

Numerically Efficient Self Tuning Strategies for MPC of Integral Gas Engines

Harald Waschl* Daniel Alberer** Luigi del Re**

* *Hoerbiger Research Institute for Smart Actuators*

** *Institute for Design and Control of Mechatronical Systems at the
Johannes Kepler University of Linz, Austria. (phone:
+43-732-2468-6210 e-mail:
harald.waschl,daniel.alberer,luigi.delre@jku.at)*

Abstract: Integral gas engines represent a critical element in the US gas pipeline network. The stricter emission requirements now facing also these engines have prompted the question for advanced control strategies. Among them MPC has turned out to be a promising choice. Unfortunately, MPC has two main drawbacks a high computational and a demanding and non intuitive tuning effort. These issue become even more critical if speed and precision of the hardware is limited, as it is often the case in embedded solutions. This paper extends previous work on the control of such systems and concentrates on the latter issue, the selftuning. It proposes a tuning approach based on the well-known equivalence between MPC and a set of linear controllers as well as on the use of a mixed cost function which explicitly includes the numerical condition of the MPC problem. As the paper also shows, this yields to an impressive increase in the efficiency and performance of the online algorithm by reducing the number of QP iterations necessary at each step.

Keywords: Predictive control, Self tuning control

1. INTRODUCTION

The natural gas distribution system of the United States is a grid of pipelines and compressor stations to compensate pressure losses and adjust the flow to the actual gas consumption of the end users. These stations are typically equipped with integral engines – with combustion cylinders and the compressor acting on the same crankshaft. Mainly, these are turbocharged two stroke combustion engines, which are fueled directly with natural gas from the pipeline, see Alberer et al. [2008].

Due to the very high installation cost, only a few changes occur during the lifetime of a compressor station. These engines were of course not designed for the upcoming emission levels. In particular NO_x emissions have to be reduced, these are strongly connected to transient changes of the engine. In this case advanced control methods are a sensible alternative besides a mechanical upgrade. The MIMO nature of the plant, the existence of hard bounds on the manipulated variables and soft bounds on the outputs speak for a model predictive approach. Moreover, load changes are known in advance and therefore can be taken into account in the control task. Limitations of predictive control are the extensive tuning process and the high computational effort, because at every time instant an optimization problem has to be solved. However, recent developments in available computational power and efficient online solvers Ferreau et al. [2008] cope with this limitation and extend the range of applications to fast realtime systems, see Ferreau et al. [2007]. An alternative method is to shift the computational effort from online to offline by use of an explicit MPC structure Bemporad et al. [2000]. In Al-

berer et al. [2008] explicit MIMO model predictive control was applied to the considered compressor station and the transient NO_x emissions could be reduced significantly. In a subsequent work the explicit MPC was replaced by a realtime online MPC using qpOASES Ångeby et al. [2009]. Although these works led to promising results, still the challenging task of the optimal tuning of MPC has not been addressed, even though this task is critical for the industrial success, as many different installations exist, each of whom requires a specific tuning to be performed by a technician without advanced control background.

The question, in itself, is not new. Indeed several tuning guidelines for MPC can be found in the literature (e.g. in Soeterboek and Toumodge [1992] or Maciejowski [2002]) as well as their application, e.g. in Fan [2003], a strategy for tuning of MPC for large scale two dimensional actuator systems, like paper machine cross directional processes, is presented. Furthermore in Fan and Stewart [2007] a patent application based on the strategy presented in Fan [2003] and focused on the automatic tuning of multi variable MPC were published. In Vega and Francisco [2007] norm based approaches for automatic MPC tuning are proposed in combination with a mixed multi objective optimization. An alternative automatic tuning strategy presented in Al-Ghazzawi et al. [2001] is based on a linear approximation of the closed loop output corresponding to the tuning parameters and according to predefined time domain performance specifications. In a recent work a different method is proposed, where the unconstrained closed loop behavior of the MPC was matched to a given desired controller Di Cairano and Bemporad [2010].

Unfortunately, all these methods have been developed for generic systems and do not take in account specific problems of embedded solutions, in particular the numerical and time aspects. The connection between both is that the the online solution is essentially an iterative solution and a better numerical condition can reduce the number of necessary iterations and therefore the sample time.

This work starts from the earlier work Ångeby et al. [2009] and extends it by a self tuning strategy which takes in account the time and numerical limitations of embedded systems. The main idea is to exploit the degrees of freedom available in the cost functions to maintain the system tracking performance while enhancing the numerical condition.

2. SYSTEM DESCRIPTION

The considered integral gas engine consists of a six cylinder two stroke turbocharged reciprocating gas engine with counterflow scavenging and a compressor with three cylinders (Clark TLA 6) of which the used simulation models were derived from measurements.

Typically these engines are designed to run in a narrow operating range, i.e. at almost constant engine speed and power. The two main control inputs for the engine are the injected fuel amount, with the according control signal GFC (governor fuel command) and the wastegate position of the turbocharger WG , which determines the boost pressure and thus the air supply to the combustion chambers (both signals are normalized from 0% to 100%). An additional input is the desired load which is characterized by the total clearance volume of the compressor. This input can be seen as a measured disturbance, because it cannot be used actively for the control task (only a slight delay of the command is allowed) and is predetermined by the pipeline conditions and the operation strategy of the engine grid.

Currently these engines are equipped with two SISO PID controllers, one to keep the engine speed n constant by the use of GFC , and a second one uses the turbocharger wastegate to control the fuel to air ratio ϕ and consequently the NO_x emissions too¹. To provide low NO_x emissions – no catalytic aftertreatment devices are applied – and a high fuel efficiency, the engine is usually operated at very lean mixtures. Therefore during steady state operation low emission levels can be achieved, however during load transients high excursion of ϕ arise and consequently of NO_x too. The load control is performed with discrete volumes (pockets) that can be added to the compressor clearance volume.

Against the standard SISO PIDs the basic idea of the actual approach is to take into account the interconnected system behavior as well as the a priori knowledge of a forthcoming load change. The used control scheme can be separated in four parts, the integral engine with the inputs GFC , WG and the compressor load torque T_c , a MPC to control n and ϕ based on load information, system states and given setpoints, a state observer and a load handler.

¹ For a certain engine and constant ignition timing a unique (non-linear) relation exists between NO_x and ϕ .

In this case Gray box modeling was used, because some parts of the model are based on physical principles and so it is possible to assume a fixed structure with unknown parameters. During identification these sub models are identified separately in two stages. First all static gains are determined and in a next step the dynamics are covered. After the identification of the single parts all components are merged to one linear discrete state space model

$$x_{k+1} = A_p x_k + B_p u_k \quad y_k = C_p x_k$$

$$\text{with } u_k = [GFC, T_c, WG]^T \quad \text{and } y_k = [n, \phi]^T,$$

where A_p , B_p and C_p are the according system matrices. To cope with steady state offsets the plant model is extended with an output error disturbance model and a Kalman filter was used to provide the state information for the MPC.

3. MPC FORMULATION

In order to focus on the uprising numerical problems and the influence of the available tuning parameters on these conditions, we shortly recapitulate the QP MPC formulation used for the online MPC in the following section. The main idea of MPC is to use a model of the system to predict the future outputs and find an optimal control sequence that meets the requirements concerning performance and constraints Camacho and Bordons [2004], Maciejowski [2002]. The optimization problem is formulated at every time instant by taking into account the system model, whereas a commonly used objective function is to minimize the squared tracking error from a reference trajectory and additionally penalizing the applied control effort. The determination of the control signal can be formulated as optimization problem, where the values of u during the control horizon n_{CH} have to be determined. The objective function is calculated over the whole prediction horizon n_{PH} , whereas the control is kept constant after the control horizon. Due to the multivariable system structure the setpoints are defined as $y_{ref} = \begin{bmatrix} y_n \\ y_\phi \end{bmatrix}$ and the control action is given by $u = \begin{bmatrix} u_{GFC} \\ u_{WG} \end{bmatrix}$. The objective function can be defined as

$$\min_u \frac{1}{2} \sum_{k=0}^{n_{PH}} (y_k - y_{ref,k})^T S (y_k - y_{ref,k}) + \Delta u_k^T T \Delta u_k \quad (1)$$

s.t.

$$\begin{aligned} u_k &= u_{k-1} + \Delta u_k \\ x_{k+1} &= A_p x_k + B_p u_k \\ y_k &= C_p x_k \\ \underline{u} &\leq u_k \leq \bar{u} & k = 0 \dots n_{CH} - 1 \\ \underline{\Delta u} &\leq \Delta u_k \leq \bar{\Delta u} & k = 0 \dots n_{CH} - 1 \\ \Delta u_k &= 0 & k = n_{CH} \dots n_{PH} \end{aligned}$$

where the tracking error is weighted by $S = \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix}$ and the actuator movement is penalized by $T = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}$ with weighting factors $\{S_1, S_2, T_1, T_2\} > 0$.

In the actual case of a linear state space representation, the MPC optimization problem (1) can be stated in the form of a standard QP. By use of the tracking error formulation $e_k = y_k - y_{ref}$ the problem can be written as

$$\min \frac{1}{2} \xi^T E^T \tilde{S} E \xi + \Psi^T \Delta U^T \tilde{T} \Delta U \Psi \quad (2)$$

with

$$E = \begin{bmatrix} C & 0 & 0 & \dots & -I \\ CA & CB & 0 & \dots & -I \\ CA^2 & CAB & CB & \dots & -I \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \quad \xi = \begin{bmatrix} x_0 \\ u_0 \\ \vdots \\ u_{CH-1} \\ y_{ref} \end{bmatrix}$$

$$\Delta U = \begin{bmatrix} -1 & 1 & \dots & 0 \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{bmatrix} \quad \Psi = \begin{bmatrix} u_{-1} \\ u_0 \\ \vdots \\ u_{CH-1} \end{bmatrix} \quad \begin{matrix} \tilde{S} = \text{diag}(S \dots S) \\ \tilde{T} = \text{diag}(T \dots T) \end{matrix}$$

The matrices $E^T \tilde{S} E$ and $\Delta U^T \tilde{T} \Delta U$ can be structured in dependency of the parts of ξ and Ψ . In (2) the block (1, 1) of $E^T \tilde{S} E$ is multiplied from both sides with x_0^T from the vector ξ and the block (1, 2) is multiplied with x_0^T from the left and $(u_0 \ u_1 \ \dots \ u_{CH-1})^T$ from the right side and so on.

$$\underbrace{\begin{bmatrix} \dim x \times \dim x & \dim x \times CH \dim u & \dim x \times \dim y \\ CH \dim u \times \dim x & CH \dim u \times CH \dim u & CH \dim u \times \dim y \\ \dim y \times \dim x & \dim y \times CH \dim u & \dim y \times \dim y \end{bmatrix}}_{E^T \tilde{S} E}$$

A similar partition scheme can be applied for $\Delta U^T \tilde{T} \Delta U$ where the blocks are multiplied with the elements of Ψ . Blocks, which are multiplied with the vector of optimization variables from the left and right, are collected in the Hessian

$$H = \left[E^T \tilde{S} E \right]_{(2,2)} + \left[\Delta U^T \tilde{T} \Delta U \right]_{(2,2)}. \quad (3)$$

All blocks that are multiplied only from the left with the vector of optimization variables are collected in the gradient matrix f

$$f = \left[\left[E^T \tilde{S} E \right]_{(1,2)}, \left[\Delta U^T \tilde{T} \Delta U \right]_{(1,2)}, \left[E^T \tilde{S} E \right]_{(3,2)} \right]^T \cdot \Theta, \quad (4)$$

where the feedback vector $\Theta = [x_k, u_{k-1}, y_{ref}]^T$ contains the actual state, the past input and the actual reference. The resulting QP can be formulated to

$$\begin{aligned} \min_u & \frac{1}{2} u^T H u + u^T f \\ \text{s.t.} & \quad lb_G \leq G u \leq ub_G \\ & \quad lb \leq u \leq ub. \end{aligned} \quad (5)$$

The Hessian H and constraint matrix G remain constant, only the gradient f and the constraint vector have to be updated at every time instant. lb_G and ub_G represent the input rate constraints and lb and ub are the absolute constraints on the input. The QP formulation of (5) can finally be used for the real time application of the MPC. Basically every realtime QP solver can be used for this task, in this case a fast and numerical efficient QP solver was applied (qpOASES Ferreau et al. [2008]). To solve the optimization problem (5) matrix inversion are needed and it is known that the accuracy of an inversion depends strongly on the condition number of the matrix for a given precision.

4. SELF TUNING STRATEGY

As mentioned the numerical condition, in particular the condition number of the Hessian ($\kappa(H)$) are important

for the accuracy of the solution. $\kappa(H)$ can be seen as an amplification factor for data uncertainties and roundoff errors, hence the smaller the condition number the more accurate results can be achieved. Moreover, when using an online solver, which uses a homotopy method (like Ferreau et al. [2008]) and several iterations can be necessary to solve the QP, the accuracy and therefore the condition number has an influence on the necessary number of iterations² and of course on the achievable sample time. In the considered application on the compressor station an embedded hardware system with single precision arithmetic is available. This configuration affects the relation between the condition number and the accuracy of the solver. Theoretically, the possible accuracy is the machine precision, which is about 10^{-7} when using single precision arithmetic like in the particular setup³. In the best case the optimal solution of the QP can be obtained with an accuracy of seven significant digits.

A rule of thumb states that the condition number of the Hessian is proportional to the degeneration of the accuracy of the QP solution due to roundoff errors ($\Delta n \propto \kappa(H)$). For an accurate QP solution, at least two digits have to be exact, which leads to the requirement that the condition number has to be lower than 10^5 .

The self tuning strategy proposed in this section provides optimal weighting factors for the MPC objective function while the basic parameters like horizons and constraints are considered constant and constraints on control action are given by the actuator limits. Notice that in this work we only use a linear representation of the QP MPC and linear simulation for the self tuning algorithm. Although a more general approach by using the full closed loop system with the MPC QP solver would be possible, but mainly for reasons of computational burden and focus on the later application this option was neglected.

4.1 Closed loop formulation of unconstrained MPC

In this section the closed loop plant representation required for the use in the parameter optimization is presented. With the unconstrained MPC QP formulation

$$\min_u \frac{1}{2} u^T H u + u^T f \quad (6)$$

and the optimal solution of this QP

$$U_{opt} = -H^{-1} \cdot f \quad (7)$$

it is possible to transform the unconstrained MPC as a linear feedback form. The feedback is inherited by the update formulation in the gradient f . The gradient can be split up into two parts a fixed one f^* and the vector Θ with the current state and reference information, see (4).

$$U_{opt} = -H^{-1} \cdot f^* \Theta = K_{MPC} \cdot \Theta \quad (8)$$

As well known in MPC U_{opt} contains all control actions in the control horizon n_{CH} , but the MPC only applies the first control move to the system and discards the remaining entries. In other words, the next control move u_k can be

² Notice that in an extreme case of a too high condition number in relation to the machine precision it might even lead to a highly imprecise and therefore possibly useless solution of the QP.

³ When a hardware with double precision arithmetic is used the machine precision is about 10^{-16} and in this case the ill conditioning is less relevant.

obtained by selecting only the first entry in the matrix and the affine control law for the unconstrained MPC is defined by

$$u_k = [k_{mpc,x_k}, k_{mpc,u_{k-1}}, k_{mpc,y_{ref}}] \cdot \Theta \quad (9)$$

With the unconstrained solution k_{mpc} of the MPC a closed loop representation of the whole plant can be formulated. For a fixed control and prediction horizon this representation is a function of the four tuning parameters (S_1, S_2, T_1, T_2) . Fig. 1 shows this scheme. The closed loop plant consists of the linear plant model, the MPC representation and a state observer. The inputs for this system are the two setpoint values $y_{ref,n}, y_{ref,\phi}$ and the measured disturbance from the requested compressor torque T_{req} and the resulting actual torque T_c , whereas the outputs are y_n and y_ϕ .

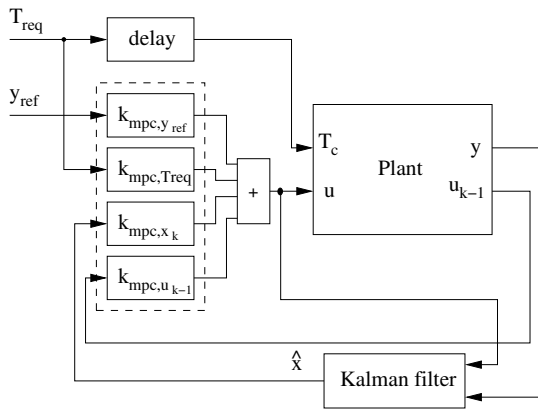


Fig. 1. Unconstrained closed loop plant representation

To reduce the computational effort during the parameter optimization the Hessian and the gradient can be rewritten in a tuning parameter dependent form. Due to the diagonal structure of the weighting matrices it is possible to separate each part in dependency of a single tuning parameter and afterwards combine the matrix again. Therefore, the Hessian can be formulated as $H = H_{S_1} \cdot S_1 + H_{S_2} \cdot S_2 + H_{T_1} \cdot T_1 + H_{T_2} \cdot T_2$. For example the relation $E^T \tilde{S} E$ can be separated in dependency of S_1 as

$$E^T \begin{bmatrix} S_1 & 0 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 & 0 \\ 0 & 0 & S_1 & 0 & 0 \\ 0 & 0 & 0 & S_2 & 0 \\ 0 & 0 & 0 & 0 & \ddots \end{bmatrix} E = S_1 \cdot \underbrace{E^T \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots \end{bmatrix} E}_{E_{S_1}} + S_2 \dots$$

to determine the formulation of the matrix H_{S_1} , which is given by $H_{S_1} = [E_{S_1}]_{(2,2)}$. With this extension it is not necessary to perform the condensing at every change of the tuning parameters and therefore the optimization is less time consuming.

4.2 Parameter optimization

The idea is to treat the closed loop plant as one single MIMO state space system only in dependency of the four remaining tuning parameters (S_1, S_2, T_1, T_2) and to perform the tuning with a time based approach.

For the time based strategy a tuning scenario has to be defined. In the actual case of the integral engine a set point

profile for $y_{i,ref}$ was used, where initially a *LS* change was done at constant n and ϕ and afterwards steps on n and ϕ were performed at constant load. The used scenario should lie within a representative operation and disturbance range of the compressor station to provide a suitable tuning criteria. Nevertheless it would be also possible to use several different tuning scenarios. Regarding the application at the gas engine the selected tuning scenario is typical for the main operation range. Additionally it should be mentioned that the simulations during optimization are performed with the linear closed loop model and are therefore subject to the feasible range of the model.

To determine the optimal parameters for this plant an optimization problem is defined, where the tracking error and the applied actuator energy are used as objective function. Here the numerical condition of the Hessian is implemented as a nonlinear constraint for the optimization problem, see (10c).

$$\min_{S_1, S_2, T_1, T_2} (J_{e,n} + s_{c1} J_{e,\phi} + s_{c2} J_{u,GFC} + s_{c3} J_{u,WG}) \quad (10a)$$

$$\text{s.t. } \{S_1, S_2, T_1, T_2\} > 0 \quad (10b)$$

$$\kappa(H) \leq \kappa_{ub} \quad (10c)$$

$$J_{e,n} = \sum_{k=0}^N (y_{k,n} - y_{k,nref})^2 \quad J_{e,\phi} = \sum_{k=0}^N (y_{k,\phi} - y_{k,\phi ref})^2$$

$$J_{u,GFC} = \sum_{k=0}^N (u_{k,GFC})^2 \quad J_{u,WG} = \sum_{k=0}^N (u_{k,WG})^2$$

With the length $N = \frac{t_{ref}}{T_s}$ according to the sampling time T_s and the time length of the reference trajectory t_{ref} . Notice that this optimization task is different to the MPC formulation and can be seen as an overlying loop to provide optimal parameters for the MPC in dependency of the given input profile and objective function. The weighting factors (s_{c_x}) in (10a) are used to provide appropriate weightings for all four criteria. The value of the upper bound κ_{ub} should be set to a reasonable value – during the tests the maximum allowed condition number was set to $\kappa_{ub} = 100$.

With this method alternative criteria like the numerical condition can be incorporated in the tuning process. This is motivated by the observation that in this application the objective function, when only considering tracking error and applied actuator energy, provides a similar value over a wide range of parameter combinations. For example in Fig. 2 both objective functions, whether with or without considering $\kappa(H)$, are compared⁴.

An alternative solution would be to directly consider the condition number in the cost function, which was tested too and essentially led to the same results. The optimization problem (10a) is a nonlinear program (NLP) which was solved by a sequential quadratic programming

⁴ Please note that due to the fact that there are four different tuning parameters which cannot be visualized within one figure, the presented objective functions were obtained by fixing the values for T_1 and T_2 and varying S_1 and S_2 . In addition it should be mentioned that the cost function is only presented for the feasible parameter range (10b).

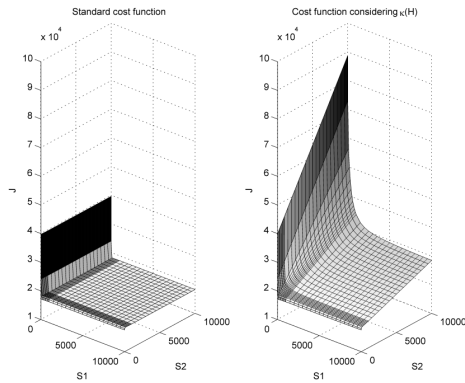


Fig. 2. Comparison between objective function with and without considering the numerical condition

Table 1. Set of parameters obtained by self tuning

	S_1	S_2	T_1	T_2	$\kappa(H)$
Initial	1	1	1	1	$4.9 \cdot 10^5$
Self tuning	0.117	$8.11 \cdot 10^3$	51.5	$1.03 \cdot 10^2$	100

method. This method approximates the NLP at every major iteration step by a QP and updates an estimate of the Hessian of the Lagrangian.

4.3 Algorithm

The tuning strategy can be summarized by the following steps:

- (1) **Condensing:** With the identified plant model the condensing is performed to provide the QP.
- (2) **Closed loop representation:** Based on the optimal QP solution the closed loop system is transferred into one state space representation.
- (3) **Parameter optimization:** A time domain based optimization is performed with a predefined evaluation scenario considering the tracking performance and control action as cost function and the numerical condition as nonlinear constraint.
- (4) **Verification:** The control performance is evaluated for the linear plant.

5. RESULTS

The presented tuning algorithm was used to obtain the parameters for the MPC, whereas the required initial parameters were set to 1. In Table 1 the initial values and the gained parameters are listed and additionally the condition number of H is given. The corresponding trajectories for both parameter sets are depicted in Fig. 3. It can be seen that the initial values provide an inferior performance regarding the ϕ tracking but due to the optimization the performance of the tracking was enhanced.

To evaluate the performance of the self tuning MPC a (more detailed) nonlinear simulation model of a compressor station was used, which was calibrated with measurements from the real station. In addition in this simulation environment also the available numerical accuracy of the QP solver on the target system, discretization effects and measurement noise are considered.

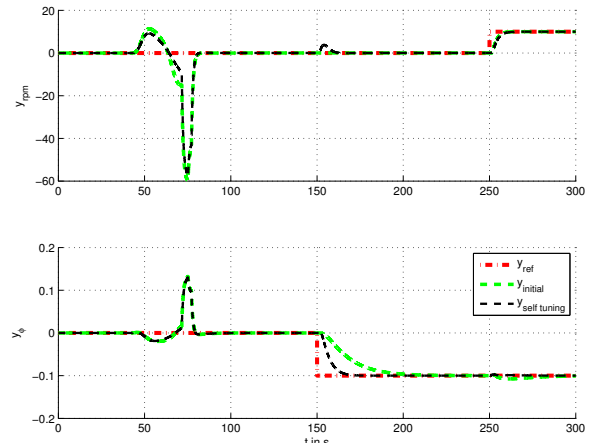


Fig. 3. Result of self tuning at linear simulation model

Table 2. Comparison of control performance

Tuning	$J_{e,n}$	$J_{e,\phi}$	$J_{u,GFC}$	$J_{u,WG}$	$\kappa(H)$
On-site	100 %	100 %	100 %	100 %	$7.4 \cdot 10^5$
Self tuning	78.2 %	94.8 %	100.1 %	100.1 %	100

Based on the obtained self tuning parameters from Table 1 a simulation on the described nonlinear model was performed. As test scenario a representative load change at $t = 20$ s and one setpoint change on each reference was performed. To compare the performance of the parameters obtained by the self tuning method the trajectories are compared against a parameter set which was tuned on site by an experienced control engineer. In Fig. 4 the resulting trajectories are shown, where it can be seen that the achievable performance is similar to the one with manual tuning.

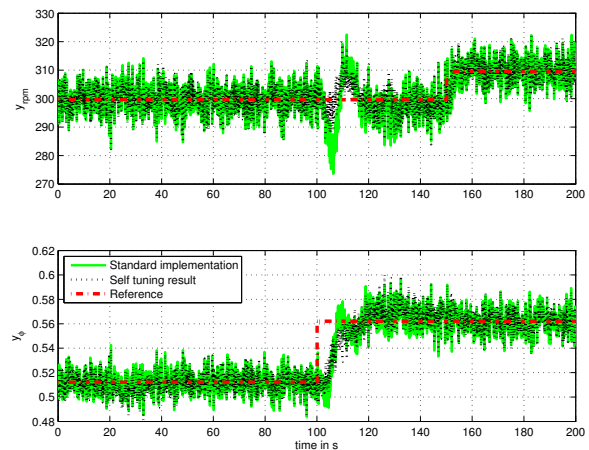


Fig. 4. Output tracking with default implementation tuned on-site and with the self tuning result

In Table 2 the cost functions for the integrated squared tracking error and the used actuator energy are presented – the relative values are scaled to the performance achieved by on-site tuning. In the last column the condition number of the Hessian is listed, where a major improvement compared to the manual tuning can be obtained by the self tuning strategy. It was possible to reduce the condition of the Hessian from $\approx 10^5$ to 10^2 . The immediate consequence of this reduction is summarized in Table 3 and

Table 3. Comparison of needed QP iterations

	max. No. iterations	avg. No. iterations
On-site tuning	10	5,131
Self tuning	6	0.006

can also be seen in Fig. 5, where the number of necessary iterations of the QP solver is depicted for the standard and the self tuning case. Due to implementation reasons on the real hardware system the maximum number of allowed iterations was set to 10. As it can be seen the number of iteration often approaches the allowed limit when using the manual tuning parameters, whereas in the self tuning case the QP often could be solved within the first iteration. Only during the setpoint change more iterations were required.

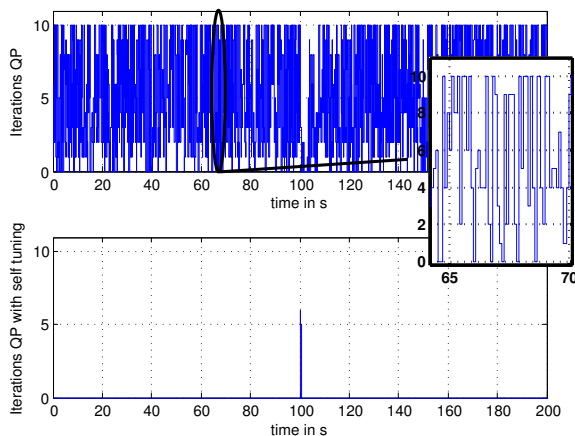


Fig. 5. Necessary QP Iterations in standard case or with self tuning

A reason for the high number of necessary iterations in the standard case is, that due to the ill conditioning the amplified roundoff errors prevent the QP from converging to the solution. Indeed, when comparing the tracking performance it seems that the obtained solution of the QP-solver with the standard tuning is sufficient close. This can be a benefit of the used homotopy path strategy in qpOASES that provides a sequence of optimal solutions along the homotopy path. So the solution at each iteration, even if not enough time for the calculation of the optimal solution is available, tends to the optimal solution of the QP. Another reason for the increase of iterations could be the fact that the used feedback information in the QP (i.e. the state information provided by the Kalman filter) contains measurement noise, which can affect the QP and the optimal solution. However for both experiments a Kalman filter with identical settings was used.

6. CONCLUSIONS AND FUTURE WORKS

Although the problem of limited available numerical precision is not that frequent it can become important on embedded systems. In this work an important additional tuning aspect, to consider the numerical condition, is introduced. The application of the proposed self tuning strategies obtained satisfactory results. It was possible to achieve an equal control performance as a controller which was tuned by an experienced control engineer during a previous field test at the station, while the numerical

efficiency was improved distinctly. It should be also mentioned, that the proposed method is not limited to this particular application and could be used for different MPC problems in applications with restrictions in the available numerical accuracy. The near future work will focus on an implementation on the real engine to further evaluate the proposed self tuning method.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the sponsoring of this work by the COMET K2 Center "Austrian Center of Competence in Mechatronics (ACCM)". The COMET Program is funded by the Austrian federal government, the Federal State Upper Austria and the Scientific Partners of ACCM.

REFERENCES

- A. Al-Ghazzawi, E. Ali, A. Nouh, and E. Zafriou. On-line tuning strategy for model predictive controllers. *Journal of Process Control*, 11(3):265–284, June 2001.
- D. Alberer, M. Ranzmaier, L. del Re, and M. Huschenbett. MIMO model predictive control for integral gas engines under switching disturbances. In *IEEE International Conference on Control Applications, 2008. CCA 2008*, pages 317–322, 2008.
- J. Ångeby, M. Huschenbett, and D. Alberer. MIMO Model Predictive Control for Integral Gas Engines. In *Models, Methods and Applications, Workshop on Automotive Model Predictive Control*, Feldkirchen, Austria, February 2009.
- A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The Explicit Solution of Model Predictive Control via Multiparametric Quadratic Programming. In *Proceedings of the American Control Conference*, pages 872–876, Chicago Illinois, June 2000.
- EF Camacho and C. Bordons. *Model predictive control*. Springer Verlag, London, 2004.
- S. Di Cairano and A. Bemporad. Model predictive control tuning by controller matching. *IEEE Transactions on Automatic Control*, 55(1):185–190, jan. 2010.
- J. Fan. *Model predictive control for multiple cross-directional processes: Analysis, Tuning and Implementation*. PhD thesis, University of British Columbia, Vancouver, Canada, September 2003.
- J. Fan and G.E. Stewart. *Automatic Tuning Method for multivariable model predictive controllers*, 2007.
- H.J. Ferreau, P. Ortner, P. Langthaler, L. del Re, and M. Diehl. Predictive control of a real-world diesel engine using an extended online active set strategy. *Annual Reviews in Control*, 31(2):293 – 301, 2007.
- HJ Ferreau, HG Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.
- J.M. Maciejowski. *Predictive control: with constraints*. Pearson education, Harlow, UK, 2002.
- R. Soeterboek and S. Toumodge. *Predictive control: A unified approach*. Prentice Hall, Englewood Cliffs, NJ, 1992.
- P. Vega and M. Francisco. Norm based approaches for automatic tuning of Model Based Predictive Control. *Proceedings of ECCE-6 (Copenhagen)*, September 2007.