\Phi(p)\|^2, \quad (66)$$

$$\overline{Y}_+=\overline{Y}+G(0;p)\Delta U_*\Phi^T(p)/\|\Phi(p)\|^2, \quad (67)$$

where $\overline{U}_+$ and $\overline{Y}_+$ denote the updated values of the respective weight matrices of the regressions (50), (51) and the regression matrix $\Phi(p)$ is as defined by (50)–(51), (52)–(58). The update (67) for the regression matrix $\overline{Y}$ in (51) complements the update of $\overline{U}$. The update (67) is synchronized with (66) so that for the current value of p, the change in $Y_*(p)$ and the change in $U_*(p)$ are related in accordance with the affine model (63).

Note that the derived updates of the regression model weights in (50)–(51) closely follow the parametric optimization algorithms described in [10]. Therefore, the convergence of these updates can be established following the results of [10].

5 Simulation

The overall controller design, implementation and validation procedure was as follows. First the predictive model of the VCT engine was identified and modeled as described above. At this stage, the nominal transient sequences for control U_* and measured output Y_* were initialized to be zero through the entire domain of the transient parameters. Next, the training of the neural network approximator was performed. This training update is performed while running the designed controller with the plant and on-line updating the approximation for the nominal transients. After the training process convereges, the adaptive (training) update of the control law is disabled. The run-time implementation of the controller has fixed weights for approximation of the nominal transients. The two following subsections present the results for controller training and verification.

5.1 Controller Design and Training. The receding horizon predictive control algorithm (48), (43)–(49) requires downloading weights of the Chebyshev network for approximating input–output Jacobian matrices. These weights are computed as described in Section 3 and stored in a disk file.

The transient regime update (50)–(51), (61)–(62), (65)–(67) based on the transient regime regression approximation (50)–(51), (52)–(58) was also implemented as a part of run-time.

After the nonlinear regression approximation schemes in approximating Jacobians (Sections 3.1 and 3.2) and the transient regime (Section 3.3) have been chosen, only a few parameters of the controller design need to be selected. These free controller design parameters were chosen as follows. The control penalty ρ in the performance index (42) was selected to be $\rho=0.04$. The predictive control horison was chosen to be $N_h=N_f=90$. Finally the duration of the transient process was assumed to be $N_*=90$.

The developed algorithms were used in learning the transient feedforward for a generic sequence of step and gradual changes in the disturbance variables as specified in Section 2. In the process of training the Chebyshev network (nonlinear regression) for approximating the transient model, the step changes in the DVs were processed such that only some of them were selected for the transient update. The update of the transient model was only performed if two consecutive steps in the DVs are more than $N_*=90$ samples apart. In this way a complete transient process has time to evolve and converge to a new steady state. Note that this limitation was only applied to the update of the transient model. The feedforward based on the transient model for current transient parameter vector p was computed in all cases irrespective of the time interval between two consecutive step changes of the DVs.

Overall, the training (on-line adaptive update) for the nonlinear Chebyshev network model of the transient was performed for the duration of about 100 000 samples, which corresponds to 5.5 hours of real time. In the process of the training the amplitude of the change of DVs was initially scaled down to a fraction of the full range of the DV change and was gradually expanded before reaching the full specified range of change for the DVs. In this way a reasonable amplitude of the transients was maintained throughout the training process.

5.2 Simulation Results for On-line Operation. After the completion of the training process, the obtained regression weight matrices $\overline{U}$ (50) and $\overline{Y}$ (51) were stored into a disk file. By downloading these matrices into the memory during the controller ini-

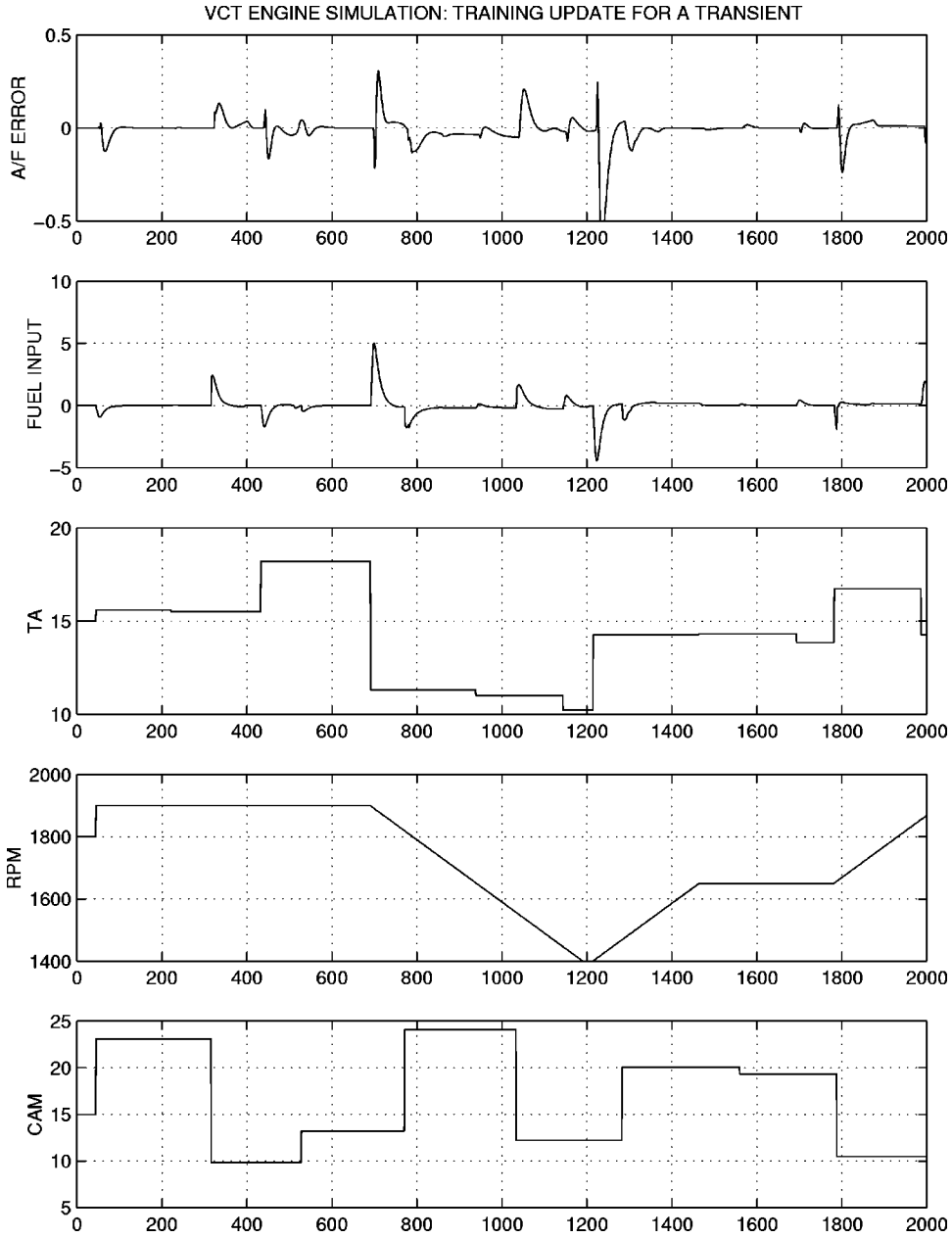

Fig. 7 Simulation results with the designed controller after the end of the training process. The plots top to bottom are: The Air–Fuel ratio deviation from stoichiometry, the control effort (incremental Fuel input), TA, RPM, and CAM histories

tialization, the designed nonlinear controller is able to operate without the necessity of any further training update.

To demonstrate the performance of the obtained controller, it was tested in a simulation run 2000 samples long. Results are illustrated in Fig. 7. The lower three plots in Fig. 7 show the history of the three DVs: CAM, RPM, and TA (bottom to top). The uppermost plot displays the history of the CV output (A/F deviation from the stochiometry). This deviation determines the controller performance. The second upper plot illustrated the MV (Fuel input) history. Overall a good quality of the distrurbance rejection is demonstrated in Fig. 7.

The controller performance can be described by a single loss index

$$J=\sum_{t=1}^{N_s} y^2(t), \tag{68}$$

where $y(t)$ is the plant output (A/F deviation from the stochiometry) at simulation time t and $N_s=2000$ is the simulation duration. For the simulation results in Fig. 7 the loss index (68) is

$$J_1=9.3541. \tag{69}$$

The performance index value (69) indicates excellent controller performance. The results for the designed controller were compared with the results for the controller described in [19,20]. The controller of [19,20] is a nonlinear controller designed assuming a detailed knowledge of the nonlinear dynamics equations for the VCT engine. The loss index for this controller is only about 5% smaller than for the controller developed in this work.

The predictive controller described in this paper is designed using an automated black-box model based process. It has a A/F regulation performance comparable to that of controller [20], which uses a detailed plant model, and depends on the accuracy of

this model. It must be noted here that the controller in [20] had multiple objectives (good torque response and A/F control (and used the model to determine the best cam setpoints to meet emission objectives). The main advantage of the controller designed in this work is that it is based exclusively on the input–output data for the plant and can be used even if a detailed model of the plant is not available.

References

[1] Bitmead, R. R., and Wertz, V., *Adaptive Optimal Control: The Thinking Man's GPC*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[2] Garcia, C. E., Prett, D. M., and Morari, M., 1989, ''Model predictive control: Theory and practice—a survey,'' *Automatica*, **25**, No. 3, pp. 335–348.

[3] Muske, K. R., and Rawlings, J. B., 1993, ''Model predictive control with linear models,'' AIChE J., **39**, No. 2, pp. 262–287.

[4] Qin, S. J., and Badgwell, T. J., 1997, ''An overview of industrial model predictive control technology,'' *Chemical Process Control-V*, edited by J. C. Kantor, C. E. Garcia, and B. Carnahan, pp. 232–256, Tahoe, CA.

[5] Maine, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., 2000, ''Constraint model predictive control: Stability and optimality,'' *Automatica*, **36**, No. 5, pp. 789–814.

[6] De Nicolao, G., Magni, G., and Scattolini, R., 1997, ''Stabilizing Predictive control of nonlinear ARX models,'' *Automatica*, **33**, No. 9, pp. 1691–1697.

[7] Parisini, T., and Zoppoli, R., 1995, ''A receding-horizon regulator for nonlinear systems and a neural approximation,'' *Automatica*, **31**, pp. 1443–1451.

[8] Prasad, G., Swidenbank, E., and Hogg, B. W., 1998, ''A local model networks based multivariable long-range predictive control strategy for thermal power plants,'' *Automatica*, **34**, No. 10, pp. 1185–1204.

[9] Gorinevsky, D., Cook, J., Feldkamp L., and Vukovich, G., 1999, ''Predictive design of linear feedback–feedforward controller for automotive VCT engine,'' *American Control Conference*, San Diego, CA.

[10] Gorinevsky, D., 1997, ''An Approach To parametric optimization of nonlinear system and application to task-level learning control,'' IEEE Trans. Autom. Control, **42**, No. 7, pp. 912–927.

[11] Gorinevsky, D., and Feldkamp, L., 1996, ''RBF network feedforward compensation of load disturbance in idle speed control,'' IEEE Control Syst. Mag., **16**(4), pp. 18–27.

[12] Gorinevsky, D., and Vukovich, G., 1998, ''Control of flexible spacecraft using nonlinear approximation of input shape dependence on reorientation maneuver parameters,'' Control Eng. Pract., **5**, No. 2, pp. 1661–1671.

[13] De Nicolao, G., and Strada, S., 1997, ''On the stability of receding-horizon LQ control with zero-state terminal constraint,'' IEEE Trans. Autom. Control, **42**, pp. 257–260.

[14] Zheng, A., and Morari, M., 1994, ''Robust control of linear time-varying systems with constraints,'' Proc. American Control Conf., Baltimore, MD, June, pp. 2416–2420.

[15] Hsieh, S. C., Stefanopoulou, A. G., et al. 1997, ''Emission and drivability tradeoffs in a variable cam timing SI engine with electronic throttle,'' *Proc. American Control Conf.*, Albuquerque, NM.

[16] Jankovic, M., and Frishmuth, M., 1997, ''Disturbance rejection in SI engine with variable cam timing,'' *Proc. American Control Conf.*, Albuquerque, NM.

[17] Puskoris, G. V., and Feldkamp, L. A., 1994, ''Neurocontrol of nonlinear dynamical systems with Kalman filter-trained recurrent networks,'' IEEE Trans. Neural Netw., **5**, No. 2, pp. 279–297.

[18] Puskoris, G. V., Feldkamp, L. A., and Davis, Jr., L. I., 1996, ''Dynamic neural network methods applied to on-vehicle idle speed control,'' Proc. IEEE, **84**, No. 10, pp. 1407–1420.

[19] Stefanopoulou, A. G., Cook J. A. et al., 1998, ''Control-oriented model of a dual equal variable cam timing spark ignition engine,'' *ASME Journal of Dynamic Systems, Measurement and Control*.

[20] Stefanopoulou, A. G., Cook J. A. et al., 1995, ''Modeling and control of a spark ignition engine with variable cam timing,'' *Proc. American Control Conf.*, Seattle, WA, pp. 2576–2581.