Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Hybrid Model Predictive Control of a Variable Displacement Engine Mode Management

Pawel Majecki*, Michael J. Grimble**
Ibrahim Haskara*, Chen-Fang Chang*****

* ISC Ltd, Glasgow, UK
** University of Strathclyde, Glasgow, UK
*** General Motors, Detroit, USA

Abstract: The use of Hybrid Predictive Control for model-based propulsion control applications is considered. A Variable Displacement Engine (VDE) control problem is considered, involving both continuous-time dynamics and discrete control actions in the form of activating/deactivating the engine cylinders. Hybrid Model Predictive Control is one of the most successful hybrid control schemes and builds upon predictive control methods developed for engine torque management. The ways in which preview information can be used to improve controller performance are considered, as well as simplifications to the hybrid control algorithms to reduce the computational burden. Several hybrid control design approaches are compared using a simulation of a VDE engine. The aim is to optimize the total system behaviour to provide good torque tracking, reduced fuel consumption and smooth cylinder switching. The main contribution is the demonstration that hybrid predictive control can provide a practical solution to an engine control application with the potential to enhance performance and with options to reduce complexity.

Keywords: Variable displacement engine, hybrid systems, predictive control, nonlinear control.

1. INTRODUCTION

A hybrid control system involves the control of plants or processes that feature both continuous-time dynamics and discrete event systems including switching or logical decision-making. Some applications involve multiple modes of operation, while in others there are decision variables that can only assume binary or integer discrete values. Hybrid control is suitable for applications in automotive systems, such as traction control, active suspension control and engine control (Giorgetti, 2006). Most approaches to the control of mixed continuous and discrete event systems use ad-hoc switching based upon heuristic rules. The interaction of discrete and continuous dynamic behavior make control challenging (Borrelli, 2017). Switching can cause unexpected behavior even when dynamics in each mode is simple and understood.

A hybrid systems approach offers a unified design framework that should lead to superior overall control performance (Zhu, 2015). It also enables general control design procedures to be produced for total systems control. From a practical viewpoint, there is the possibility of reducing design effort, since both continuous control and logic switching use one uniform strategy. However, this is still a relatively new research field for applications, and due to the complexity of hybrid systems important issues remain unresolved (Lunze, 2009).

Model Predictive Control (MPC) has been proposed for optimal control of different types of systems, including hybrid systems. The growing interest in MPC is due to the simplicity of handling constrained, multivariable control problems, by making use of a mathematical model to predict the future behavior (Camacho, 2013). The technique provides a systematic approach to control constrained hybrid systems, promising high levels of control performance in conjunction with limited tuning effort. Whether this is achievable in practice is one of the problems considered in this paper.

The use of MPC in hybrid systems is a challenge, because of the discrete variables in the computation of the control action. Recall that at every sample instant MPC solves a constrained optimization problem. For hybrid systems, the optimization involves discrete variables and the problem becomes NP-hard, which means that in the worst case the solution-time grows exponentially with the problem size (Rivotti, 2015). The computational complexity of the online algorithm has limited the application of HPC to slower nonlinear systems.

The solution builds on previous experience using Linear Parameter Varying (LPV) and Linear Time Varying (LTV) versions of MPC for engine torque management in a supervisory control structure. The current work is on hybrid MPC for torque management of a variable displacement engine where the engine displacement volume is represented by a set of quantized values. The aim was to explore the use of HPC algorithms in automotive control applications, to demonstrate benefits, identify challenges, and propose solutions. We also explored algorithms for executing the Mixed-Integer Quadratic Programming (MIQP) problems, investigating simpler alternative solutions to the hybrid predictive control problem and leveraging additional features of the MPC algorithms to improve the hybrid control solution (preview information, gain-scheduling, dynamic weightings).

The HPC problem is formulated in §2, and the VDE nonlinear model and baseline control in §3. Add-on enhancements are described in §4 and hybrid predictive control laws in §5. Performance is demonstrated in §6. Key aspects of HPC development and value of hybrid control in production in §7.

2. HYBRID PREDICTIVE CONTROL PROBLEM

The input-output structure of a general hybrid system involves a mixture of discrete and continuous inputs and outputs. There is a question whether a hybrid system requires a novel theoretical design approach, or whether heuristic solutions are sufficient. In fact, mixed continuous and discrete control actions and effects have been used in control systems for many years. Gain scheduling, artificial intelligence and fuzzy control have been used with some success to implement supervisory control and logic systems. However, systems are becoming more complex with more measurements and actuators, and empirical methods may not be adequate in future.

2.1 Hybrid systems modeling and control

For hybrid control two modelling methods are popular:

i) Piece-Wise Affine (PWA) systems, involving switching between a set of linear models defined according to the operating conditions and where control inputs are continuous variables (zone-based LTI approach)

ii) Mixed Logical Dynamical (MLD) systems, involving both discrete and continuous control variables.

There is an equivalence between PWA, MLD and other hybrid models. However, the representation will normally be chosen based on the physical description of the application. PWA models can be used within gain-scheduling schemes, but are not in a suitable form for transforming control synthesis problems into general compact optimization problems (Joelianto, 2013). MLD-type models were used here in a so-called linear-parameter-varying (LPV) form. These can describe general classes of hybrid systems but the modeling and design problems may become more complex.

A hybrid system can be thought of as being in one of several 'modes'. In each mode the system behavior is described by continuous dynamics, and the mode switching occurs due to 'events' or according to control inputs (where mode transition is caused by a control signal or autonomously, because of the dynamics of the system i.e. crossing a boundary/threshold). Several methods have been developed for the control of special classes of hybrid systems. A body of literature exists on stabilizing controllers using Lyapunov arguments and linear matrix inequalities (Lazar, 2006). Most approaches are based on solving a Mixed Integer Programming (MIP) problem. If the problem of interest involves LPV models and constraints, a quadratic cost-function, and some optimization variables that have integer values (x_z), then a mixed integer quadratic programming problem results, which can be represented as:

$$\min_x \left(0.5x^T Hx + f^T x\right) \ \textit{subject to}: Ax \le b,\ x = \left[x_c^T, x_z^T\right]^T \quad (1)$$

If all variables are continuous ($x = x_c$), the problem collapses to a quadratic program (QP), for which efficient algorithms exist. On the other hand, the MIQP problem is inherently non-convex and therefore challenging to solve. Some variants of branch-and-bound methods are used to obtain a solution.

2.2 HPC Problem Formulation

The basic MPC approach of optimizing a cost function over a finite and receding control horizon can be applied directly to hybrid systems. The problem involves three components:

1. Dynamic model system: $x(t+1) = f(x(t),u(t))$ (1)
2. Constraints: $g(x(t),u(t)) \le 0$ (2)
3. Cost: $J_n = \sum_{i=1}^{N} e_{n+i}^T Q e_{n+i} + \sum_{j=0}^{N_u-1} u_{n+j}^T R u_{n+j}$ (3)

where N is the prediction horizon, N_u is the control horizon, $e = r - y$ is the tracking error, and $Q \ge 0$ and $R > 0$ are constant diagonal error and control weighting matrices. In the case of linear or LPV models, constraints, and a quadratic criterion, the HPC problem can be recast as a MIQP optimization problem (1). These are classified as NP-hard, which means that in the worst case the solution time grows exponentially with problem size (i.e. number of integer/binary variables). This can be compared with the polynomial complexity O(n^3) of the standard QP problem, which is convex and has a unique solution (if it exists). However, systems with discrete variables are not convex (or smooth), and solutions require a search through the space of possible discrete variables.

3. VARIABLE DISPLACEMENT ENGINE

3.1 VDE Nonlinear Model

In a variable displacement engine, cylinders can be activated and deactivated, making the problem hybrid in nature. The VDE model used in this work has been developed based on a 2.4L Chevrolet Equinox engine, introducing the cylinder deactivation feature, and considering the number of active cylinders (n_{cyl}) as a control variable (Fig. 1). For a detailed model of the engine, see Majecki et al., 2017. To simplify and focus on the hybrid aspects, the fuel path (i.e. lambda control) was decoupled from the air path (torque control).

Intake manifold dynamics (one-state):

$$\dot{P}_{im} = -k_1 VE(P_{im}) N n_{cyl} + k_2 \Psi(P_{im}) u_{th} \quad (4)$$

where $\Psi(.)$ is the throttle-flow function dependent on pressure ratio and the volumetric efficiency η (or VE) was identified from driving cycle data as a regression model. For control design, this model was discretized using the Euler method.

Total cylinder air charge (defined as output to address fuel economy requirement):

$$CAC_e = k \cdot VE(P_{im}) \cdot P_{im} \cdot n_{cyl} \quad (5)$$

Torque model: The brake torque TQ can be expressed as:

$$TQ = TQ_{combustion} - TQ_{pumping}$$

$$= c_1 \cdot CAC \cdot f(N, SA) - c_2 \left(P_{em} - MAP\right) \quad (6)$$

where SA is spark advance and P_{em} exhaust manifold pressure. Torque is modeled as a simple regression and the dependence on lambda was dropped (stoichiometric mixture assumed).

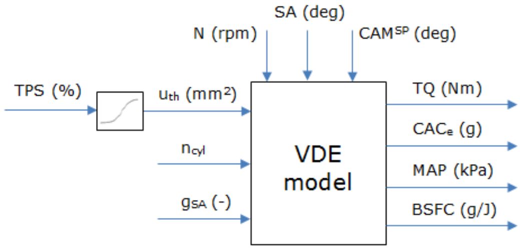

Fig. 1 I/O structure of Variable Displacement Equinox engine model

Pumping losses were assumed proportional to the pressure difference between the exhaust and intake manifolds and a MAP term was included (here MAP = P_{im}). Losses due to friction were not modeled explicitly, and model for torque:

$$TQ_n = p_1 + p_{10}(N_n)\cdot CAC_{n-2} + p_3 \cdot MAP_{n-2}$$

$$+p_{20}(SA_{n-1})\cdot CAC_{n-2}\cdot g_{SA,n-1} \tag{7}$$

where $p_{10}(N) = p_2 + p_3 \cdot N + p_4 N^2$, $p_{20}(SA) = p_5 \cdot SA_{n-1} + p_6 SA_{n-1}^2$ and n here denotes discrete time index.

The spark multiplier g_{SA} was included as a control input that represents the effect of the spark-timing deviating from its nominal (MBT) setting.

The input/output model is shown in Fig. 1. There are 3 control inputs and the throttle is primarily used to track the torque demand, n_{cyl} is switched to maximize MAP and hence minimize fuel consumption (assuming torque set-point is followed), and the spark multiplier g_{SA} ($g_{SA} \leq 1$) should smooth out torque transients. The effective control input u_{th} is obtained from the throttle position through a static monotonic map (Majecki *et al*., 2017), but CAC$_e$ is not directly measurable.

3.2 Control Objectives

The objectives are tracking the torque command, minimizing fuel consumption and limiting transient effects due to cylinder switching. Large displacement engines are good for maximum torque and acceleration, while small engines are better for fuel economy. Under part-load conditions, large engines often operate with a small throttle opening, leading to a partial vacuum in the intake manifold. This makes it harder for the engine to draw the air and pump it through to the exhaust, giving larger pumping losses and wasting power. Ideally, the throttle should be wide open at all times and intake manifold pressure (MAP) should approach ambient pressure (for engines without turbocharging). Variable displacement engines attempt to achieve this by deactivating some cylinders if torque demand is low. A VDE involves switching the number of active cylinders, (denoted n_{cyl} or simply n).

3.3 Baseline Controller and Spark Compensation Scheme

The baseline controller involves a classical low-level torque controller, gain-scheduled cylinder switching, and spark compensation. Lookup table map n_{cyl}*(TQSP,N) was obtained from the engine static characteristics (Fig. 2). This returns the minimum number of cylinders needed to produce a demanded torque for a given engine speed. This number, ncyl*, is treated as a known input disturbance by the low-level controller.

A feature of the VDE problem is the instantaneous effect of the cylinder switching on total cylinder air charge and torque.

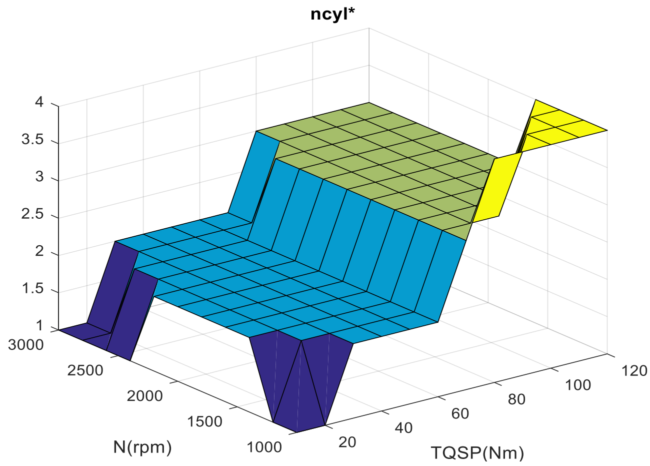

Fig. 2 Plot of n_{cyl}* (Minimum No of cylinders for requested TQSP)

This usually results in torque 'spikes' and non-smooth transient responses. These can be mitigated by introducing a bespoke spark compensation scheme. To smooth out the torque transients, a common solution is to modulate the spark timing, which has an instantaneous effect on the torque produced, and ideally reduces the cylinder switching effect. There are other means of torque smoothing during cylinder deactivation/reactivation like using the transmission system, but the spark compensation method will be applied here to illustrate the benefits of preview action in hybrid MPC. The spark compensation term is defined as:

$$SA = SA_0(N, TQ^{SP}) + \delta SA(n_{cyl}^{old}, n_{cyl}^{new} N, TQ^{SP}) \tag{8}$$

where SA_0 is the baseline MBT value and δSA is the additional spark advance or retard. Recall the g_{SA} multiplier in (7) should be less than unity, while the possible TQ reduction is also normally limited, imposing a lower limit on g_{SA}. The asymmetry leads to the Spark Compensation Scheme (SCS) being composed of two different control strategies:

1. Cylinder deactivation: To maintain TQ at its current setpoint with a reduced ncyl, CACc and hence MAP must be increased, but the pressure build-up involves a dynamic response. The SCS therefore prepares the switch, by first increasing MAP to the right level (by opening the throttle). At the same time the multiplier g_{SA} is adjusted to maintain TQ. Once MAP reaches its target value the cylinder switch is executed and g_{SA} reset to its nominal value of unity.

2. Cylinder reactivation: When reactivating a cylinder, no preparation phase is necessary. Since cylinder reactivation requires a temporary reduction in the generated torque, this can be handled by reducing the spark compensation multiplier (g_{SA}) multiplier from its nominal value of 1. The requested n_{cyl} switch can therefore be applied without any delay.

The above compensation logic is independent of the low-level controller. The solution 'in principle' is illustrated in Fig. 3. In theory, the algorithm should not lead to a TQ transient, but in practice, due to model mismatch and g_{SA} limits, there will usually be some residual TQ response. The SCS is a bespoke scheme that addresses the issues of the VDE control problem. One of the goals was to investigate if the desired behavior could be achieved with a more generic optimal control scheme.

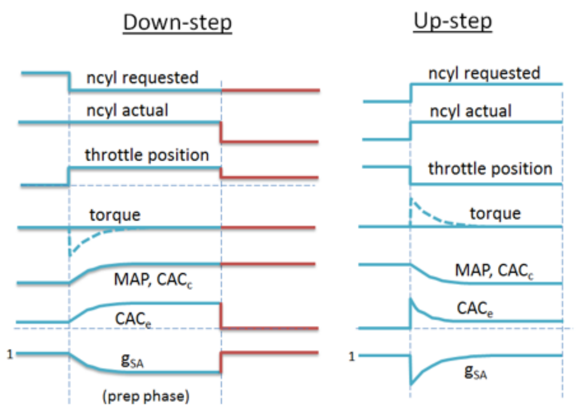

Fig. 3 Conceptual illustration of the Spark Compensation Scheme: Cylinder deactivation (left) and reactivation (right). TQ transients without SCS indicated by dashed lines.

3.4 VDE Hybrid Control Problem

For the design of the MPC, a suitable form of the VDE model is required. A nonlinear model could be used, but the computational complexity of MPC involving nonlinear models is still a limiting factor for real-time use. Questions of uniqueness and existence of the solution to the optimization problem also arise. LPV models provide an alternative to approximate the dynamics of nonlinear systems, combining the modeling capabilities of Linear Time-Invariant (LTI) systems with scheduling for adaption to the nonlinear dynamics. The simulation model was based on the physical equations, but the design model was simplified for controller synthesis. It can also be found using system ID (Toth, 2010).

The system model includes the number of active cylinders n_{cyl} and the spark multiplier as control inputs. The Quasi-LPV (Q-LPV) VDE model has the following state-space structure:

$$x_{n+1} = A(p_n)x_n + B(p_n)\cdot u_n$$
$$y_n = C(p_n)x_n + D(p_n)\cdot u_n + g(p_n) \qquad (9)$$

where matrices A, B, C, D and the affine term g vary with the scheduling parameter vector p_n:

$$p = \left[MAP, N, ICam, ECam, T_{im}, P_{amb}, T_{amb}, SA_{n-1}, x_{CAC2}\right] \qquad (10)$$

and with the following definition of signals:

$$y_m = \begin{bmatrix} MAP \\ TQ \\ MAF \end{bmatrix}, y_c = \begin{bmatrix} TQ \\ CAC \end{bmatrix}, u = \begin{bmatrix} u_{th} \\ g_{SA} \\ n_{cylB} \end{bmatrix}, x = \begin{bmatrix} x_{MAP} \\ x_{CAC1} \\ x_{CAC2} \\ x_{gSA1} \end{bmatrix} \qquad (11)$$

For simplicity, the ICam, ECam and SA_0 are not assumed to be functions of n_{cyl} and are obtained from the original lookup tables (i.e. calibrated for $n_{cyl} = 4$) although in actual practice they will follow from engine calibration. The control variable n_{cylB} is related to n_{cyl} by $n = n_0 + n_B$, with n_0 set to unity.

4. PREDICTIVE CONTROL ADD-ONS

MPC algorithms for hybrid systems can be customized, introducing additional features aimed at improving the control performance. This can be done by improving the fidelity of the prediction model, exploiting information about future signal profiles, or introducing adaption in the controller parameters. Different enhancements or add-ons are now discussed.

4.1 Dynamic Weightings and Gain-Scheduled Weights

Traditional optimal control problems involve constant matrix weighting functions on error (or output) and control signals. However, frequency sensitive weightings can improve performance and this may be used in both hybrid and non-hybrid predictive control systems. This involves using dynamic cost-weighting functions applied to the error and control signals. For example, a dynamically weighted error:

$$e_{p,n} = P_c(z^{-1})\left(r_n - y_n\right) \qquad (12)$$

where the transfer $P_c(z^{-1})$ normally includes high gain at low frequencies, providing integral action in the controller. The e_p term replaces the error signal e in the cost-function (3).

The use of dynamic cost weightings when they are applied to continuous-time signals is well understood but the way in which the system behaves for weightings on integer variables is not clear. A high-pass filter was introduced to prevent chattering or excessive switching of the cylinders.

Gain scheduling the weighting functions provides a means of introducing some logic-based decisions whilst using the optimal control framework. Different objectives may be difficult to meet with a single fixed set of weightings. Gain-scheduled weights can be switched, or be allowed to decay exponentially to their base values, which is very flexible. In VDE control the idea was used to force the spark multiplier g_{SA} to assume the nominal value of 1 outside switching, while still using it for compensation in the immediate vicinity of the switching instants. The gain-scheduled design reflects some extent the spark compensation scheme. In this case, the spark weighting is scheduled with the ('feedforward') n_{cyl} reference step and persists for a pre-set number of events, after which it reverts to the base settings.

4.2 Reference control inputs

The standard MPC criterion can be modified by replacing the absolute control terms with deviations from some reference values. This can be considered a form of feedforward, since under large weighting on this term the control approaches the reference (In the case of the n_{cyl} input, this will be the best switching sequence based on non-dynamic computations). A smaller penalty will give the controller more freedom to deviate from the reference. The modified criterion:

$$J = E_p^T E_p + (U - U_{ref})^T R(U - U_{ref}) \qquad (13)$$

The reference signals for the three HPC control inputs correspond to their optimal steady-state values and are defined based on the pre-computed look-up tables:

$$n_{cyl}^{ref} = f_n(TQ^{SP}, N),\ u_{th}^{ref} = f_{th}(TQ^{SP}, N, n_{cyl}^{ref}),\ g_{SA}^{ref} = 1 \qquad (14)$$

4.3 Preview

By adding information about the future values of the signals using preview a modified control action results; improving performance by exploiting future knowledge. Preview involves updating the cost-function and constraints, given future knowledge of reference signals/disturbances. A more sophisticated use of preview is to update the prediction model matrices. MPC can provide anticipative control action having such information. The introduction of preview in the VDE control problem is now considered and the modifications to add preview in the MPC control law. The 'preview' action can be used to estimate the engine torque demand over the next few seconds, such that the hybrid predictive control algorithm improves torque tracking, by appropriate cylinder switching. Preview information can be introduced in the VDE hybrid LPV-MPC control law in different ways. Possible approaches to exploit future information include acting on the future reference, or on the known steady-state value of the control input, or based on the structure of the LPV model as follows:

Reference signal: The sequence of future values of the reference signal $r(t)$ is denoted as $R_N(t)$ = [r(t), r(t+1), … r(t+N)]^T, where N is the prediction horizon. It is included naturally in MPC and any preview information of $R_N(t)$ is used automatically in both linear and nonlinear MPC.

Input feedforward: Steady-state mappings can be defined for the system control inputs, as static functions of the scheduling parameters and the reference signal $u_{ref} = f(p,r)$. These can be considered as feedforward controls, or reference inputs. The future u_{ref} sequence, is based on $R_N(t)$ and denoted as $Uref_N(t)$.

LPV model update: The VDE LPV model scheduling parameters are directly dependent on input/state model variables. Three options are possible:

1. $U_N(t)$ = [uopt(t-1)]$_N$ - hold the previous control action.
2. $U_N(t)$ = $Uopt_N(t$-1, 2:N) - use the full optimal sequence computed in previous step.
3. $U_N(t)$ = $Uref_N(t)$ - use estimated steady-state reference inputs. If no preview available let $Uref_N(t)$ = [uref(t)]$_N$.

The use of preview information can be very effective in improving tracking of sudden reference changes. As the control action starts early, the error is distributed before and after the setpoint change, and this results in less overshoot.

5. PREDICTIVE CONTROL POLICIES

In this section, hybrid predictive control schemes for driving the VDE system are presented. Controllers are designed using the LPV model of the VDE. The general schematic diagram of the different hybrid VDE controllers is shown in Fig. 4. Engine torque is assumed measured, but an estimator such as a Kalman Filter will normally be needed.

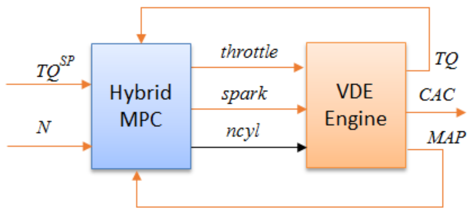

Fig. 4 Hybrid MPC structure for VDE problem (discrete input highlighted in black)

Hybrid controls investigated: There are many hybrid control strategies compatible with the VDE control problem formulation and several were investigated. In this summary, three representative solutions are presented:

1. Scheduled Predictive Control (baseline controller with low-level MPC control)
2. Full optimal HPC scheme (using MIQP solver)
3. Quantized Predictive Control (using QP solver followed by n_{cyl} rounding/quantization).

All controllers use dynamic error weightings to provide integral action (and offset-free tracking). The use preview is also explored, assuming future knowledge of torque profile.

5.1 Scheduled Predictive Controller (nGS-MPC)

The benchmark control scheme is referred to as nGS-MPC (**G**ain **S**cheduling of **n** cylinders, with low-level **MPC** control). It is similar to the baseline controller described in Section 3.3, but with MPC in the low-level controller. The number of active cylinders is obtained from a simple look-up table and is treated as a disturbance input by the MPC. The bespoke spark compensation scheme is included to minimize torque transients due to switching as shown in Fig. 5.

5.2 Hybrid Predictive Controller (HPC)

The controller is based on a HPC where the cost-function has both continuous (throttle position, cylinder fuel charge) and discrete (number of active cylinders) variables. The resulting optimization problem is MIQP. A simplified block diagram of this scheme is shown in Fig. 6. The HPC algorithm does not include any bespoke spark compensation scheme. One reason to investigate the HPC system was to determine if effective spark action could be obtained 'naturally.' It was found that utilizing "preview" information within the HPC allowed the controller to prepare for the switch and to improve the transient performance under switching. Static 'control reference' values in (13) were also included, which improved responses. A large weighting on this control deviation term resulted in the control action approaching the "control reference" level and smaller penalty allowed the controller to deviate.

5.3 Quantized Predictive Controller (QPC) The main issue of the optimal HPC design is the computational burden to solve the MIQP problem. This problem can be addressed by a quantization approach. The Quantized Predictive Controller (QPC) formulates the HPC problem by artificially removing the integrality constraints on the control variable n_{cyl}. This converts the MIQP problem to an easier to solve QP problem.

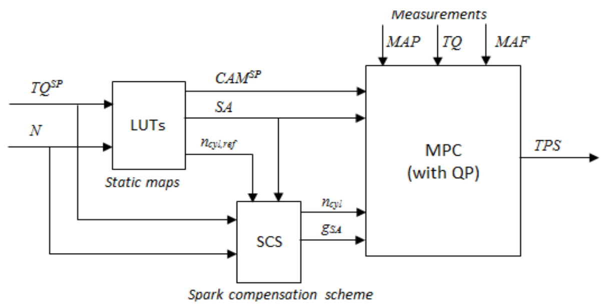

Fig. 5 nGS-MPC controller structure. The measurements of MAP, TQ and MAF are used by the Kalman Filter (inside the MPC block)

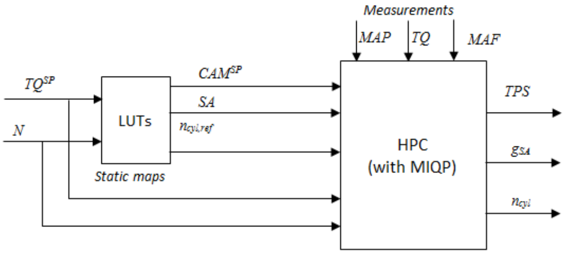

Fig. 6 HPC structure. The measurements of MAP, TQ and MAF are used by the Kalman Filter (inside HPC block) to correct estimates

The continuous n_{cyl} solution, denoted n_{cylC}, is then simply rounded off to the closest integer. This value is passed on as the final control, and fed to the Kalman Filter. A simplified block-diagram of this scheme is shown in Fig. 7. This approach is suboptimal, but is straightforward to execute and gave results that were close to the optimal HPC solution.

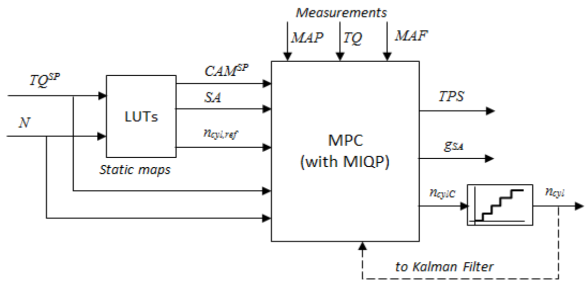

Fig. 7 QPC structure. Measurements MAP, TQ and MAF are used by Kalman Filter (inside MPC block) to correct the state estimates

6. Simulation Results

The simulation results for the VDE hybrid control problem include: (a) effect of cylinder switching on torque responses and performance; (b) introduction of baseline control solution in the form of nGS-MPC scheme; (c) comparison between baseline and HPC controllers for several scenarios; (d) comparison against the simplified quantized solution; (e) effect of preview information. The MIQP calculations were executed using the BONMIN solver included in the Matlab OPTI Toolbox (Bonami *et al*., 2008).

6.1 Basic VDE simulations

Figs. 8a/b show the results of tracking torque reference ramp profiles with nGS-MPC controller, for cases without and with spark compensation. The TQSP slowly varying was chosen to visualize the effect of uncompensated cylinder switching on the torque transient response. The engine speed (not shown) was maintained at 1700 rpm. Note the torque setpoint TQSP = 40 Nm can be achieved with two cylinders. When TQSP increases, an additional cylinder is switched in at about TQSP = 65 Nm, when the torque starts diverging from reference. A reverse switch is performed during the down-ramp, when a cylinder is deactivated to improve fuel economy (see CAC plots that also represent fuel consumption). With no SCS, the cylinder (de-) activation results in undesirable and significant torque spikes. The SCS nullifies these effects using the engine model and adjusting the spark timing. As in §3.3, the SCS algorithm is different for cylinder deactivation and results in a small delay between the request and actual switch (Fig. 8b).

6.2 Optimal HPC results

The HPC designs include "preview information" and "control reference" (u_{ref}) values in the criterion. The Figs 9a/b show the ramp profile responses for: (1) No preview, (2) Preview on the TQSP/u_{ref} predictions and future LPV model variations. Preview generally gives smoother and tighter TQSP ramp tracking and a reduced torque transient during the cylinder down step. It is important to select the relative penalties on the torque tracking and spark action, to avoid the latter deviating too much from the nominal (unity) during the constant-n_{cyl} periods (spark should only deviate from its MBT value to compensate for torque spikes during cylinder switching).

Evaluation on driving cycle data: The performance of the HPC controller with and without preview was also assessed on two fragments of the US06 driving cycle data that include variations in torque demand and engine speed. The two fragments represent 'less aggressive' and 'more aggressive' driving profiles, as measured by the average rate of change of TQSP variations. The results are shown in Fig 10 and Fig 11. Several performance metrics were computed as in Table 1.

Table 1 Comparison of HPC performance with and without preview information, for two fragments of US06 driving cycle.

Drive cycle	preview	mse(TQ_{err})	max($\lvert TQ_{err}\rvert$)	sum(CAC_{eng})
less aggressive	no	24.2	35.3	10593
	yes	23.4	25.8	10567
aggressive	no	19.7	42.0	6601
	yes	11.1	34.1	6600

As with the ramp scenario, the pre-emptive control action due to preview generally leads to tighter tracking of torque demand. The improvements in torque tracking are greater for the more aggressive driving fragments (step TQSP changes).

6.3 Quantized MPC

The QPC approach was compared with the HPC controller using typical scenarios on the Equinox VDE such as step and ramp responses, and fragments of driving cycles. The difference between the two controllers is minimal, and cylinder switching was identical in both cases. Similar performance was found because the "control reference" signals (u_{ref}) are dominant; with MPC action accounting for uncertainties and tighter tracking, (when u_{ref} was removed designs they became very sensitive). The QPC solves a QP problem, rather than MIQP and was much faster to execute.

Evaluation on driving cycle data: To evaluate the various hybrid designs on a different realistic scenario, a fragment of an FTP18 driving cycle was used, again with varying engine speed as well as torque demand. The results are summarized in Table 2. The HPC design gives the best combination of torque tracking performance and fuel consumption.

Table 2 Controller performance metrics incl. relative simulation run time (evaluated on a fragment of the FTP18 driving cycle)

Controller	mse(TQ_{err})	max($\lvert TQ_{err}\rvert$)	sum(CAC_{eng})	Run time
MPC (n = 4)	10.3	29.0	4430	1.0
nGS-MPC	19.4	37.7	3857	1.0
HPC	11.5	24.3	3820	25.5
QPC	13.4	24.3	3804	4.5

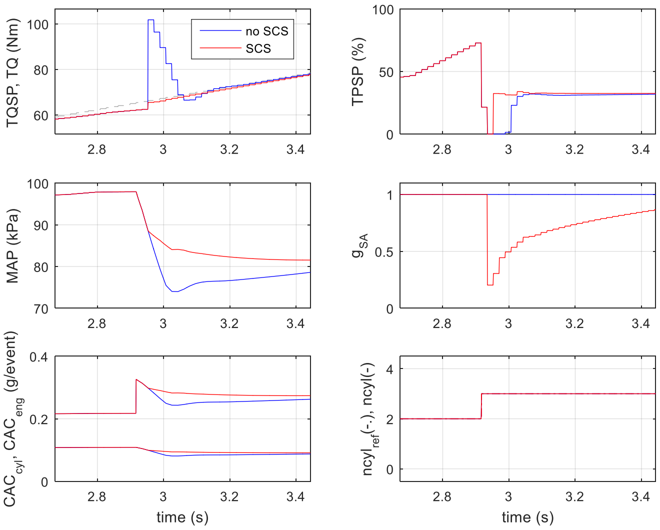

Fig. 8a Cylinder activation and deactivation using a simple TQSP ramp profile scenario: nGS-MPC control with and without SCS (detail view of the cylinder up-step)

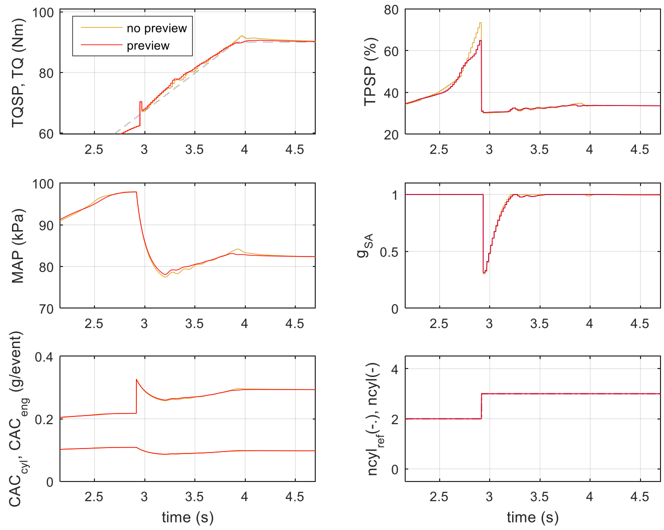

Fig. 9a HPC results with and without preview information for a simple TQSP ramp profile scenario (cylinder up-step)

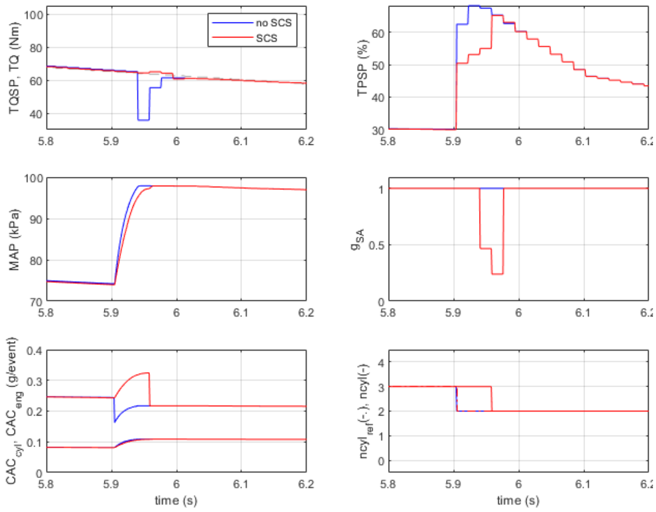

Fig. 8b Cylinder activation and deactivation using a simple TQSP ramp profile scenario: nGS-MPC control with and without SCS (detail view of the cylinder down-step)

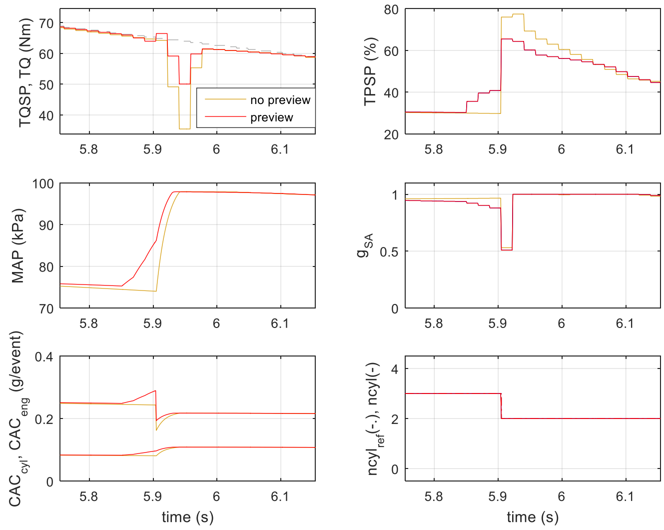

Fig. 9b HPC results with and without preview information for a simple TQSP ramp profile scenario (cylinder down-step)

The nominal full-cylinder MPC gives lowest TQ mean square error, due to the absence of switching, however, it does not improve fuel economy. The simple baseline nGS-MPC controller, using ad-hoc spark compensation, gives a reasonable trade-off between torque tracking and fuel economy. Finally, the QPC solution metrics are close to the optimal HPC, motivating the choice of this computationally cheaper algorithm for this problem. Fig. 12 compares the full-cylinder MPC control (with fixed $n = 4$) and the HPC solution. The effect of the latter on MAP is clearly visible, which leads to lower losses and improved fuel economy.

7. CONCLUSIONS

The aim was to understand the potential of hybrid control methods in applications. The VDE engine study has provided valuable experience on different aspects of the hybrid control systems design and tuning problem. In practice, hybrid control normally involves a two-layer supervisory structure.

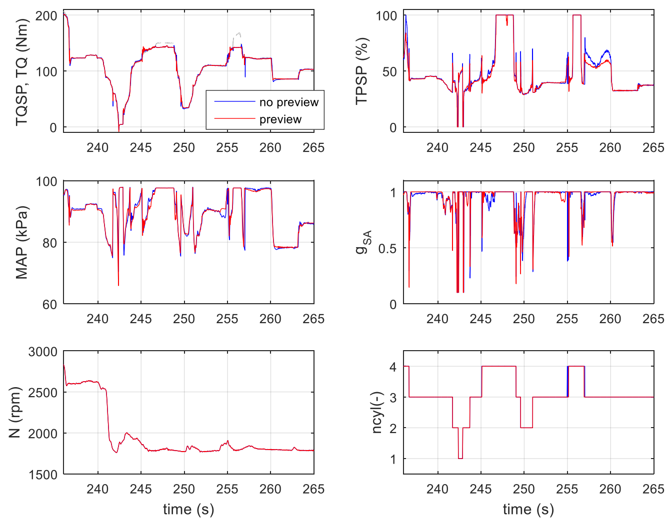

Fig. 10 HPC control, less aggressive fragment of US06 driving cycle: without preview and with preview

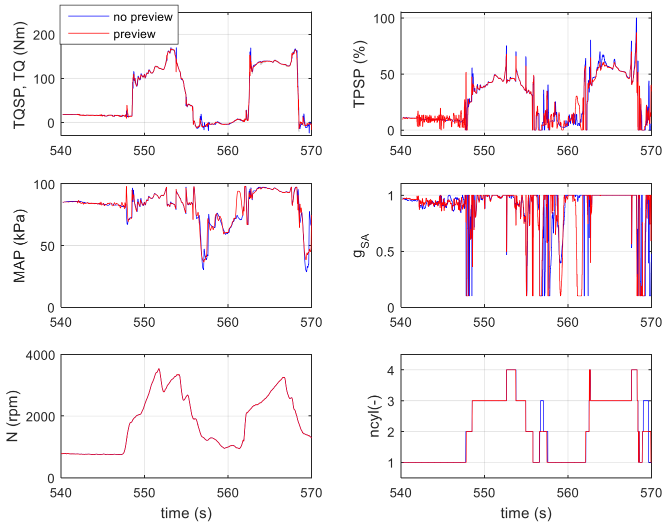

Fig. 11 HPC control, more aggressive fragment of US06 driving cycle: without preview and with preview

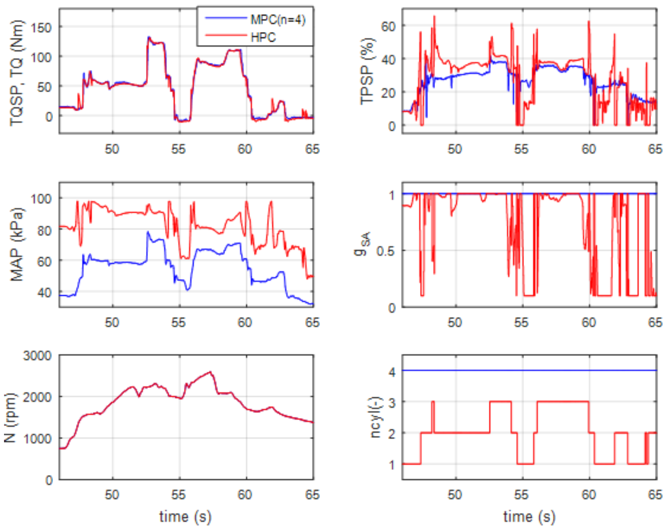

Fig. 12 Comparison of full-cylinder MPC and the HPC solution on a fragment of the FTP18 driving cycle

The switching is handled in an upper layer (determined by a static map), with continuous control in the lower layer. Since the changes of the discrete variables are essentially step disturbances for the lower control layer, additional ad-hoc compensation schemes will normally be required to reduce the associated transients, as illustrated in the baseline solution.

If the switching affects critical behavior, such as the fuel consumption, the switching can be optimized. A model-based control solution understands the cross-coupling effects and transients due to switching when minimizing the criterion. This approach should therefore provide a more integrated and optimized solution, and provide a unified framework for systems with discrete/logic and continuous regulation.

Although hybrid control theory is complex mathematically and algorithmically, this does not affect the usability. However, the standard optimal control solution did not provide effective compensation for switching and some tailoring of the algorithms was needed. A criterion that aims to limit mean square errors may not cope with spikes in transient responses. The hybrid solution can optimize the total system rather than subsystems enabling trade-offs to be made and designs to be obtained quickly. It was easy to enhance the basic hybrid MPC control algorithm to include dynamic cost-function weightings on the error and control terms and for these to be scheduled. "Control reference" values (u_{ref}), were obtained from static lookup tables and considered a form of 'feedforward' that was essential for consistent and stable control.

Tracking performance improved with preview action that was needed in the case of sudden (stepwise) reference changes. The 'advanced' form of preview control, involving modifications to the LPV model dynamics over the prediction horizon, also gave improved performance. These beneficial effects were more pronounced for aggressive driving styles.

The main approach to simplifying the HPC control problem involved the use of quantization. Rather unexpectedly, the results obtained using this method were similar to the full optimal solution. The theoretically optimal MIQP solution may not therefore be necessary for all HPC types of problem.

REFERENCES

Bemporad, A., Bernardini, D., Long, R., and Verdejo, J., (2018). Model Predictive Control of Turbocharged Gasoline Engines for Mass Production, *SAE 01-0875.*

Bemporad, A., Bernardini, D., Livshiz, M., and Pattipati, B., (2018). Supervisory model predictive control of a powertrain with a continuously variable transmission. *SAE 01-0860.*

Bonami, P. et al., (2008). An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs, *Discrete Optimization*, pp. 186–204.

Borrelli, F., Bemporad, A., Morari, M., (2017). *Predictive Control for Linear and Hybrid Systems,* CUP, Cambridge.

Camacho, E. F., Bordons, C., (2013). *Model predictive control.* Springer Science and Business Media.

Giorgetti, N., Bemporad, A., Tseng, H. E., Hrovat D., (2006). Hybrid model predictive control application towards optimal semi-active suspension, *Int J of Cont*, pp. 521-533.

Lazar, M., Heemels, W., Weiland, S., and Bemporad, A., (2006), Stabilizing Model Predictive Control of Hybrid Systems. *IEEE Trans. on Aut. Control*, pp. 1813-1818.

Lunze, J., Lamnabhi-Lagarrigue, F. (Eds.)., (2009). *Handbook of Hybrid Systems Control,* CUP.

Majecki, P., Gerrit, M., Grimble, M. J., Haskara, I., Hu, Y. and Chang, C.-F. , (2015). Real-Time Predictive Control for SI Engines Using Linear Parameter-Varying Models. *5th IFAC Conf. on NMPC*, Seville, Vol. 48, pp. 94-101.

Majecki, P., Grimble, M. J., Haskara, I., Hu, Y. and Chang, C.-F. , (2017). Total engine optimization and control for SI engines using linear parameter-varying models, *American Control Conference (ACC)*, Seattle, WA, pp. 3631-3637.

Salmah, S., Joelianto, E., Budiyono, A., Wijayanti, I. E., Megawati, N. Y., (2013). Tracking control for hybrid system of unmanned small scale helicopter using predictive control. *IEEE Int. Conference on Robotics, Biomimetics, and Intelligent Computational Systems.*

Toth, R., (2010). Modeling and identification of linear parameter-varying systems (Lecture notes in control and information sciences; Vol. 403), Springer Berlin.