Abstract—This article describes the development and implementation of a practical explicit model predictive control (MPC) approach that allows subcontrollers to receive and accommodate time-varying setpoints and constraints from higher levels in standard industrial automotive controller hierarchies. The MPC approach requires a small computational footprint that is suitable for implementation within a modern electronic control unit (ECU). This article presents the approach, method of implementation, and preliminary results on a 2.2 litre turbodiesel engine.

I. INTRODUCTION

The development of diesel engines requires the engine maker to achieve a successful tradeoff between fuel economy, drivability, and engine cost while respecting the constraints imposed by emissions legislation. As legislated emissions constraints are tightened, diesel engine makers are being required to increase the number of components in an engine - including sensors, actuators, and new subsystems such as diesel particulate filters (DPF) and selective catalytic reduction (SCR) [19]. At the same time the higher levels of performance require increased component reliability and consideration of the interactions between subsystems. This situation has ramifications in both the development of hardware components and the software implementation of control schemes that can deliver the required performance.

The automotive industry uses a wide array of approaches for the development and calibration of control functions and advanced control has begun making inroads in several applications. However, there yet remains a predominance of manually tuned PID control algorithms and lookup tables often in the service of SISO control of subsystems. It is generally recognized that such control structures are expensive and complex to develop and tune\(^1\) and inherently challenging to generalize into MIMO schemes or to accommodate changing constraints on actuators or engine states.

Model based techniques for control design, such as the model predictive control (MPC) approach discussed in this work, may be considered as a systematic approach to control development and calibration along with providing a multivariable control scheme that automatically accommodates actuator and state constraints. However, the implementation of MPC has remained elusive for several reasons. First, it is challenging to implement the run-time component within the limited footprint constraints of modern automotive electronic control units (ECUs). Second, a new control approach must consider the complex interactions with the associated functions within an ECU. Finally one must consider the needs of all of the developers and users who are required to interact with the control at each stage in the life cycle from functional development, software coding and testing, calibration, certification, and maintenance [18].

In Model Predictive Control (MPC) a model of the plant is used to predict the future evolution of the system [12]. Based on this prediction, at each time step \( t \) a performance index is optimized under operating constraints with respect to a sequence of future input moves in order to best follow a given trajectory. The first of such optimal moves is the control action applied to the plant at time \( t \). At time \( t + 1 \), a new optimization is solved over a shifted prediction horizon.

Parallel advances in theory and computing systems have enlarged the range of applications where real-time MPC can be applied [5], [10], [27]. Yet, for a wide class of “fast” applications the computational burden of nonlinear MPC is still a serious barrier for its implementation.

Nevertheless, the capability of handling constraints in a systematic way makes MPC a very attractive control technique, especially for applications where the process is required to work in wide operating regions and close to the boundary of the set of admissible states and inputs. In particular, a control strategy must consider the hierarchical architecture of typical automotive controllers and a subcontroller can be expected to accept and respond to time varying setpoints and constraints that are determined and distributed by a supervisory layer of functionality.

This paper describes several aspects of our work in developing control approaches for diesel engine air induction systems and discusses a subset of the relevant requirements to be met by the proposed control scheme. In particular we concentrate on the performance requirements of these controllers with respect to the highly nonlinear plants, and the requirements for retrofitting a new strategy into existing ECU architectures as a replacement for lower level control components.

The paper is structured as follows. Section II-A overviews the requirements imposed by real-world ECU considerations. Section II-B briefly discusses the physical characteristics of a representative example turbodiesel air induction system and includes a brief survey of prior work in modeling single-stage turbocharged systems with high-pressure exhaust gas recirculation (EGR). Section III presents our proposed explicit model predictive control approach including its parametriza-

---

\(^1\)This cost may be measured in terms of the engineering and test cell time required to develop and tune a controller.
tion in terms of constraints. Section IV describes the implementation of the proposed control strategy on a 2.2 litre turbodiesel engine.

II. PROBLEM DEFINITION

A. Controller Implementation Environment

A controller designed for practical use in automotive applications must consider the restrictive computing environment into which it will be implemented. Depending on the application, current industrial ECUs may be expected to have a processor speed less than 60MHz and less than 3MB of flash memory. The loop time of a controller function depends on its purpose and the sample time of industrial air and EGR handling controllers is typically faster than 100ms. One must also consider that a single ECU will typically run several functions in addition to any function under development. This means that the implementation footprint of any developed function must be kept as small as possible when developing any new control function. The development of control functions should not only consider their coupling through the sharing of limited resources but also their functional interaction.

In current industrial practice a common approach to control architecture is to organize the control functionality in a hierarchical structure as illustrated in Figure 1 (see for example [18]). A supervisory function will determine the state of the engine system, which then distributes information to the individual subsystem controllers in the form of setpoints, feedforward actuator values, actuator constraints, and engine state constraints. Some standard components of the engine state include but are not limited to the engine speed limitations, and special modes such as cold start and regeneration of an aftertreatment device. These signals are generally a function of the operation and ambient environment of the engine and necessarily change as a function of time. The time and effort required for the calibration of the subcontrollers to accommodate such time varying commands is considered to be a critical problem to practitioners (see [6] for example).

![Fig. 1. A common architecture of automotive controller code. The symbols \( y_i(t) \) and \( u_j(t) \) represent the \( i^{th} \) and \( j^{th} \) subsets of sensor and actuator information respectively. The symbols \( f_k(t) \) denote the information transferred to the \( k^{th} \) subcontroller and may include setpoints, feedforward actuator values, and time varying constraints for actuators and engine states. The MPC proposed in this article will replace one (or more) of the illustrated subcontrollers.]()
dependencies of the various mass flow rates on the model states and exogenous inputs [22]. Let \( f \) and \( h \) denote the state update and output functions, respectively:

\[
\dot{x}(t) = f(x, u, v), \quad y(t) = h(x, u, v)
\]

where \( y \) represent the controlled variables, the array \( u \) represents the actuator setpoints to be computed by the controller, and \( v \) represents the exogenous inputs to the system.

A class of nonlinear systems (1) that has received considerable attention from the control community in recent years is that of a single-stage turbodiesel engine with high pressure EGR loop as illustrated in Figure 2. The controlled variables \( y = [y_1, y_2] \) are intake manifold air pressure (MAP) and compressor flow (MAF) respectively and are controlled by \( u = [u_1, u_2] \) being the variable geometry turbine (VGT) and EGR valve actuators respectively. The exogenous inputs \( v = [v_1, v_2] \) are typically considered as the engine speed and the injected fuel quantity. Depending on the modeling requirements the exogenous variables may be extended to include additional parameters such as ambient conditions (temperature, pressure, moisture).

The nonlinearities of such a system can be particularly troublesome for control design. It was noted in [11], [22] that even the sign of the steady-state gain from the VGT actuator \( u_1 \) to the compressor flow \( y_2 \) may change when closing the VGT vanes, particularly when the high pressure EGR valve \( u_2 \) is in an open position. We have also found turbodiesel applications where the sign of the gain from \( u_1 \) to the intake manifold pressure \( y_1 \) changes as the VGT vanes are closed. Thus the consideration of these nonlinearities is crucial to the development of a practical controller. Engine data exhibiting both cases is illustrated in Figure 3.

Various authors use different formulations for the control oriented model of this system. In order to compare approaches, we will write down a linearized version of (1) as follows:

\[
\begin{align*}
\dot{x}_3(t) &= A(\theta)x_3(t) + B(\theta)u_3(t) + E(\theta)v_3(t) \\
\dot{y}_3(t) &= C(\theta)x_3(t) + D(\theta)u_3(t) + F(\theta)v_3(t)
\end{align*}
\]

where \( x_3(t) \in \mathbb{R}^n, u_3(t) \in \mathbb{R}^{n_u}, v_3(t) \in \mathbb{R}^{n_v}, \) and \( y_3(t) \in \mathbb{R}^{n_y} \), are the state, input, exogenous disturbance, and output vector, respectively. The matrices \( A, B, C, D, E, F \) are continuous valued functions of \( \theta : [0, \infty) \rightarrow \mathbb{R}^{n_y} \). Model (2) can be derived from a linearization of (1). In this case the signals in (2) must be interpreted as deviations from a parameter-dependant equilibrium \( \{x_{eq}(\theta), u_{eq}(\theta), v_{eq}(\theta), y_{eq}(\theta)\} \) (also known as trim points) such that \( x_3 = x - x_{eq}(\theta) \), and the coefficient matrices are obtained from derivatives evaluated at the same equilibria such that \( A(\theta) = \frac{\partial f}{\partial x}(x_{eq}(\theta), y_{eq}(\theta)) \) etc. Model (2) can also arise from an LPV or a quasi-LPV derivation then no equilibria points are required except for the origin and the signals in (2) are equivalent to the signals in (1) [17].

Typically the function \( \theta \) is time varying and is a function of the exogenous variables \( v \) in (1). Depending on the modeling, \( \theta \) may also be a function of the state variables \( x \). For air induction in diesel engines the latter quasi-LPV case is discussed in [9], [26].

In [11] a nonlinear model (1) is used and the state vector \( x \in \mathbb{R}^7 \) is comprised of the pressures, densities, and fractions of inert gases for the intake and exhaust manifolds and the turbocharger rotational speed. This formulation is extended in [22] to include two additional states - one each to accommodate the dynamics of the VGT and EGR actuators. In the 3rd order quasi-LPV models of [9] and [26], the states are given by intake manifold pressure, exhaust manifold pressure, and compressor power in the former case and by intake and exhaust manifold pressures and the compressor mass flow in the latter case. In [15] a linear dynamic model with \( x \in \mathbb{R}^2 \) is obtained from a black box identification experiment performed at each of 12 partitions of the engine speed and fuel plane.

In addition to the physical pressure and flow variables discussed above, the example in Section IV considers implementing a constraint on the (measured) engine-out NOx emissions. The control oriented modeling of engine-out emissions requires an additional understanding of the role of the properties of the induced gas on the in-cylinder conditions that influence the formation of NOx. The interested
The following section describes the implementation of an explicit Model Predictive Controller (MPC) [1], [2]. We will first present the standard MPC scheme used. Then, we will enumerate a list of practical implementation issues which have been addressed by our research in order to implement the scheme on an industrial automotive ECU.

We consider a piecewise affine (PWA) discrete-time approximation of the system dynamics

\[
\begin{align*}
x(k+1) &= A_\sigma x(k) + B_\sigma u(k) + B_\sigma^w v(k) + f_\sigma \\
y(k) &= C_\sigma x(k) + C_\sigma^w v(k) + g_\sigma
\end{align*}
\]

(3)

where \(x(k), u(k), y(k), v(k)\) are the state, input, output and exogenous disturbances (either measured or estimated), respectively at time \(kT_s\) where \(T_s\) is the sampling time. The natural number \(\sigma(k) \in \{1, 2, \ldots, M\}\) is the operating point at time \(kT_s\) and it is a function of inputs \(u(k)\), states \(x(k)\) and disturbances \(v(k)\). The set \(\{C_\sigma\}_{i=1}^M\) is a polyhedral partition of the state, input and measured disturbance set. System (3) is subject to the following time varying constraint on inputs and outputs for all \(k \geq 0\).

\[
u(k) \in U(u_{\text{min}}(k), u_{\text{max}}(k)), y(k) \in \mathcal{Y}(y_{\text{min}}(k), y_{\text{max}}(k))
\]

(4)

\textit{Remark 1:} We remark that there is a difference between the scheduling of a plant model in which the equilibrium points or parameters of an LPV model are scheduled as a function of \(\theta\) in (2), and the scheduling of the controller with a function \(\sigma\). It is quite common in practice that the controller scheduling \(\sigma \neq \theta\), where the function \(\sigma\) may represent a simplification of a higher fidelity scheduling function \(\theta\).

To clarify further. The usual interpretation of \(\theta\) is that a nonlinear system (1) is equivalent to (or approximated by) a linear system such as the parameterized model family in (2) for frozen values of the scheduling variable \(\theta(t) = \theta_0\) [17]. In [15], [22], [25] the respective authors design the controller scheduling such that the controller gains are scheduled with \(\sigma\) as a function\(^3\) of engine speed and injected fuel quantity. On the other hand Figure 3 above (and Figures 8 and 10 in [22]) illustrate that the plant remains highly nonlinear at fixed values of engine speed and fuel flow and thus \(\sigma \neq \theta\) for the usual interpretation of \(\theta\).

\textit{Remark 2:} Compared to the original model (3), the model (6b) used in the MPC is augmented with an additional state \(w(k) \in \mathbb{R}^n_w\). The vector \(w(k)\) is an additive disturbance which is used to model the mismatch between the nonlinear system (1) and its PWA model (3) [13].

In problem (5)-(6) the following assumptions are used

\textit{Remark 1:} \(H_0 > H_e\) and the control signal is assumed constant for all \(H_e \leq k \leq H_p\). This allows the reduction of the computational complexity of the MPC scheme.

\textit{Remark 2:} The exogenous disturbance \(v\) is assumed constant over the horizon. If PWA prediction models for \(v(k)\) are available they could be included in the MPC formulation (5)-(6).
A3  Constant region \( \sigma \) over the horizon (6c).

A4  Soft Constraints on outputs, i.e.

\[
\mathcal{Y}(y_{\text{min}}(k), y_{\text{max}}(k)) \equiv \epsilon \Delta \mathcal{Y}(y_{\text{min}}(k), y_{\text{max}}(k)).
\]

Remark 3: Assumption (A3) basically implies that for any given time we simply implement a linear MPC for one member of the set of linear systems. Ideally the assumption should be removed in order to predict switches between affine dynamics over the horizon \( H_p \). This would improve both performance and attractiveness region of the closed loop system. Nevertheless, we have been forced to use assumption (A3) by the current limitations of automotive ECU. In fact, by removing (A3), problem (5)-(6) becomes a mixed integer quadratic program (MIQP) whose explicit solution [2] requires more floating point operations for its evaluations and more memory for its storage.

The optimization problem (5)-(6) can be recast as a quadratic program (QP).

\[
\begin{align*}
\min & \quad \frac{1}{2} \Delta U_{H_i|k}^\top H \Delta U_{H_i|k} + H c^2 + \\
\text{sub. to} & \quad G \Delta U_{H_i|k} \leq W + E \phi(k)
\end{align*}
\]

where \( \phi(k) \equiv [x \ u(k-1) \ u(k) \ w(k) \ y_{\text{min}} \ y_{\text{max}} \ y_{\text{min}} \ y_{\text{max}}] \) and \( \phi(k) \in \mathbb{R}^{n_p} \). Problem (8) is a multiparametric quadratic program that can be solved by using the algorithm presented in [1]. Once the multiparametric problem (8) has been solved, the solution \( \Delta U_{H_i|k}^* = \Delta U_{H_i|k}^*(\phi(k)) \) of problem (5)-(6) and therefore \( u^*(k) = u^*(x(k)) \) is available explicitly as a function of the set of parameters \( \phi(k) \) for all \( \phi(k) \in X_0 \). \( X_0 \subseteq \mathbb{R}^{n_p} \) is the set of initial parameters \( \phi(0) \) for which the optimal control problem (5)-(6) is feasible. The following result [1] establishes the analytical properties of the optimal control law and of the value function.

**Theorem 1**: [1] The control law \( \delta u^*(k) = f_\sigma(\phi(k)) \), \( f_\sigma : \mathbb{R}^{n_p} \to \mathbb{R}^{n_p} \), obtained as a solution of (8) is continuous and piecewise affine on polyhedra

\[
f_\sigma(\phi) = F^\sigma_\phi \phi + g^\sigma_\phi \quad \text{if} \quad \phi \in CR^\sigma_i, \quad i = 1, \ldots, N^\sigma
\]

where the polyhedral sets \( CR^\sigma_i = \{ \phi \in \mathbb{R}^{n_p} | H^\sigma_i \phi \leq K^\sigma_i \} \), \( i = 1, \ldots, N^\sigma \) are a partition of the feasible polyhedron \( X_0 \).

As discussed in [1] the implicit form (8) and the explicit form (9) are equal, and therefore the stability, feasibility, and performance properties are automatically inherited by the piecewise affine control law (9). Clearly, the explicit form (9) has the advantage of being easier to implement, \( M \) lookup tables, one for each operating point \( \sigma \), are uploaded on the ECU and at each time step \( k \) the MPC resorts to selecting the current operating point \( \sigma \) searching for the region \( CR^\sigma_i \) containing the current vectors of parameters \( \phi(k) \) and implementing the corresponding controller \( F^\sigma_i \phi(k) + g^\sigma_\phi \).

The following lists some of the major practical issues which have been encountered while implementing the proposed MPC on an automotive ECU. In this paper we will focus on only the final listed issue.

- **State estimation.** Estimation of the state \( x(k) \) and \( w(k) \) of system (3) is a nontrivial task. We have used a bench of \( M \) Kalman filters which run in parallel in order to smoothen the estimation during switch between different operating points.

- **No offset.** In order to obtain offset-free control with MPC, the PWA system model (3) is augmented with a constant disturbance model (included in \( w(k) \) in system (6c)) which is used to estimate and predict a constant mismatch between measured and predicted outputs. The state and disturbance estimates are used to initialize the MPC problem [4], [16].

- **Limited ECU memory.** We had to modify the explicit implementation in order to be able to run the MPC (5)-(7) in an industrial ECU (even for short horizons). We have used the Karush-Kuhn-Tucker conditions which lead to (9) in order to reduce the memory required for storing \( F_\sigma \) and \( g_\sigma \) in (9). More details can be found in [3].

- **Time-varying actuator and state constraints.** In contrast to techniques such as [15], the proposed formulation has been designed to address constraints that vary arbitrarily as a function of time and thus independently of the state variables or gain scheduling parameters.

- **Constraint satisfaction under steady state disturbances.** For a range of steady-state exogenous disturbance \( v(k) = \bar{v} \) the reference \( y_{\text{ref}} \) might be infeasible for the given input and output constraints and thus not be trackable. At steady-state, the objective function will be composed of two terms with conflicting objectives: satisfy the constraints \( (\rho \leq 2) \) and track \( y_{\text{ref}} \). In addition, model uncertainty and high \( \rho \) weights (usually used to strictly enforce soft constraints) can lead to oscillating behavior and poor performance (see Section IV for experimental evidence). We solved this problem as follows. For each operating point \( \sigma \) we compute a feasible polyhedron \( \mathcal{F}_\sigma \) in the space of the variables \( y_{\text{ref}}, \bar{v}, \bar{w}, u_{\text{min}}, u_{\text{max}}, y_{\text{min}}, y_{\text{max}} \) where \( y_{\text{ref}}, \bar{v} \) and \( \bar{w} \) are references, exogenous disturbances and additive disturbances at steady state, respectively. If at time instant \( k \) the vector \( [y_{\text{ref}} \bar{v} \bar{w} u_{\text{min}} u_{\text{max}} y_{\text{min}} y_{\text{max}}] \) belongs to \( \mathcal{F}_\sigma(k) \), then the reference \( y_{\text{ref}} \) is feasible and no action is taken. If the vector \( [y_{\text{ref}} \bar{v} \bar{w} u_{\text{min}} u_{\text{max}} y_{\text{min}} y_{\text{max}}] \) does not belong to \( \mathcal{F}_\sigma(k) \) then \( y_{\text{ref}} \) is not feasible and the weights in the cost function (5) are modified by bringing \( Q \) to zero and increasing \( \rho \). In this way, the controller will try to satisfy the constraints and do not try the tracking of the infeasible outputs. Below we show how to compute \( \mathcal{F}_\sigma \) in the case \( A_\sigma \) has no eigenvalues at 1. We remark that this approach is similar but less computationally expensive than the one used to compute feasible references at each time step in [16].

From model (6b) we write

\[
y_{k|k} = C_\sigma x_{k|k} + C_\sigma^w v_{k|k} + C_\sigma^w w_{k|k} + g_\sigma
\]

By using model (6b) and considering all values at steady state (denoted by a barred symbol), we can write

\[
\bar{x} = (I - A_\sigma)^{-1} [B_\sigma \bar{u} + B_\sigma^w \bar{v} + B_\sigma^w \bar{w} + f_\sigma]
\]
and thus
\[
\tilde{y} = C_\sigma (I - A_\sigma)^{-1} [B_\sigma \tilde{u} + B_\sigma^w \tilde{w} + B_\sigma^v \tilde{v} + f_\sigma + C_\sigma^w \tilde{w} + C_\sigma^v \tilde{v} + g_\sigma]
\] (10)
We project the set described by (10) and by the constraint sets
\[
\tilde{u} \in U(\bar{u}_k(k), u_{max}(k)), \quad \tilde{v} \in \mathcal{Y} (y_{min}(k), y_{max}(k))
\]
into the space \([\tilde{y} \tilde{v} \tilde{w} u_{min} u_{max} y_{min} y_{max}]\) and then set \(\tilde{y} = y_{ref}\). The result is a polyhedron \(\mathcal{F}_\sigma\) which identifies if the reference \(y_{ref}\) is feasible for a fixed \([\tilde{v} \tilde{v} u_{min} u_{max} y_{min} y_{max}]\) and a fixed operating point \(\sigma\).

IV. PRESENTATION AND DISCUSSION OF RESULTS

Section III has proposed an MPC framework and gave an overview of some major technical obstacles that were overcome in the course of its development in order to satisfy the requirements of Section II. Given the limitations of space, in this Section we provide an on-engine example where we concentrate on the issue of time-varying constraint satisfaction under steady-state disturbance discussed above.

The engine was a 2.2 litre, Euro IV certified, turbocharged diesel engine with a variable geometry turbine and a high pressure EGR loop as illustrated in Figure 2. The ECU bypass was executed using an “on-target” rapid prototyping system (ATI No-Hooks OnTarget [14]) in which the designed controllers were downloaded into a common industrial ECU - a Motorola MPC555 - with a 40MHz processor and 2MB flash memory.

The control variables in question are the intake manifold pressure \(y_1\), the compressor air flow \(y_2\). We will consider imposing constraints on the engine-out NOx emissions \(y_3\).

Active control over engine-out NOx has potential application in forming a part of a coordinated engine and NOx aftertreatment control strategy. The actuators are the VGT vane position \(u_1\) and the EGR valve position \(u_2\). In this example we will concentrate on the operating point defined by fixed values of the exogenous inputs fuel injection quantity of 42.33mm\(^3\)/cycle and an engine speed of 2500rpm, actuator positions of 77\% (0=open, 100=closed) for the VGT vanes and 69\% (0=closed, 100=open) for the EGR valve corresponding to an intake manifold pressure of 207kPa, a compressor flow of 1119mg/cycle and engine-out NOx emissions of 379ppm. The production inline NOx sensor was located downstream of the turbine. The sample time of the control was 100ms.

Referring to the piecewise affine model in (3) and considering the fuel quantity \(v\) as a measured exogenous disturbance of the local linear model, we can write down a linear dynamic model of the plant at this operating point as
\[
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} g_{11}(q) & g_{12}(q) \\ g_{21}(q) & g_{22}(q) \\ g_{31}(q) & g_{32}(q) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} g_{13}(q) \\ g_{23}(q) \\ g_{33}(q) \end{bmatrix} v \] (11)
As discussed in Section II-B above, grey-box models of the turbodiesel engine elements found in the literature may be up to 7th order. In this case, engine in question was found to be well-modeled using first order transfer functions for each of the elements \(g_{ij}(q)\).

\[
g_{ij}(q) = \frac{b_{ij}q^{-1}}{1 - a_{ij}q^{-1}}
\] (12)
the numerical values of the coefficients \(a_{ij}, b_{ij}\) that resulted from a black-box model identification may be found in the Appendix.

To illustrate the proposed approach we will consider the practically-motivated problem of a designer who wishes to implement tracking of intake manifold pressure \(y_1\) and compressor flow \(y_2\) in (11) to exogenous setpoint targets \(y_{1,ref}\) and \(y_{2,ref}\) while respecting exogenous constraints as described in (4) and (6)

\[
u_{min}(k) = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \quad u_{max}(k) = \begin{bmatrix} 95 \\ 95 \end{bmatrix},
\] (13)
\[
y_{3, max}(k) = \text{NOx}(k)
\] (14)
for all \(k\) where \(\text{NOx}(k)\) are time-varying max NOx constraints and all the other outputs are unconstrained. As discussed in Section II-A, \(\text{NOx}(k)\) in (14) could be imposed on a low level subcontroller by a higher level function in the control hierarchy. The identified models and constraints in (11)–(14) were next used in a MPC problem of the form (6) with no terminal constraint and with \(H_x = 1, H_p = 40, Q = 0.25 \cdot I_2, R = 128 \cdot I_2\) in (5). The attractivity region of the nominal closed-loop system can be calculated by using reachable set algorithms for PWA systems [2]. Robustness to parameter uncertainty has been validated by extensive experimental testing. The closed-loop results in Figures 4-6 were obtained using a soft constraint parameter \(\rho = 10^6\) in (6).

Figure 4 illustrates the performance of the controller as it simultaneously tracks changing setpoints in MAP and MAF \(y_1\) and \(y_2\) respectively. The constraint on the engine-out NOx \(y_3\) was set to a very high value of \(\text{NOx}(k) = 2000\text{ppm}\) in order to demonstrate the unconstrained closed-loop performance. Next the NOx constraint was lowered to \(\text{NOx}(k) = 410\text{ppm}\) and the ability of the controller to accommodate this tightened constraint is illustrated in Figure 5 when subject to the same setpoint steps as Figure 4. However, when the NOx constraint is further lowered to \(\text{NOx}(k) = 390\text{ppm}\) we observe the closed-loop instability illustrated in Figure 6.

The reason for this instability may be understood by considering the effect of the soft constraint parameter \(\rho\) in (6) and the degree to which the output constraint has been violated. A large value for \(\rho\) and a larger degree of constraint violation result in the MPC problem generating a very aggressive control action in an effort to return to a feasible position. The combination of such an aggressive controller along with the inevitable model uncertainty associated with real systems leads to the instability illustrated in Figure 6.

The controller is then returned with a smaller value of \(\rho = 10^3\) in (6). As illustrated in Figure 7, using a less
aggressive value of $\rho$ allows the closed-loop to satisfy the tighter constraint on NOx emissions NOx($k$) = 390ppm in (14).

The examples illustrated in Figures 4-7 were selected in order to highlight the delicate interaction between the soft constraint parameter $\rho$ and the output constraints and to motivate the development described in Section III in which the tuning of the controller is switched as a function whether or not the reference $y_{ref}$ is evaluated to be feasible or unfeasible. In fact, experimental results have shown that detuning is very complex when the controller has to operate in wide operating regions subject to larger uncertainties.

The current implementation includes the switched tuning proposed in Section III and several operating points which cover the whole operating region of the engine. The results on the transient FTP cycle will be included in a manuscript which is under preparation.

Fig. 4. Closed-loop run showing response to setpoint $y_{ref}$ changes in $y_1$ (manifold air pressure) and $y_2$ (compressor flow) with controller tuning $N = 1$, $Ny = 40$, $Q = 0.25 \cdot L_2$, and $R = 128 \cdot L_2$ in (5). The constraint on NOx ($y_3$) was set to $NOx(k) = 2000$ ppm in (14) to illustrate the unconstrained closed-loop performance.

Fig. 5. The constraint on NOx emissions is stepped down to $NOx(k) = 410$ ppm in (14) with an aggressive soft constraint parameter of $\rho = 10^6$ in (6).

Fig. 6. The constraint on NOx is further reduced to $NOx(k) = 390$ ppm in (14) for the same controller used in Figures 4 and 5. The combination of an aggressive soft constraint parameter and a higher degree of constraint violation led to an overly-active controller resulting in the illustrated closed-loop instability. (Note that the y-axis has a different scale relative to Figures 4, 5, and 7.)

V. CONCLUSIONS

In this article we have discussed several requirements for the practical control of industrial diesel powertrains. These included the computational requirements imposed by the limited computing resources, the hierarchical architecture of existing software, and time-varying constraints. In addition the requirements imposed by the highly nonlinear air induction problems under consideration must be addressed in any successful control design. This article proposed an explicit model predictive control (MPC) approach designed to work with the time-varying setpoints and constraints as are required from the low-level control layer in the hierarchy.

The proposed approach was demonstrated on production ECU controlling a real 2.2 litre diesel engine in which the VGT and EGR actuators were used to track setpoints on MAP and MAF while respecting a time-varying constraint on engine-out NOx emissions. The example underscores the nature of interaction between output constraints and model uncertainty.

REFERENCES

[1] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. Automatica,
Fig. 7. Controller tuning same as Figures 5-6 but with a reduced value of the soft constraint parameter $\rho = 10^3$ in (6). In contrast to Figure 6 it may be observed that the more conservative approach to output constraints achieves a stable closed-loop while satisfying the constraint $NO_x(k) = 390$ppm in (14).


APPENDIX

The coefficients $a_{ij}$ and $b_{ij}$ in the dynamical model of the engine in (11) and (12) are given by:

$$
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23} \\
  a_{31} & a_{32} & a_{33}
\end{bmatrix}
= 
\begin{bmatrix}
  0.9756 & 0.9244 & 0.9367 \\
  0.9582 & 0.9095 & 0.9342 \\
  0.9639 & 0.9658 & 0.9768
\end{bmatrix},
$$

$$
\begin{bmatrix}
  b_{11} & b_{12} & b_{13} \\
  b_{21} & b_{22} & b_{23} \\
  b_{31} & b_{32} & b_{33}
\end{bmatrix}
= 
\begin{bmatrix}
  0.1235 & -0.1446 & 1.783 \\
  -0.632 & -0.2427 & 1.111 \\
  -0.558 & -0.02537 & 0.147
\end{bmatrix}.
$$