Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

of the 18$^{\text{th}}$ International Conference on Process Control

Hotel Titris, Tatranská Lomnica, Slovakia, June 14 – 17, 2011

ISBN 978-80-227-3517-9

http://www.kirp.chtf.stuba.sk/pc11

Editors: M. Fikar and M. Kvasnica

Behrendt, S., Dünow, P., Lampe, B.P.: An Application of Model Predictive Control to a Gasoline Engine, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, 57–63, 2011.

Full paper online: http://www.kirp.chtf.stuba.sk/pc11/data/abstracts/033.html

An Application of Model Predictive Control to a Gasoline Engine

Stefan Behrendt* Peter Dünow* Bernhard P. Lampe**

* *Hochschule Wismar, University of Applied Sciences, Faculty of Engineering, Germany (Tel: +49 (0)3841 753 154; e-mail: stefan.behrendt@hs-wismar.de)*
** *University of Rostock, Faculty of Computer Science and Electrical Engineering, Institute of Automation, Germany (e-mail: bernhard.lampe@uni-rostock.de)*

Abstract: The coordination of available actuators in modern gasoline engines is a challenging task. An available scheme for efficient coordination that respects the actuator constraints is model predictive control, but a specialised implementation of the incorporated optimisation algorithm is necessary to cope with the timing requirements. The numerical efficiency of the developed algorithm and the performance of the realised torque and speed control are presented in simulation and real-time in a Volkswagen T5 transporter respectively.

Keywords: Engine control, Model Predictive Control, Torque Coordination

1. INTRODUCTION

The number of the actuators in modern gasoline engines increases due to rising requirements regarding emissions and fuel consumption. The coordination of the available actuators (e.g. throttle, ignition plug, exhaust gas recirculation and turbo charging) is a challenging task. The electronic control unit (ECU) needs to calculate appropriate signals, which are hardly optimal because of the disregarded coupling within the actual structure, the non-linear characteristics and limitations on the actuator signals. These computed signals are mostly based on heuristics and look-up tables that are to a large extent chosen manually. This leads to high expenses and a later time-to-market.

A possibility to arrange the available torque sources in a systematic way is model predictive control (MPC). By superordinating this controller an efficient coordination of the subsidiary torque control structures is possible, due to the predictive nature of the controller and inherent consideration of constrained actuators.

Over the decades the MPC concept has proved successfully in controlling plants with complex dynamics. Due to its high computational complexity its usage is being limited to plants with slow dynamics like in the process industry. Numerous companies developed reliable software for process automation systems (PAS) (e.g. Aspen Technology, Inc. (2010)) and programmable logic controllers (PLC) (e.g. Siemens AG (2008)).

In contrast, controlling plants with fast dynamics still poses a problem. The fast sampling times necessary are perceived to prevent the MPC of these plants on standard embedded systems, beside the interest in this control approach is growing in the automotive industry and the benefits are already proven by e.g. Saerens et al. (2008) and del Re et al. (2010). To address this limitation several techniques have been developed to enlarge the field of MPC to embedded systems with small sampling times.

An implementation with sufficient worst case timing for the MPC of a single-input single-output (SISO) system is presented by Wills et al. (2008). It suggests the enhancement of embedded systems by a digital signal co-processor (DSP) for the fast evaluation of the underlying algorithms.

The implementation strategies and the actual implementation on field programmable gate array (FPGA) chips are presented by Knagge et al. (2009) and Ling et al. (2006) respectively. While the former address the specific architecture, like parallelism, explicitly and therefore the greater improvement is expected, the effort of implementing the necessary algorithms in a hardware description language should not be underestimated. To compete against high-potential micro-controllers in terms of computational time is a challenging task.

Another approach is the combination of on-line optimisation and a partial enumeration method presented by Pannocchia et al. (2006). The solutions of the optimisation problem with active constraint sets that appear with highest frequency are computed off-line and stored in a table. This table is searched on-line for the best control. In case, that meanwhile the on-line computation an active constraint set does not exist in the table, it is adapted. With this method a significant speed-up is possible. The drawbacks are performance degradation and the memory requirements.

The explicit MPC has gained much attention in the recent years. Therefore the state space is partitioned into polyhedral regions. The control law is formulated as a function of the plant state and the piecewise linear solutions to the control problem with respect to the constraints are calculated as described by Bemporad et al. (2002). The on-line computational complexity reduces to the selection of the appropriate control law depending on the actual state. Numerous successful applications followed by e.g. Ortner et al. (2006), Naus et al. (2008) and Arce et al. (2009). On the other hand, explicit MPC is limited to plants with a small number of inputs and short control horizons (for an explanation of this term see section 2) as the number of regions grows exponentially with these parameters. A prohibitively large amount of memory would be necessary, which is addressed e.g. by Rossiter and Grieder (2004). Additionally the control parameters and the constraints of the actuating variables are commonly fixed, which is undesired in some control problems. This limitation is addressed by Baric et al. (2005), but results in additional optimization variables and therefore a larger demand on memory. Hence, it quickly exceeds available resources in practical situations.

With the advent of multi-core controllers for embedded systems (e.g. XMOS Ltd. (2010)) and multi-core DSPs (e.g. Texas Instruments Inc. (2010)) it seems reasonable that parallel algorithms for MPC could lower the computational burden. To the knowledge of the author the main developments occur in the field of large-scale and sparse problems (e.g. Gondzio and Grothey (2006) and Gondzio and Grothey (2007)) and focus on the parallelisation of incorporated operations (e.g. matrix multiplication and inversion) as by Ruano and Daniel (1997). In a recent publication by Behrendt et al. (2010) a parallelisation approach on the optimisation level is presented that allows for super-linear speed-up of the computation at time-critical sampling instances.

The purpose of this paper is to show that the MPC of gasoline engines on a commonly used ECU is already feasible by utilising a specifically tailored algorithm and a limited number of actuators. This conclusion is evident by simulative results and by measurements in a Volkswagen T5 transporter with a 2l TFSI (Turbo Fuel Stratified Injection) engine.

The paper is organised as follows. The basics of MPC and the features of the applied algorithm for the solution of the incorporated optimisation are presented in section 2. The section 3 explains the utilised model structure. In section 4 the numerical properties are obtained by a Hardware-In-The-Loop simulation and the practical relevance is proven by controlling the engine of the Volkswagen T5.

2. MODEL PREDICTIVE CONTROL BASICS

In this section a short introduction to the MPC fundamentals is presented. For a more comprehensive survey on the theory of MPC the reader is referred to Maciejowski (2002) and Camancho and Bordons (2004).

The MPC method combines the advantages of predicting the behaviour of the plant, namely the output, and respects constraints on the actuators. Therefore the cost function

$$J(k) = \sum_{i=1}^{H_p} \|\hat{y}(k+i|k) - w(k+i|k)\|^2_{Q(i)} + \sum_{i=0}^{H_u-1} \|\Delta u(k+i|k)\|^2_{R(i)} \tag{1}$$

with the prediction horizon H_p, the control horizon H_u, the weights on the control deviation Q, the weights on the rate of change of the difference control action R, the predicted output $\hat{y}$, the reference w and the difference control action Δu needs to get minimized with respect to Δu.

The prediction follows from the state equations of the discrete-time linear plant

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k) \tag{2}$$

$$y(k) = \mathbf{C}x(k) \tag{3}$$

with the states $x \in \Re^{n_\mathrm{x}}$, the input $u \in \Re^{n_\mathrm{u}}$ and the output $y \in \Re^{n_\mathrm{y}}$ by

$$\hat{y}(k) = \Psi\hat{x}(k) + \Upsilon u(k-1) + \Theta\Delta U(k)\,. \tag{4}$$

Thereby Ψ, Υ, Θ and $\Delta U(k)$ are in the notation as presented by Maciejowski (2002). With the reference signal $w(k+i|k)$, $i = 1, \ldots, n_\mathrm{y}H_\mathrm{p}$, the control difference over the prediction horizon

$$\varepsilon(k) = \begin{bmatrix} w(k+1|k) \\ w(k+2|k) \\ \vdots \\ w(k+H_p|k) \end{bmatrix} - \Psi\hat{x}(k) - \Upsilon u(k-1) \tag{5}$$

leads to the cost functional

$$J(k) = \Delta U(k)^\mathrm{T} H \Delta U(k) - g(k)^\mathrm{T}\Delta U(k) \tag{6}$$

with

$$H = \Theta^\mathrm{T}\mathcal{Q}\Theta + \mathcal{R} \text{ and } g(k) = 2\Theta^\mathrm{T}\mathcal{Q}\varepsilon(k)\,. \tag{7}$$

By means of the diagonal matrices $\mathcal{Q} \geq 0$ und $\mathcal{R} > 0$, which elements consist of $Q(i)$ with $i = 1, \ldots, H_\mathrm{p}$ and $R(i)$ with $i = 0, \ldots, H_\mathrm{u} - 1$ respectively, the resulting control is parametrised.

Additionally constraints on the actuating variables are defined by the linear matrix inequality

$$A\Delta U(k) \leq b(k)\,. \tag{8}$$

The minimization of the cost function (6) subject to the constraints (8)

$$\min_{\Delta U(k)} \left\{\Delta U(k)^\mathrm{T} H \Delta U(k) - g(k)^\mathrm{T}\Delta U(k) \;:\; A\Delta U(k) \leq b(k)\right\} \tag{9}$$

defines a mathematical standard problem that can be solved by quadratic programming (QP). A variety of methods for solving the QP are commonly used. For an overview we refer to Nocedal and Wright (2006). The basis of the herein used algorithm is the active-set method described by Fletcher (1981), but is optimised for the solution of the QP within the MPC algorithm. Therefore several enhancements are applied like

tuning for the specific structure of the constraint The applied algorithm only accounts for constraints on the actuating variables. Therefore the matrix product $A\Delta U(k)$ equals an accumulated sum of the elements of $\Delta U(k)$ that belongs to the same actuating signal. Therefore a decreases number of operations in necessary.

restriction to a certain process class The actual implementation requires a certain process class that occurs in engine control as shown by Fritzsche et al. (2009). This specialisation allows for the on-line adjustment of the process model with a reasonable demand on processing time.

enhanced warm-starting feature The usage of the solution of the QP at the previous time instant for initialisation often decreases the processing time significantly. In case of reference changes or large disturbances this approach can have the opposite effect. An enhanced initialisation routine based on the procedure by Hildreth (1957) is capable of avoiding such conditions (see Behrendt et al. (2010)).

pre-computing the system of equations In each iteration of the active-set method a system of equations needs to get solved that varies in size depending on the number of active constraints. The application of a LDL^{T}-decomposition allows to pre-compute this system of equations to some extent. Behrendt (2009) has shown that this leads to a significantly lowered numerical complexity for the online iterations.

multiple activation and deactivation of constraints The active constraints in the solution are identified sequentially in the active-set method. Because of the independence of $\Delta U(k)$ terms in (8) that are associated to different actuating signals a concurrent activation of constraints is viable. This often leads to a decreased number of iterations and improves the result in case of an early interruption of the algorithm.

exploit the influence of active constraints Active constraints correspond to optimisation variables that are fixed. Hence, the computed step towards the minimum of (6) is equal to zero. Due to this a priori knowledge the calculation of the step is avoided and a decreased number of operations has to be performed.

These enhancements limit the region of attraction, but allow for real-time control of a certain process class.

3. A SPECIAL CLASS OF SYSTEMS FOR ENGINE CONTROL

This paper deals with a special class of systems that consists of a number of main control variables Y_i, $i = 1, \dots, m$ (e.g. torque and engine speed), a main actuating variable U_1 (e.g. torque by air path) and a number of auxiliary actuating variables U_i, $i = 2, \dots, n$ (e.g. torque by ignition path, exhaust gas recirculation and electric engine). The general system can be described by the discrete-time transfer function

$$\begin{bmatrix} Y_1(z) \\ \vdots \\ Y_m(z) \\ A_1(z) \\ \vdots \\ A_n(z) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) & \cdots & G_{1n}(z) \\ \vdots & \ddots & \ddots & \vdots \\ G_{m1}(z) & G_{m2}(z) & \cdots & G_{mn}(z) \\ 0 & G_{\mathrm{aux}}(z) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{\mathrm{aux}}(z) \end{bmatrix} \begin{bmatrix} U_1(z) \\ U_2(z) \\ \vdots \\ U_n(z) \end{bmatrix} . \tag{10}$$

The purpose of the auxiliary variables is to dynamically support the main actuating variable, but return to their references in the steady state. The transfer function G_{aux} in (10) represents the dynamic of the auxiliary variable realisation. In case, that the auxiliary variable is freely alterable within the constraints in every engine cycle the transfer function

$$G_{\mathrm{aux}}(z) = z^{-1} . \tag{11}$$

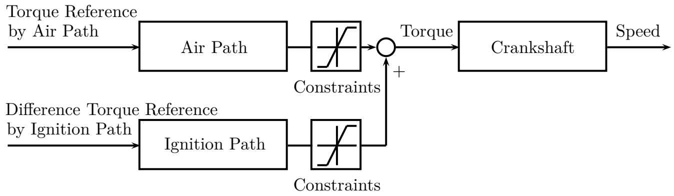

Fig. 1. Considered process structure

Therefore by controlling the variables $A_i, i = 1, \dots, n$ the auxiliary variables are controlled to a prescribed reference (e.g. optimal ignition angle) provided by the combustion process supervision. The relation between the main control variables and the main actuating variable described by $G_{i1}(z)$, $i = 1, \dots, m$ is of slower dynamic in comparison to the remaining sub-systems. This encourages the evident active aid by the auxiliary variables for controlling the plant.

The considered process structure for the specific purpose of this paper is shown in Fig. 1. Two actuating variables have been chosen for engine control. Namely the torque by the air path (AP), which determines the amount of air in the cylinder by altering the throttle opening angle. The amount of air directly affects the resulting torque at the crankshaft. The difference torque by the ignition path (IP), which determines the ignition angle, serves as auxiliary variable and allows for an efficiency deterioration. Thus, it is possible to establish a torque reserve by means of the ignition angle. Because the time constant of the ignition path is significantly smaller than that of the air path a faster response to reference changes or disturbances is available. Additionally both actuating signals are subject to constraints which are inherently respected by MPC. These constraints result from safety reasons in case of the air path, because an exceeding torque could damage the engine. The ignition path is constrained to prevent knocking in the cylinder. For an explanation of this term we refer to Gupta (2006).

The control variables are chosen to be the generated torque at the crankshaft T and the resulting engine speed n. The resulting model can be written as

$$\begin{bmatrix} T(z) \\ n(z) \\ \Delta T\_IP(z) \end{bmatrix} = \begin{bmatrix} G_{T\_AP}(z) & G_{T\_IP}(z) \\ G_{n\_AP}(z) & G_{n\_IP}(z) \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} T\_AP(z) \\ \Delta T\_IP(z) \end{bmatrix} . \tag{12}$$

Naturally the torque and speed cannot be controlled independently. The controller may take only one control variable at a time into account. This can be achieved by not considering the control offset of the particular process output by setting the associated element in the weight of the control deviation q to zero. Otherwise the MPC would attempt to track both references that would lead to an offset on both control signals.

Hence, a single controller structure grants to track different references in dependency on the choice of one parameter. In the field of automotive control this would allow to join the torque controller that tracks the commands of the driver by the gas pedal and the idle speed controller. This approach would significantly simplify the control structure within nowadays ECUs.

Table 1. Control parameters and implementation details

Parameter	Value
prediction horizon H_p	20
control horizon H_u	3
control deviation weight q	[0.1 0.004 0.05]
control action weight r	[10 1]
min. control variable	[10 -10]
max. control variable	[55 0]
processor clock rate	150 MHz
code memory (without FP libs)	6.3 KB
code memory (with FP libs)	13.2 KB
data memory	2.0 KB

It should be mentioned again that we are not going to influence the throttle or the advance angle directly. Instead the MPC is used as superordinate controller that supplies optimal references for the air and ignition path. These references are tracked by the subsidiary control structures that are already included in the ECU. This approach leads to some extent to a linearisation of the underlying structures and therefore a linear MPC can be applied.

4. RESULTS

The results presented in this section are based on the model (12) derived in the previous section. The parameters are obtained by experimental identification at a Volkswagen T5 transporter with 2l TFSI engine.

4.1 Hardware-In-The-Loop Simulation

By means of a Hardware-In-The-Loop (HiL) simulation on the target platform Tricore TC1796 by Infineon Technologies AG (2010) we will show the performance of the control and the algorithm. The TC1796 is a high-potential microcontroller that meets the demands of nowadays engine control. It incorporates an effective floating point unit (FPU), a digital signal processor (DSP) with fixed point arithmetic and a peripheral controller (PCP).

The experimentally chosen control parameters for MPC are summarised in Table 1 and the resulting control is

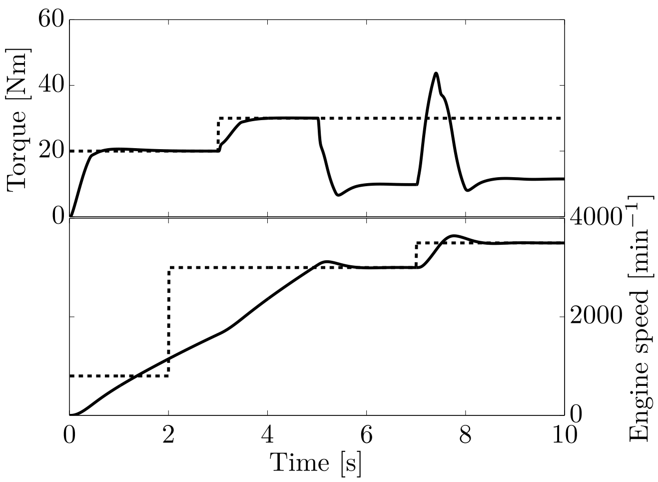

Fig. 2. Control signals (solid) and the references (dashed)

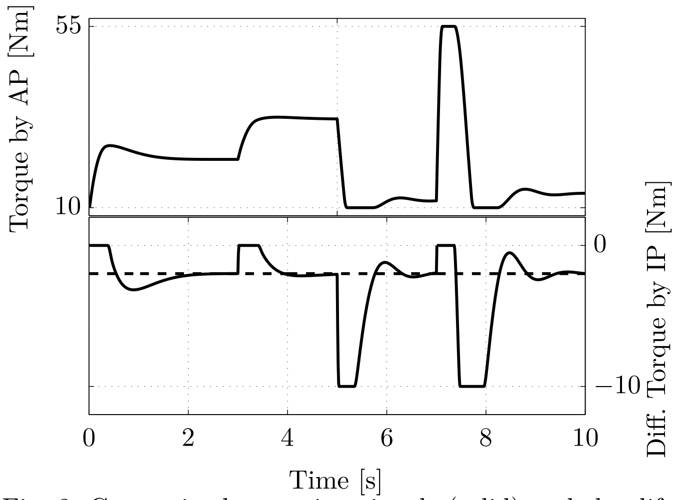

Fig. 3. Constrained actuating signals (solid) and the difference torque reference by IP (dashed)

shown in Fig. 2. Until $t = 5\,\mathrm{s}$ the torque is controlled to follow the reference. Therefore the weight on the speed control deviation q_2 is set to zero. At $t = 5\,\mathrm{s}$ the control objective changes by setting the weight on the speed control deviation according to Table 1, but the weight on the torque control deviation q_1 is set to zero. This modifies the MPC to track the speed reference until the end of the simulation.

The difference torque by the ignition path (IP) actively supports the torque by the air path (AP) to reduce the control deviation in the transient phases as shown in Fig. 3. As the control deviation tends to zero the difference torque by IP returns to its reference of $-2\,\mathrm{Nm}$. This realises a torque reserve to allow for quickly reaction on reference changes and disturbances. Nevertheless the constraints on the actuating signals according to Table 1 are respected at any time.

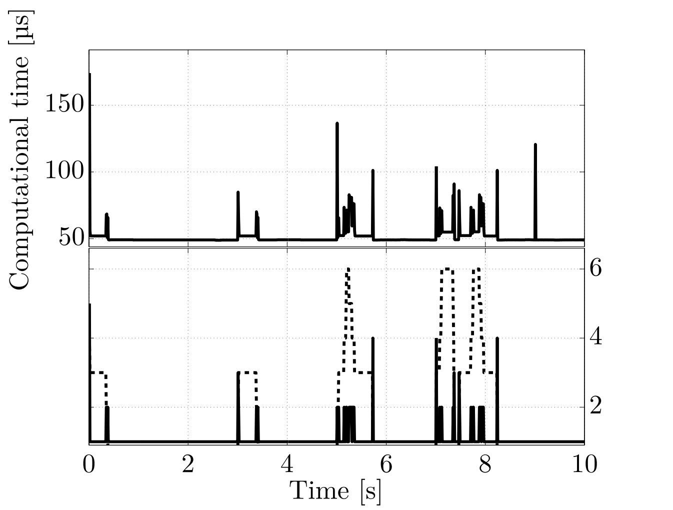

Fig. 4. The computational time for calculating the appropriate actuating signals is shown in the upper graph and the lower graph shows the number of iterations of the active-set algorithm (solid) and the number of active constraints (dashed)

According to the intended operation in an ECU the sample time is set to 10 ms, because that is the fundamental sample time for the air path functionalities. Fig. 4 reveals that the necessary computational time is sufficient for real-time computation of the MPC. The maximum execution time is approximately 180 µs at the beginning of the simulation that leaves enough resources for the remaining functions in the ECU. At the time $t = 5$ s the control objective is adapted by changing the weights and at time $t = 9$ s the plant parameters are slightly changed to demonstrate the effect on the computational time. Both changes cause the re-computation of parameters of the QP, but show moderately increased computational time. The lower graph of Fig. 4 shows the benefit of warm-starting the optimisation algorithm with the solution of the previous time instant. The number of iterations equals the number of necessary changes to the identified active constraints, but not to the number of active constraints itself.

4.2 Evaluation in a Volkswagen T5 transporter

The internal bypass concept by Accurate Technologies (2011) facilitates the easy insertion of additional functionality into existing software states without changing software source code. This allows for expanding an ECU for engine control by the MPC algorithm. The algorithm bypasses the ECU functions that would calculate the torque by AP and the torque by IP in a production ECU. By means of this technique the evaluation of the MPC on the intended hardware platform provides an insight into the applicability of the entire approach.

The first measurements occurred in the idle running of the engine. The Fig. 5 shows the controlled torque (green) and the torque reference (red) that is subject to changes. In case of a reference change the difference torque by IP (magenta) supports the torque by AP (blue) to reduce the control deviation e.g. at the time $t = 452$ s, but does not exceed the minimal difference torque of -10 Nm. In the steady state it returns to its reference (cyan) again. The changes to the difference torque reference by IP are stationary compensated by means of the torque by AP and show minor influence on the controlled torque itself e.g. at the time $t = 477$ s.

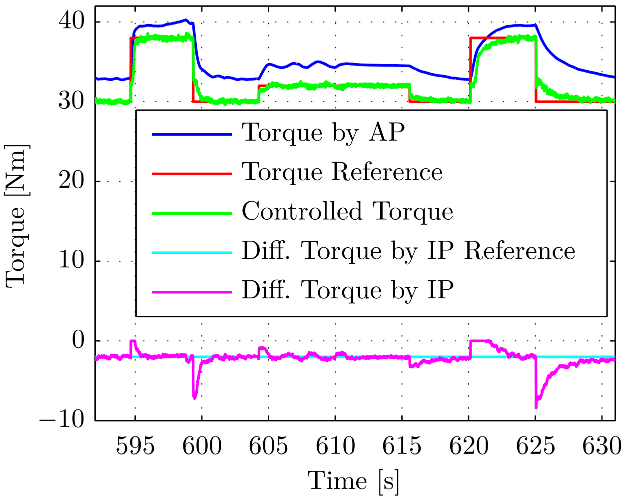

Fig. 6. Torque control at the vehicle in idle running subject to changes of control action weight of torque by AP

The applied MPC algorithm provides the opportunity to change the control parameters during operation. In Fig. 6 an adjustment of the control action weight for the torque by AP takes place. At the time $t = 594$ s the AP is manipulated dynamically to quickly reduce the control deviation due to the reference change. The difference torque by IP does not exceed the maximal difference torque of 0 Nm and immediately returns to its reference. In contrast at the time $t = 620$ s the weight is adapted to moderately utilise the AP. Consequently the torque control deviation and the deviation of the difference torque by IP persists for a longer period of time. This can be interpreted as a sport and economy mode respectively as it is available in nowadays vehicles.

Fig. 7 reveals the possibility of changing the control objective. Until the time $t = 128$ s the torque (green) is

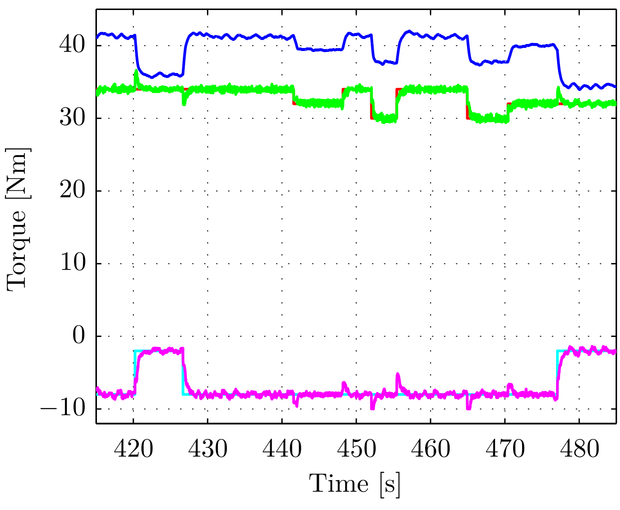

Fig. 5. Torque control at the vehicle in idle running subject to changes of the difference torque reference by IP

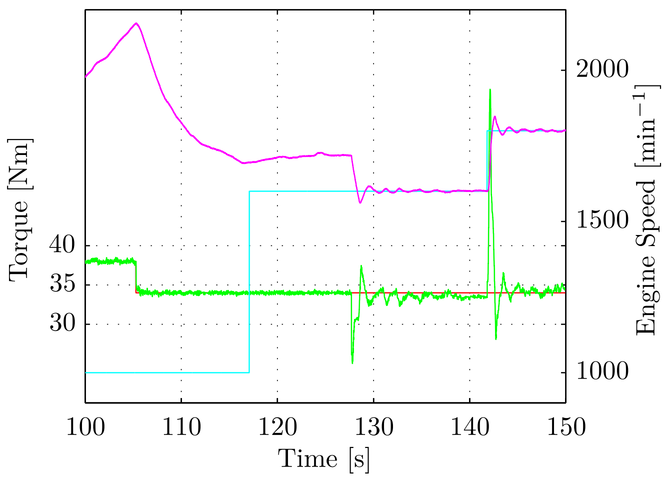

Fig. 7. Online change of the control objective from torque control to engine speed control at the vehicle in idle running

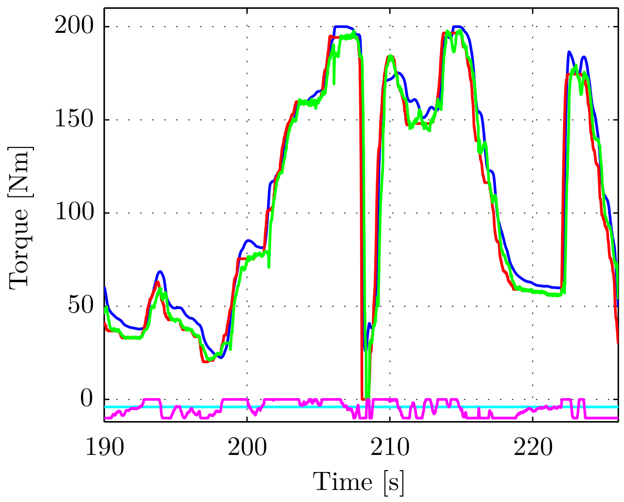

Fig. 8. Torque control at the vehicle on a test track

controlled to its reference (red). Afterwards the control objective is adapted to force the engine speed (magenta) to its reference (cyan).

The evaluation of the MPC on a test track proves the ability to dynamically control a driving scenario. The torque reference in Fig. 8 is shaped by the driver by means of the gas pedal. The actuating variables are manipulated accordingly. Because of seldom steady state phases the difference torque by IP almost always differs from its reference, but never exceeds the constraints. The figure shows, that the torque by AP also remains below the safety constraint of 200 Nm. The vehicle accelerated from 20 km h^{-1} to 66 km h^{-1} and at the time $t = 209$ s the gear is changed. Nevertheless, an adaptation of the process model is not necessary to cope with the changes in the process behaviour.

5. CONCLUSION

While the attractive features of MPC are especially interesting in the automotive industry for e.g. engine control, the numerical complexity of the incorporated optimisation is perceived to prevent its application in embedded systems like production ECUs. This concern has motivated this paper and it could be shown that model predictive engine control is feasible by utilising a specifically tailored algorithm and a limitation to two actuators. A worst case timing of approximately 180 µs during a sample control scenario by using a common micro-controller leaves the necessary resources for the remaining functions in the ECU. Further studies have unveiled that a worst case timing of 800 µs results if four actuators are utilised. That could be recognised of being acceptable as well.

The evaluation in the vehicle has pointed the practicability out. The actuating signals are manipulated in an efficient manner and the ability to control the torque and the engine speed by a single MPC could significantly simplify the existing control structure in nowadays ECUs. Nevertheless, the approach is not yet usable in production units. More work is necessary to account for e.g. overrun fuel cut-off, cylinder shut-off and the integration in the existing structure.

ACKNOWLEDGEMENT

The authors would like to thank Malte Köller and Christian Steinbrecher at IAV GmbH for the technical and organisational support. Their insights in the ECU and valuable help were crucial to the successful evaluation in the vehicle.

REFERENCES

Accurate Technologies (2011). No-Hooks Software. URL `http://www.accuratetechnologies.com/en/products/no-hooks-software-246-243.html`.

Arce, A., del Real, A.J., Bordons, C., and Ramirez, D.R. (2009). Real-Time Implementation of a Constrained MPC for Efficient Airflow Control in a PEM Fuel Cell. IEEE Transactions on Industrial Electronics.

Aspen Technology, Inc. (2010). aspenONE - Advanced Process Control. http://www.aspentech.com.

Baric, M., Baotic, M., and Morari, M. (2005). On-line tuning of the explicit solution to model predictive control. Technical Report AUT05-10, ETH Zurich, Automatic Control Laboratory. URL `http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=2218`.

Behrendt, S. (2009). *Echtzeitfähige Modellprädiktive Regelung für Verbrennungskraftmaschinen*. Master's thesis, Hochschule Wismar.

Behrendt, S., Dünow, P., and Lampe, B. (2010). Simulation study of parallel model predictive control. In *Proceeding of 7th EUROSIM Congress on Modelling and Simulation*. Prague, Czech Republic.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. In *Automatica*, volume 38, 3 – 20.

Camancho, E. and Bordons, C. (2004). *Model Predictive Control*. Springer.

del Re, L., Allgöwer, F., Glielmo, L., and Guardiola, C. (2010). *Automotive Model Predictive Control*. Springer.

Fletcher, R. (1981). *Practical Methods of Optimization*, volume 2: Constrained Optimization. John Wiley & Sons.

Fritzsche, C., Dünow, P., Behrendt, S., Seemann, P., Schnaubelt, M., and Schultalbers, M. (2009). Predictive speed and torque control. In *Proceedings of 7. Symposium "Steuerungssysteme für den Antriebsstrang"*. Berlin, Germany.

Gondzio, J. and Grothey, A. (2006). *Computational Finance and its Applications II*, chapter Solving Nonlinear Financial Planning Problems with 10^9 Decision Variables on Massively Parallel Architectures. WIT Press.

Gondzio, J. and Grothey, A. (2007). Parallel interior point solver for structured quadratic programs: Application to financial planning problems. In *Annals of Operations Research*, volume 152, 319 – 339.

Gupta, H. (2006). *Fundamentals of Internal Combustion Engines*. Prentice Hall.

Hildreth, C. (1957). A quadratic programming procedure. In *Naval Research Logistics Quarterly*, volume 4, 79 – 85.

Infineon Technologies AG (2010). TC1796 (AudoNextGeneration). http://www.infineon.com.

Knagge, G., Wills, A., Mills, A., and Ninness, B. (2009). ASIC and FPGA Implementation Strategies for Model

Predictive Control. In *European Control Conference (ECC)*.

Ling, K., Yue, S., and Maciejowski, J. (2006). A FPGA implementation of model predictive control. In *Proceedings of the American Control Conference*.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Pearson Education Limited.

Naus, G., van den Bleek, R., Ploeg, J., Scheepers, B., van de Molengraft, R., and Steinbuch, M. (2008). Explicit MPC design and performance evaluation of an ACC Step-&-Go. In *Proceedings of the 2008 American Control Conference*.

Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. Springer, second edition.

Ortner, P., Langthaler, P., Ortiz, J.V.G., and del Re, L. (2006). MPC for a Diesel Engine Air Path using an Explicit Approach for Constraint Systems. In *Proceedings of the 2006 IEEE International Conference on Control Applications*.

Pannocchia, G., Rawlings, J.B., and Wright, S.J. (2006). The partial enumeration method for model predictive control: Algorithm and examples. Technical Report 1, TWMCC – Texas-Wisconsin Modeling and Control Consortium.

Rossiter, J. and Grieder, P. (2004). Using interpolation to simplify explicit model predictive control. In *Proceeding of the 2004 American Control Conference*, 885 – 890.

Ruano, A. and Daniel, H. (1997). Parallel implementation of an adaptive generalized predictive control algorithm. In *European Control Conference*. Brussels.

Saerens, B., Diehl, M., Swevers, J., and den Bulck, E.V. (2008). Model Predictive Control of Automotive Powertrains - First Experimental Results. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 5692 –5697.

Siemens AG (2008). SIMATIC PCS 7 APC-Portfolio. http://www.automation.siemens.com.

Texas Instruments Inc. (2010). TMS320C647x Multicore DSPs. http://www.ti.com.

Wills, A., Bates, D., Fleming, A., Ninness, B., and Moheimani, S.R. (2008). Model predictive control applied to constraint handling in active noise and vibration control. *IEEE Transactions on Control Systems Technology*, 16(1), 3–12.

XMOS Ltd. (2010). XS1-G4: 4-core processor. http://www.xmos.com.