Tracking explicit model predictive controllers for low-level control applications

Samo Gerkšič, Boštjan Pregelj

Department of Systems and Control, Jožef Stefan Institute

Introduction

Model Predictive Control (MPC) [bibQB03, bibMac02] is a family of advanced control algorithms based on on-line optimisation of predicted control signals using process models. The main practical advantages of MPC over conventional feedback control methods are twofold: the ability to account for constraints on signals, and consistent yet simplified control design for multivariable processes. MPC is a rare example of an advanced control method that is well established in industrial practice. Typically, MPC is used at the medium level of the control hierarchy, between the top level of plant optimisation and the bottom level of physical control. Mainstream MPC approaches rely on on-line optimisation, which involves considerable computational requirements. Consequently, capable hardware equipment is required, and sampling rates are restricted.

The idea of the recently invented explicit[1] form of MPC (eMPC) [bibPDBBM00, bibBMDP02, bibPGD07] is to move the bulk of the computational load into the off-line phase of controller design rather than doing it on-line. Hence a polyhedral partition of the whole parameter space of the controller is computed, for example by using multi-parametric quadratic programming (mpQP) [bibBao02, bibGBTM04, bibTJB03a, bibTJB03b, bibMR03, bibSKJTJ06, bibSTJ07] or multi-parametric linear complementarity (mpLC) [bibJM06] algorithms. In the on-line phase, optimisation is replaced by a simple region-search algorithm [bibTJB03c]. This enables controller implementation on standard industrial automation equipment [bibKRM10], and the application niche is extended to processes with fast dynamics. eMPC suffers from high computational demand in the off-line design phase that grows exponentially with the dimensions of the problem. More precisely, it grows with the number of possible combinations of active constraints[2] within the prediction horizon. In addition, the region-search algorithm turns out to be efficient only with small-scale problems, i.e. it may take

[1] Also known as multi-parametric.

[2] Active constraints are those inequality constraints of the eMPC problem setup which in certain conditions are at the equality boundary.

longer than the corresponding on-line QP for larger problems [bibZJM11]. Therefore, the currently known algorithms are applicable to univariable or small-scale multivariable processes, typically at the bottom (physical) level of the control hierarchy or for small stand-alone applications.

eMPC provides an explicit view of the structure of the control law, which is hidden in conventional MPC. The controller comprises a set of local affine control laws, each of them being valid in a polyhedral region characterised by the same set of active constraints. The properties of the controller in each of those regions may be studied by using local linear analysis (LLA) of the closed-loop system [bibGSB08, bibGP09]. LLA is particularly useful for tuning in the region with no active constraints, and may also be used for other regions in which the controller tends to dwell. In addition, it may be used for clusters of regions with the same or very similar control parameters. LLA helps analyse the dynamical properties that are important with controller tuning for efficient tracking, the rejection of various types of disturbances, noise attenuation, and robustness to plant-model mismatch. The latter is particularly important as the closed-loop system comprising a state controller and a state estimator may be oversensitive to the modelling error.

Issues related to the tuning of the eMPC controller for efficient and robust feedback performance are vital for application at the physical control level. The application of eMPC seems attractive in cases where eMPC algorithms are expected to be superior to PID control due to the more versatile controller structure as well as due to the improved constraints handling. However, practical applications of eMPC have been scarce so far, which is partly because the transition from eMPC theory to practical tracking controllers is not straightforward (see Section secTrackeMPC). With many theoretically sound approaches, the practically achievable prediction horizons are short, which severely impairs the ability to react to approaching violations of state or output constraints in time. Relatively long sampling times are typically used in eMPC, which may result in less efficient disturbance rejection compared to conventional PID control. Some simplified tracking approaches may also affect control performance. In addition, it is questionable whether the advanced constraints-handling of eMPC can lead to any significant practical advantage in simple univariable PID replacement applications, because univariable processes lack degrees of freedom in ranking control priorities.

The aim of this work is to develop an offset-free tracking eMPC scheme based on the disturbance-estimation approach and a "joint" controller structure (with no target calculator), which allows implementation of eMPC with relatively fast sampling and reasonably long horizons suitable for practical physical control applications, and test the scheme in an industrial pilot application where it is compared with an existing single-loop PID controller. We first summarise the basic state-control eMPC problem with a state estimator. Then, we discuss several approaches to set-point-tracking with integral action in eMPC, and describe the particular offset-free tracking setup which is used. The control method is tested in a case study of cooling water temperature control in a biogas-fuelled 350 kW combined heat and power production (CHP) unit of a municipal waste-water treatment plant

(WWTP). A specific aim of the case study is to verify whether unnecessary excursions of the cooling water temperature away from its set-point in the critical range near output constraints can be reduced with eMPC control. The eMPC controllers were designed using LLA and tested with a simplified simulation model, which includes linear nominal dynamics, disturbances that are present in the system, and constraints on the process signals. The design was verified experimentally on the CHP unit. The performance improvements due to eMPC feedback tuning and constraints-handling ability were examined. Finally, several implementation issues related to eMPC are discussed.

Explicit MPC

We begin by outlining a simple eMPC controller for constrained linear systems. Just like in conventional MPC [bibQB03, bibMac02], control is based on a state-space process model

$$\mathbf{x}(k+1) = \mathbf{A}_{\mathrm{m}}\mathbf{x}(k) + \mathbf{B}_{\mathrm{m}}\mathbf{u}(k), \qquad \mathbf{y}(k) = \mathbf{C}_{\mathrm{m}}\mathbf{x}(k) \qquad \text{(eqmodelMPC)}$$

where k is the discrete-time index, $\mathbf{u} \in \Re^{n_u}$ and $\mathbf{y} \in \Re^{n_y}$ the input and output signal vectors, $\mathbf{x} \in \Re^{n_x}$ the state vector, and $\mathbf{A}_{\mathrm{m}}$, $\mathbf{B}_{\mathrm{m}}$, and $\mathbf{C}_{\mathrm{m}}$ the model matrices. The optimality of control is defined with the 2-norm constrained finite-time optimal control (CFTOC) problem [bibBao02]. The CFTOC value function is

$$J^* = \min_{\tilde{\mathbf{u}}_{N_u}} J(\tilde{\mathbf{u}}_{N_u}), \qquad J(\tilde{\mathbf{u}}_{N_u}) = \sum_{j=0}^{N-1} \mathbf{x}_k^T(k+j)\mathbf{Q}_x\mathbf{x}(k+j) + \sum_{j=0}^{N_u-1} \mathbf{u}^T(k+j)\mathbf{R}_u\mathbf{u}(k+j) \; \text{(eqMPCcost)}$$

where N and N_u are the prediction and control horizons, $\mathbf{Q}_x$ and $\mathbf{R}_u$ the state and control weights, $\tilde{\mathbf{u}}_{N_u} = \left[\mathbf{u}^T(k) \quad \mathbf{u}^T(k+1) \quad \cdots \quad \mathbf{u}^T(k+N_u-1)\right]^T$ is the optimised sequence of current and future control signal values, and discrete-time indices in subscripts denote model predictions. The CFTOC problem typically includes the constraints

$$\mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}, \qquad \mathbf{x}_{\min} \le \mathbf{x} \le \mathbf{x}_{\max} \; \text{(eqMPCconstraints)}$$

which have to be taken into account when optimising (eqMPCcost). A receding-horizon implementation of the control law is used, meaning that only the first element $\mathbf{u}(k)$ taken from $\tilde{\mathbf{u}}_{N_u}$ is used as the current control action. The remaining elements of $\tilde{\mathbf{u}}_{N_u}$ are discarded and a new optimisation run is carried out for the next time instant with shifted horizons.

In conventional MPC, the optimizer $\tilde{\mathbf{u}}_{N_u}$ is typically computed by substituting a series of state update equations (eqmodelMPC) into the cost function (eqMPCcost) and rewriting the expanded (eqMPCcost) and constraints (eqMPCconstraints) to the standard QP form (see, e.g., Chapter 3 in [bibMac02])

$$J^* = \min_{\mathbf{z}} J(\mathbf{z}), \quad J(\mathbf{z}) = \tfrac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} + \mathbf{c}^T\mathbf{z} \quad \text{subject to} \quad \mathbf{A}\mathbf{z} \geq \mathbf{b} \qquad \text{(eqQP)}$$

where the optimization argument vector $\mathbf{z} \in \Re^n$ matches $\tilde{\mathbf{u}}_{N_u}$, $\mathbf{H} \in \Re^{n\times n}$ is the symmetric positive definite quadratic-term cost matrix, $\mathbf{c} \in \Re^{n\times 1}$ is the linear-term cost vector (the constant term is omitted because it does not affect the optimisation result), and $\mathbf{A} \in \Re^{q\times n}$ and $\mathbf{b} \in \Re^{q\times 1}$ define the stack of linear inequalities. This problem is solved in each sampling step using a suitable QP solver. However, in [bibPDBBM00, bibBMDP02] it was established that the problem (eqMPCcost)-(eqMPCconstraints) may be solved *parametrically* with respect to the vector of parameters $\boldsymbol{\theta} \in \Re^s$, where the *multi-parametric QP* problem form

$$J^*(\boldsymbol{\theta}) = \min_{\mathbf{z}}\left(\tfrac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} + \boldsymbol{\theta}^T\mathbf{F}^T\mathbf{z} + \mathbf{c}^T\mathbf{z}\right) \quad \text{subject to} \quad \mathbf{A}\mathbf{z} \geq \mathbf{S}\boldsymbol{\theta} + \mathbf{b} \qquad \text{(eqmpQP)}$$

is used, with the additional terms comprising $\mathbf{F} \in \Re^{s\times n}$ in the cost and $\mathbf{S} \in \Re^{q\times s}$ in the constraints, where $\boldsymbol{\theta}$ corresponds to the state $\mathbf{x}$ in the MPC formulation. Assuming a strictly positive definite $\mathbf{H}$, the optimizer $\mathbf{z}$* is a continuous piecewise-affine function of $\boldsymbol{\theta}$ on a full-dimensional set of admissible parameters $\Theta \subset \Re^s$ partitioned into non-overlapping convex polyhedral regions R_i, $\Theta = \bigcup_{i=1}^{N_R} R_i$, and the value function J* is a continuous piecewise-quadratic function of $\boldsymbol{\theta}$ on Θ.

Conversion of the MPC control problem (eqMPCcost)-(eqMPCconstraints) into the mpQP form (eqmpQP) is supported by several tools, e.g. the Hybrid Toolbox [bibBem06] and the open-source Multi-Parametric Toolbox [bibKGB05]. The latter includes two approaches. In the first approach, the mpQP matrices and vectors are built directly using the function `mpt_constructMatrices`. The more flexible second approach first translates the MPC problem into a Yalmip formulation using the function `mpt_yalmipcftoc`, then the mpQP matrices and vectors are generated by the Yalmip solver [bibYALMIP04].

The task of an mpQP solver is to determine the polyhedral partition $\{R_i \mid i = 1, ..., N_R\}$ and the corresponding optimisers $\mathbf{z}^*(\boldsymbol{\theta}) = \mathbf{F}_i\boldsymbol{\theta} + \mathbf{g}_i$, where $\mathbf{F}_i \in \Re^{n\times s}$ and $\mathbf{g}_i \in \Re^n$. Such solvers are available, for example, in academic eMPC libraries [bibKGB05, bibBem06]. The solvers determine the partition by first finding an initial feasible point and its associated full-dimensional region. Then they explore the adjacent space until Θ is fully covered. Each of the regions is characterised by the cost function of (eqmpQP) and the set of active constraints.

With the pre-computed polyhedral partition, on-line optimisation using a QP solver is no longer needed to compute the control signal $\mathbf{u}(k)$ for the current state

$\mathbf{x}(k)$ in each sampling instant. Instead, the active region is determined, and the corresponding local affine control law is applied. The active region may be determined using a region-search algorithm [bibTJB03c].

In most practical cases, however, the state $\mathbf{x}(k)$ is not measured, and the signals are disturbed by noise terms. The underlying model reads

$$\mathbf{x}(k+1) = \mathbf{A}_{\mathrm{m}}\mathbf{x}(k) + \mathbf{B}_{\mathrm{m}}\mathbf{u}(k) + \mathbf{G}_{\mathrm{m}}\mathbf{w}(k), \qquad \mathbf{y}(k) = \mathbf{C}_{\mathrm{m}}\mathbf{x}(k) + \mathbf{v}(k) \tag{eqmodelMPCnoise}$$

where $\mathbf{w} \in \Re^{n_w}$ is the state noise and $\mathbf{v} \in \Re^{n_y}$ the output noise, with covariance matrices $\mathbf{Q}_{K\mathrm{m}}$ = E$\{\mathbf{w}\mathbf{w}^T\}$ and $\mathbf{R}_K$ = E$\{\mathbf{v}\mathbf{v}^T\}$, respectively, assuming E$\{\mathbf{w}\mathbf{v}^T\}$ = 0; and $\mathbf{G}_{\mathrm{m}}$ specifies the access of noise to the state. Instead of the state $\mathbf{x}(k)$, the state estimate $\hat{\mathbf{x}}(k|k)$ is used in the control problem (eqMPCcost)-(eqMPCconstraints). It is computed by using the steady-state Kalman filter (KF)

$$\begin{aligned}
&\hat{\mathbf{x}}(k|k-1) = \mathbf{A}_{\mathrm{m}}\hat{\mathbf{x}}(k-1|k-1) + \mathbf{B}_{\mathrm{m}}\mathbf{u}(k-1) \\
&\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{M}_{K\mathrm{m}}\left[\mathbf{y}(k) - \mathbf{C}_{\mathrm{m}}\hat{\mathbf{x}}(k|k-1)\right]
\end{aligned}$$

where the Kalman gain $\mathbf{M}_{K\mathrm{m}}$ is calculated using the steady-state solution of the Riccati equation with $\mathbf{Q}_{K\mathrm{m}}$ and $\mathbf{R}_K$ [bibMac02].

secTrackeMPC Practical Tracking Controllers with eMPC

In practice, the academic MPC setup (eqMPCcost) needs to be extended in order to meet the requirements for offset-free tracking of asymptotically non-zero reference signals and in the presence of disturbances with asymptotically non-zero mean values, which are typically present in practical applications. A variety of tracking approaches may be applied. Specific approaches that depart from those typically used in conventional MPC appear in the published application-related eMPC literature. The problem is that the additional variables in tracking set-ups, such as disturbances and adjustable set-points, increase the parametric dimension s of the mpQP and the computational complexity of the mpQP. Another problem is that the available mpQP solvers appear to not be sufficiently numerically robust for the approaches where estimated disturbances appear as mpQP parameters and when relatively fast sampling is used [bibGer11]. We first discuss some of the considerations, then present the approach used in the case-study.

The tracking option readily available in [bibKGB05, bibBem06] is variable set-point tracking without integral action, meaning that a steady-state offset due to disturbances may occur. In [bibKGB05, bibBem06] this is implemented by augmenting the model with the integral state $\mathbf{y}_r$. An alternative implementation is possible where the reference is included in the cost function, separately from process dynamics. Constant set-point signals may be implemented using coordinate shifts by substitutions such as $\mathbf{y}_n = \mathbf{y} - \mathbf{y}_{rf}$, $\mathbf{u}_\mathrm{n} = \mathbf{u} - \mathbf{u}_{rf}$, or $\mathbf{x}_\mathrm{n} = \mathbf{x} - \mathbf{x}_{rf}$, where $\mathbf{y}_{rf}$, $\mathbf{u}_{rf}$, and $\mathbf{x}_{rf}$ are fixed target values for $\mathbf{y}$, $\mathbf{u}$, and $\mathbf{x}$, respectively. Such substitutions do not increase the complexity of the mpQP problem, as the dimension of $\boldsymbol{\theta}$ remains unchanged. If only a few discrete values of $\mathbf{y}_r$ are used, switching among a set of controllers may be used instead of $\mathbf{y}_r$, which in that case appears as an additional parameter. Penalising the control signal $\mathbf{u}$ as in (eqMPCcost) may cause tracking offset, which may be avoided by modifying the MPC cost (eqMPCcost) either by penalising the control moves $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k–1)$ with the weight $\mathbf{R}_{du}$, or by penalising the deviation of $\mathbf{u}$ from an appropriate $\mathbf{u}_r$ target instead of using the default control cost.

Tracking-error integration (TEI) is one possible offset-free tracking concept appearing in eMPC literature [bibBem06, bibGJK04]. With this approach, the model (eqmodelMPC) is first augmented with $\mathbf{y}_r$ so that the output is the tracking error ($\mathbf{y}_r - \mathbf{y}$); then, an integrator state that integrates ($\mathbf{y}_r - \mathbf{y}$) is appended. Alternatively, TEI may be implemented by adding an integral term to the MPC cost function (eqMPCcost). This approach does not require any estimator. Also, there is no need for a velocity-form augmentation if there are no rate constraints. However, integrator wind-up may occur due to unreachable set-points unless suitable anti-windup protection is used. It was also observed that a significant integral action resulting therefrom may increase overshoot in nominal tracking performance, similarly as the integrating term in PI control [bibPG08].

Disturbance estimation (DE) is an alternative offset-free tracking concept, more commonly used in traditional MPC [bibMac02, bibMB02, bibPR03]. Here, the basic model (eqmodelMPC) is appended with a model of unmeasured disturbances of an integrating character. This may be a simple "output disturbance model" comprising an integrator at the model output or some more elaborate disturbance models. With the Box-Jenkins modelling approach, the unmeasured-disturbance branch of the model is kept separate from the basic model branch, while in the AR(I)MAX model (auto-regressive integrating moving-average with exogenous input) the two branches sharing the denominator dynamics are merged. An observer or Kalman filter may be used for state estimation. The construction of the augmented model can be carried out in several different ways [bibGSB08, bibGAM08].

DE may be used in a target calculator (TC) scheme [bibMR93, bibPR03, bibSak03, bibSDPP04, bibPLR06] which decomposes the controller into the TC, which handles the steady-state, and the remaining dynamic controller (DC), which deals with transient dynamics. The total controller output is the sum of TC and DC contributions. In the eMPC context, the TC scheme may be seen as a bi-level mpQP, where the outputs of the TC, $\mathbf{x}_t$ and $\mathbf{u}_t$, appear as variable references for the DC [bibSak03]. The variable references $\mathbf{x}_t$ and $\mathbf{u}_t$ may be replaced with $\mathbf{y}_r$ and the DE integrator state, which may yield a lower dimension of $\boldsymbol{\theta}$**.** This is carried out by substituting the parametric solution of the TC into the DC mpQP problem, which is then computed for each region of the TC partition separately. This tends to result in a large joint number of regions. However, the TC scheme has several advantages; it may be used with infinite-horizon MPC costs; it allows steady-state infeasibilities to be easily screened out; for offset-free tracking it does not require the use of velocity form augmentation (needed only with rate constraints). On the other hand, the decomposition of the controller leads to certain suboptimality and feasibility problems, as the TC is oblivious to transient infeasibilities arising in the DC [bibRR99], and complicates the structure of the control law and analysis of the constrained performance. In an enhanced TC scheme [bibLAAC08], this decomposition is removed, however at the cost of computational complexity.

In our case-study, the DE concept is applied with a "joint" controller integrating the functions of constrained dynamic control and offset-free tracking [bibGP09]. This method was used because it results in a simple controller structure that allows straightforward analysis and because controllers with relatively low numbers of regions were obtained. No significant difference in performance was observed between the two DE schemes. Although typically used for square plants, the joint scheme can also efficiently handle processes with redundant control input despite the absence of a TC [bibGP09].

Implementation of the joint scheme

For the implementation of the joint-scheme DE concept following the ARIMAX approach in the state-space form, disturbance augmentation of the basic model (eqmodelMPCnoise) may be carried out by appending the additional integrator state **e** with the associated noise signal $\mathbf{w}_\mathbf{e}(k)$ either at the output

$$\begin{gathered}\begin{bmatrix}\mathbf{x}(k+1)\\ \mathbf{e}(k+1)\end{bmatrix}=\begin{bmatrix}\mathbf{A}_\mathrm{m} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{e}(k)\end{bmatrix}+\begin{bmatrix}\mathbf{B}_\mathrm{m}\\ \mathbf{0}\end{bmatrix}\mathbf{u}(k)+\begin{bmatrix}\mathbf{G}_\mathrm{m} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{w}(k)\\ \mathbf{w}_\mathbf{e}(k)\end{bmatrix}\\ \mathbf{y}(k)=\begin{bmatrix}\mathbf{C}_\mathrm{m} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{e}(k)\end{bmatrix}+\mathbf{v}(k)\end{gathered}\tag{eqmodel_dist_o}$$

or at the input

$$\begin{gathered}\begin{bmatrix}\mathbf{x}(k+1)\\ \mathbf{e}(k+1)\end{bmatrix}=\begin{bmatrix}\mathbf{A}_\mathrm{m} & \mathbf{B}_\mathrm{m}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{e}(k)\end{bmatrix}+\begin{bmatrix}\mathbf{B}_\mathrm{m}\\ \mathbf{0}\end{bmatrix}\mathbf{u}(k)+\begin{bmatrix}\mathbf{G}_\mathrm{m} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{w}(k)\\ \mathbf{w}_\mathbf{e}(k)\end{bmatrix}\\ \mathbf{y}(k)=\begin{bmatrix}\mathbf{C}_\mathrm{m} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{e}(k)\end{bmatrix}+\mathbf{v}(k)\end{gathered}\tag{eqmodel_dist_i}$$

(for details regarding the choice of the integrator position, see [bibMB02], [bibPR03]). Either of the two is further denoted as the disturbance-augmented model ($\mathbf{A}_a$, $\mathbf{B}_a$, $\mathbf{C}_a$, $\mathbf{G}_a$), with the state $\mathbf{x}_a = [\mathbf{x}^T\ \mathbf{e}^T]^T$ and noise input $\mathbf{w}_a = [\mathbf{w}^T\ \mathbf{w}_\mathbf{e}$

$^{T}]^{T}$. Its state is estimated using a steady-state KF with Kalman gain $\mathbf{M}_K$ calculated from the steady-state solution of the Riccati equation with $\mathbf{Q}_K = \mathrm{E}\{\mathbf{w}_a\mathbf{w}_a{}^T\}$ and $\mathbf{R}_K$.

Using the MPT toolbox [bibKGB05], the MPC controller for the "joint" scheme is constructed by specifying the disturbance-augmented model ($\mathbf{A}_a$, $\mathbf{B}_a$, $\mathbf{C}_a$, $\mathbf{G}_a$), the signal constraints, and the MPC cost matrices as inputs to the function `mpt_control`. Within the function, the disturbance-augmented model further undergoes a velocity-and-tracking augmentation, as described in [bibKGB05], so that the state vector of the resulting disturbance-velocity-tracking augmented model (corresponding to the parameter vector $\boldsymbol{\theta}$ of the parametric solution) is $\mathbf{x}_{avt} = \left[\mathbf{x}^T \;\; \mathbf{e}^T \;\; \mathbf{u}^T(k-1) \;\; \mathbf{y}_r^T\right]^T$ [bibGP09]. Then the mpQP problem (eqmpQP) is formulated and solved using the mpQP solver.

Industrial Case Study

This section presents a field test where possible performance improvements due to replacing an existing PID controller with eMPC in a bottom-level control application were investigated. eMPC was successfully tested both in simulation and in plant experiments. What we managed to achieve was a slight improvement in performance compared to PID control in unconstrained operational conditions. However, eMPC was unable to provide a significant improvement in performance near the output constraints, which was the main motivation for the study.

Process Description

Biogas is one of the products of waste decomposition at the waste-water treatment plant. It is used as fuel for a gas-fired rotation-engine combined heat-and-power production unit, see Figure figCHP_photo. The engine drives a 350 kW electric generator, sending energy to the grid. It also produces heating power, which is used for heating in various facilities of the plant. Due to the free fuel (biogas) and subsidized rates for electric energy from renewable sources, the unit mostly operates at constant maximum power. The biogas production exceeds the engine consumption, and storage capacity is limited, so the unused biogas must be conveyed to a burner and combusted. When the heat produced exceeds the thermal load, it is dissipated via a forced cooling subsystem, which may operate continuously in the summer season.

Fig. figCHP_photo**.** A biogas-fuelled CHP unit

Figure figCHP_diagram shows a simplified schematic diagram of the unit focussed on heat transfer control. Black lines indicate the primary engine cooling water cycle (motor water, MW), dark-grey lines indicate the secondary water cycle for heating (HW), the light-grey line indicates the forced cooling water cycle, and thin black lines indicate the control signals.

The main disturbances acting on the system originate in the thermal load of the heating network. Heating water is used for several WWTP subsystems, which may be switched on or off and have local control systems that control water flow. This may cause considerable changes in thermal load impedance, as many valves may become open or closed, depending on the outside temperature.

The process output y in our case is the temperature of water going back to the engine (T101). In the existing control scheme, the PI controller attempts to keep the process output at the set-point y_r = 78 ºC by using the position of the three-way valve V121 (the manipulated variable u). The valve position range is from 0 to 100 [%]. When fully open, all water is directed towards the heat exchangers. When the valve is closed, during engine warm-up, water returns directly to the engine. The set-point is selected for optimal efficiency of the engine and is generally constant. An increase in the temperature may reduce engine lifetime. Therefore, if the temperature rises above 80 ºC, the forced cooling system is activated. While this is normal at a low thermal load, activations of forced cooling due to disturbance transients at normal thermal loads result in unwanted wastage of heat. Therefore, the aim of alternative control algorithms is to reduce transients due to disturbances near the threshold value.

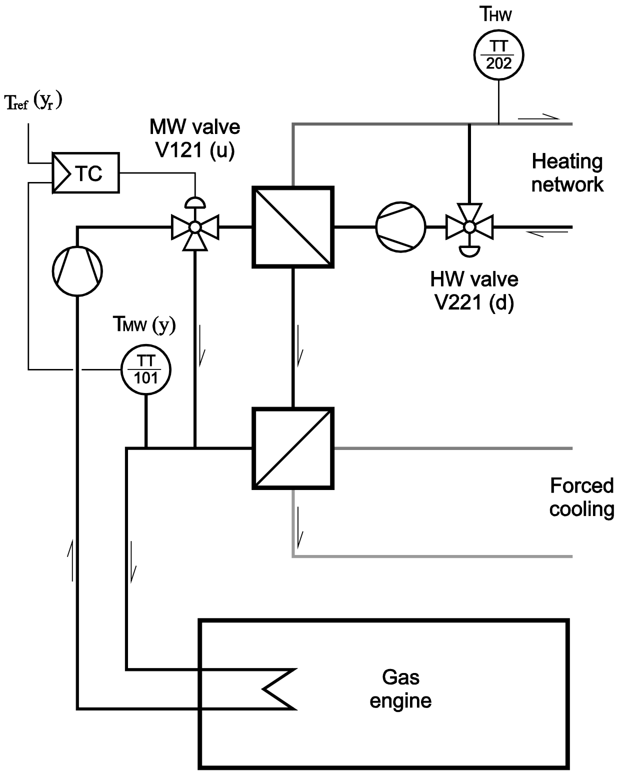

Fig. figCHP_diagram**.** Simplified schematic diagram of the CHP unit

The disturbance dynamics are slower than the open-loop process dynamics. The nearest source of disturbances is the secondary water (HW) three-way valve V221. Normally this valve is used in a HW temperature control loop that regulates the temperature T202. During the experiments with controllers of the MW loop, the HW controller was switched to manual, and disturbances were generated by changing the V221 position (unmeasured disturbance signal *d*). Undesired oscillatory disturbances of small amplitude can be noticed in all measurement signals, even in open-loop control. Generally, controllers tend to amplify this oscillation, but good control performance may be achieved if they are tuned for low sensitivity to measurement noise.

Both valves are driven by electric motors. Positioning is performed incrementally by on/off switching of the motor in both directions. There is no absolute position measurement, so the position is estimated by integration. Both valves require 120 s to move from a fully open to fully closed position or vice versa, which implies additional rate constraints that need to be considered. A dead zone is used to respect the minimum movement time of 1.2 s. This causes some tracking offset and a specific valve movement pattern.

Figure figPI_exp shows an experimental run with the existing PI controller. The top graph displays the process output y (temperature T101 in °C, solid line) and its reference signal y_r (dotted line). The bottom graph displays the controller output u (valve V121 position in %, solid line) and the load disturbance signal d (valve V221 position in %, dotted line). The test sequence comprises pairs of consecutive step changes of signals in opposite directions (first away from the initial operating point and then back towards it):

1. Set-point tracking: step change of y_r, step amplitudes –3 °C and 3 °C;
2. Rejection of the input disturbance: step changes of the artificial disturbance entering at u, step amplitudes –15% and 15%;
3. Rejection of the output disturbance: step changes of the artificial disturbance entering at y, step amplitudes –2 °C and 2 °C; and
4. Rejection of the load disturbance: step change of d (V221 position), step amplitudes –40% and 40%.

The existing PI controller was initially tuned using magnitude-optimum-based tuning rules [bibVPS99]. The integral time constant T_I was 110 s, and the proportional gain was K_P = 18. With this initial tuning shown in Figure figPI_exp up to t = 800 s, tracking performance and suppression of step disturbances were good, but suppression of measurement noise was not satisfactory since the variance of the position of valve V121 was too high. With the reduced value K_P = 8 after t = 800 s, the control signal variance was reduced, but still above the desired level. The amplitude of the step change of d was set too high, and caused y to rise above the limit 80 °C, as forced cooling was disconnected during the experiment. The disturbance amplitude was decreased in the subsequent experiments.

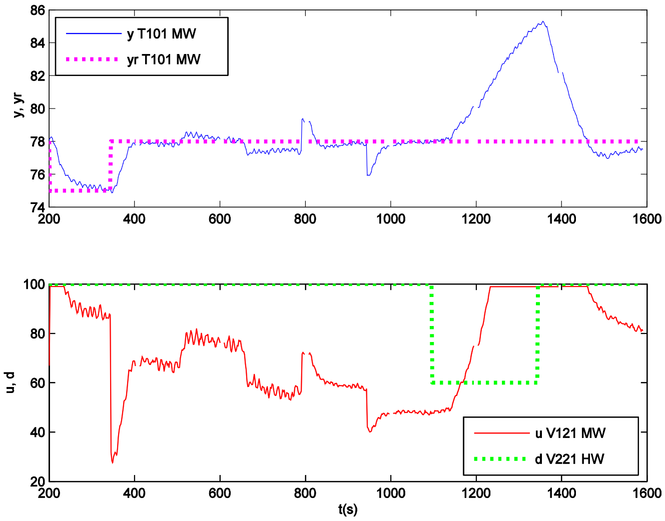

Fig. figPI_exp. PI controller, experiment

Modelling

A linear discrete-time state-space nominal model describing the dynamics from *u* to *y* is required for eMPC design. Due to the nonlinearity of the plant, such a model is obviously a poor approximation of the process dynamics. Since the process dynamics are relatively slow, experimentation is time-consuming and detailed nonlinear modelling is prohibitively expensive with respect to the anticipated gains. In industry, PID controllers for such processes are commonly tuned by applying tuning rules to linear models obtained from step response tests at the most challenging operating point. eMPC is being considered as a straightforward replacement for PID control and, consequently, the modelling and experimentation effort should be comparable to that of tuning a PID loop. The running assumption that eMPC is robust enough to cope with the model inaccuracy was verified later using LLA, simulation, and experimental tests.

Models were identified from a relatively short open-loop measurement data sequence around the operating point (y = 76 °C, u = 65%, d = 100%) which exhibits the highest process gain. Using the prediction error method in Matlab's Identification Toolbox, two candidate nominal state-space models were obtained. The first-order discrete-time model is defined by

$$a_\mathrm{m} = 0.9742,\ b_\mathrm{m} = -0.009121,\ c_\mathrm{m} = 2 \qquad (\text{eqmodel1})$$

and the second-order model reads

$$\mathbf{A}_\mathrm{m} = \begin{bmatrix} 0.3934 & 0 \\ 0 & 0.9634 \end{bmatrix},\ \mathbf{b}_\mathrm{m} = \begin{bmatrix} -0.006895 \\ -0.009435 \end{bmatrix},\ \mathbf{c}_\mathrm{m} = \begin{bmatrix} -0.9064 & 1 \end{bmatrix} \qquad (\text{eqmodel2})$$

both sampled at T_s = 2 s. Model validation in Figure figModel shows good agreement between the open-loop process response y and open-loop simulations with both models around the designated operating point y_r = 78 °C. The difference between the two candidate models is hardly visible, therefore both were subsequently used in controller design. Note also that the difference between the modelled and the measured response is considerable in the operating area away from the designated operating point, which is due to the considerable decrease in the process gain at valve positions u > 80%.

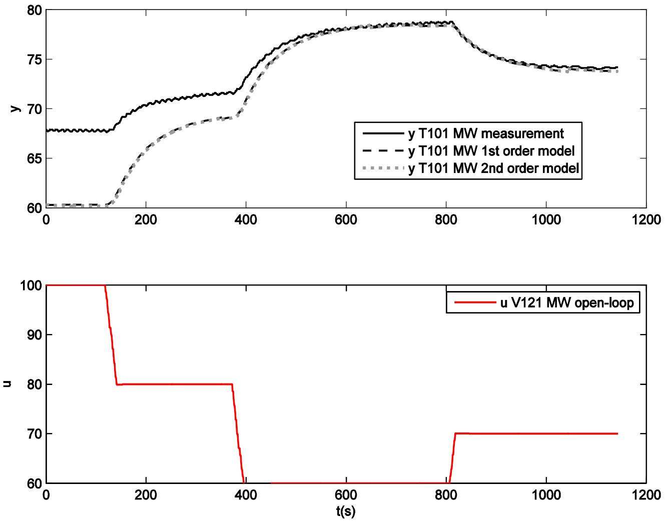

Fig. figModel. Model cross-validation

eMPC Control

Several tracking eMPC controllers with different model structure were designed and tested. Each of them was first tuned using local linear analysis (LLA) [bibGP09], then tested by simulation, and finally assessed experimentally.

The tuning approach was based on LLA of the "unconstrained" region (the region with no active constraints). LLA is based on computing the relevant transfer functions of the closed-loop system by combining the equations of the model, the controller, and the Kalman filter. LLA for the unconstrained region does not require parametric solution of the mpQP problem. However, once the parametric solution is computed, LLA is also valuable for supplementary[3] analysis of constrained regions. LLA results below are presented in the form of sigma diagrams, but root-locus diagrams were also used in tuning, and other forms of linear analysis are available as well.

Controller design is based on the nominal model, but it also allows robustness analysis with a set of different "true" models. A relatively short sampling time T_S = 2 s had to be used in order to achieve disturbance-rejection properties competitive with those of the original PID controller. A relatively long prediction horizon N is required for effective handling of output constraints, and a short N_u tends to

[3] LLA is strictly valid only when the process dwells in a certain region for a sufficiently long time.

restrict control performance. Therefore, the horizons were chosen as long as feasible for reasonably short partition computation. The tuning objectives were near-critical damping of dominant poles and well-behaved responses to the reference signal and disturbance signals at input, load, and output. Covariance estimates from open-loop experimental runs were used for KF tuning initially. However, the disturbance models were not reliable due to the short experimental data sequences. The final values were chosen using the observer concept where the diagonal elements of the KF covariance matrices were treated as tuning parameters. The design parameters of the resulting controllers are summarised in Table TAB_eMPC_par. As an illustration, Figure figeMPCxs shows projected three-dimensional cross-sections of four-dimensional polyhedral partitions of eMPC1 (top) and eMPC4 (bottom), both with y_r fixed at 78 ºC. These controllers are further discussed in the following subsections.

Table TAB_eMPC_par**.** eMPC controllers

Controller	Model order	N	N_u	R_{du}	$\mathbf{Q}_K$	Integrator position	Output constraints	Regions*
eMPC1	1	41	4	0.03	diag([10 100])	output	none	137
eMPC2	2	27	3	0.01	diag([0 0 1000])	output	none	89
eMPC3	2	33	3	0.01	$\mathbf{b}_a\mathbf{b}_a^T \cdot 10^4$ +diag([0 0 10^3])	input	none	84
eMPC4	1	41	4	0.03	diag([10 100])	output	$y \leq 79$	1048
eMPC5	1	33	3	0.005	diag([1 100])	input	none	63

All controllers: $R_y = 1$, $R_K = 1$, input constraints $0 \leq u \leq 100$, $-c_{du} \leq du \leq c_{du}$, $c_{du} = T_S \cdot 100/120$ s

* The number of regions is dependent on the chosen range of Θ.

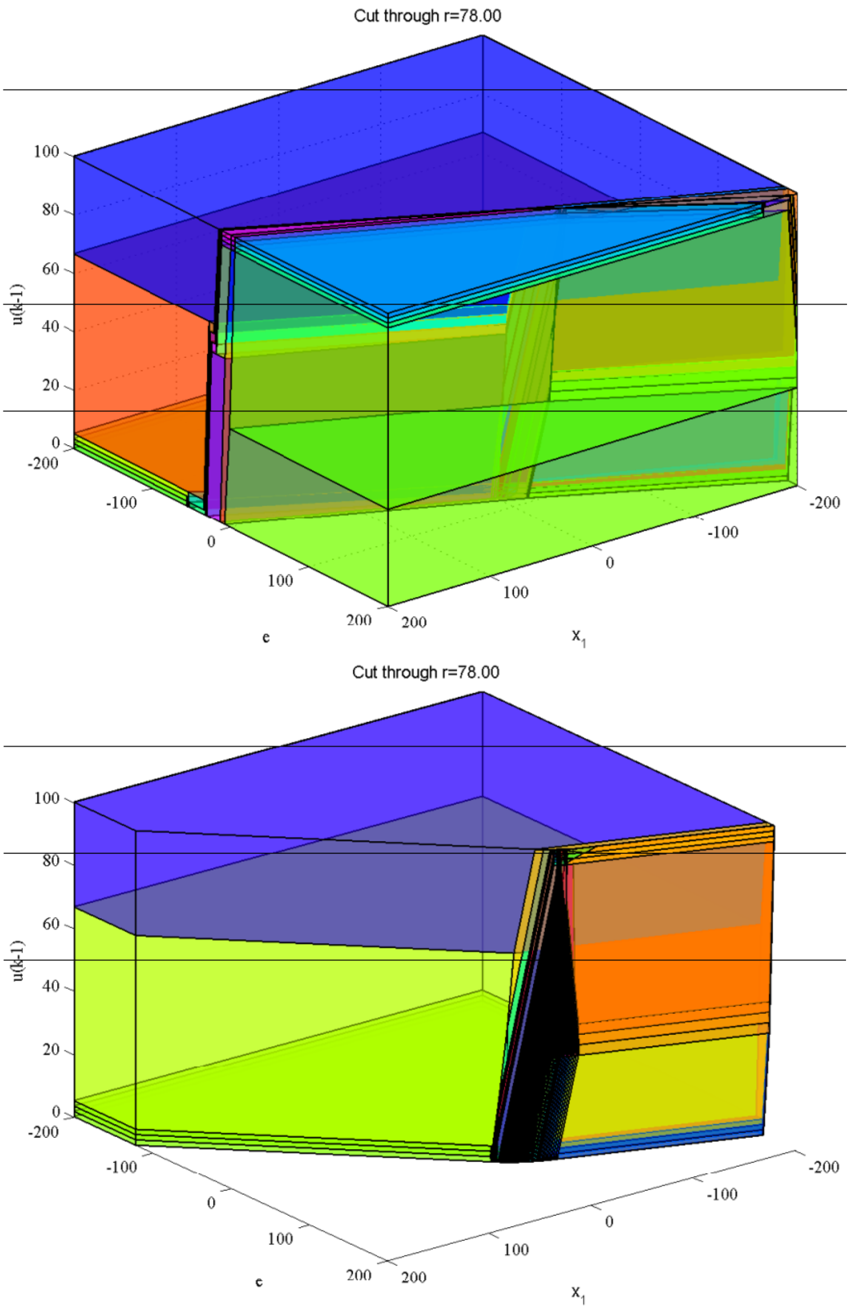

Fig. figeMPCxs**.** eMPC partition cross-sections (top: eMPC1; bottom: eMPC4)

eMPC Control without Output Constraints

In the first step, we examine the overall feedback performance of eMPC controllers where output constraints are not considered in the formulation of the MPC control law, and also study the effect of the model order choice.

The controller eMPC1 is based on the first-order model (eqmodel1), using the disturbance model with the integrator at the output. The simulated performance is shown in Figure figeMPC1_sim, while the result of the plant experiment is presented in Figure figeMPC1_exp. A similar experiment as in Figure figPI_exp was performed. The simulation results are in good agreement with the performance of the real system, with the exception of the response to the disturbance signal *d*. This difference is due to valve saturation – in this attempt, the disturbance amplitude in the experiment was too small. The results may also be compared with time-responses computed using LLA. LLA set-point and output disturbance responses match the simulation results, whereas LLA responses to input disturbances and *d* differ slightly because they do not consider signal rate limits. Compared to the performance in Figure figPI_exp, there is much less variance in *u* due to better suppression of measurement noise, while tracking and disturbance responses are comparable in this case.

Second-order models produce a slightly better estimation fit in model identification. LLA and the simulation results indicate that eMPC controllers using the second-order model structure may yield better disturbance-rejection performance, but they have not been verified experimentally[4]. Figure figeMPC2_sim shows the performance of the controller eMPC2 using the pure output-disturbance model in simulation. Figure figeMPC3_sim shows the simulation with the eMPC3 controller using the disturbance model with the integrator at the input, where KF tuning is inspired by the Loop Transfer Recovery approach [bibMac89]. The simulated performances with eMPC2 and eMPC3 are very similar, with eMPC3 having a slight advantage in terms of rejection of input and load disturbances. Compared to eMPC1, both produce faster response to input disturbances and also faster tracking performance, however there is a slight tracking overshoot.

[4] Experiments with second-order models could not be carried out in the first experimental round due to numerical problems with the mpQP algorithm. After the issue was resolved, simulation analysis indicated that another set of time-consuming experiments is not justified. The simulations predict the abovementioned improvements in overall feedback performance, but do not predict a significant improvement in performance near *y* constraints compared to the original PI control.

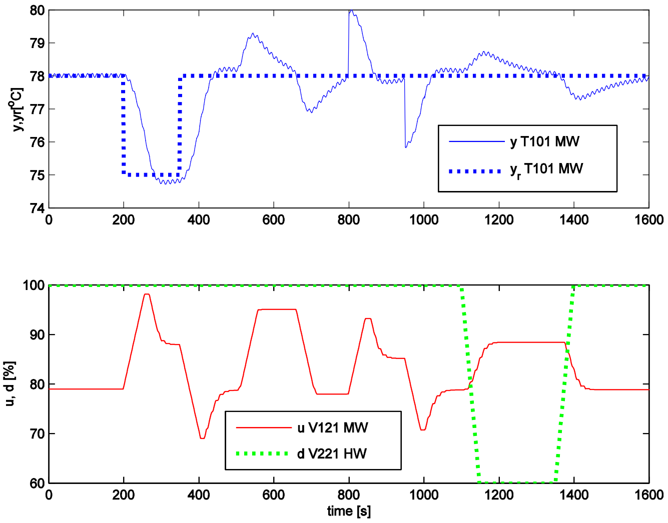

Fig. figeMPC1_sim. eMPC1 controller, simulation

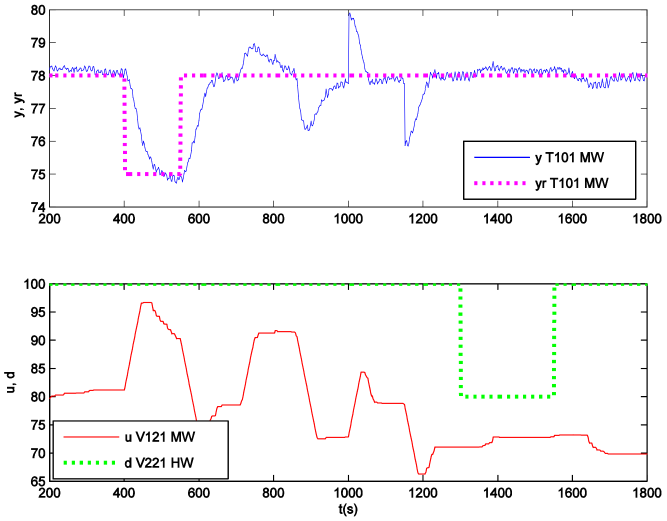

Fig. figeMPC1_exp. eMPC1 controller, experiment

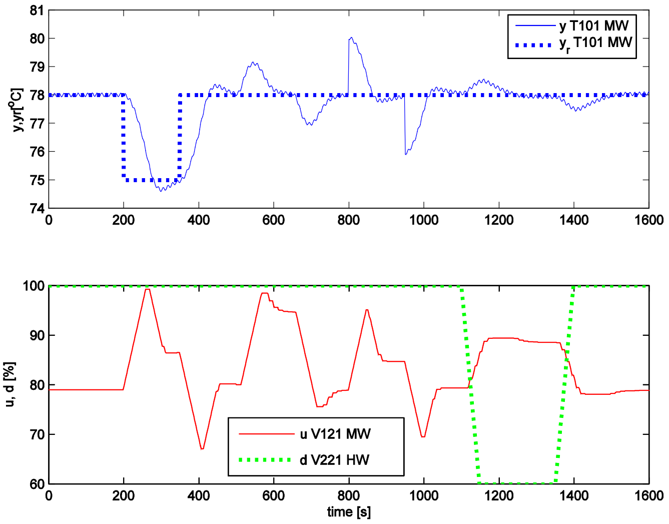

Fig. figeMPC2_sim**.** eMPC2 controller, simulation

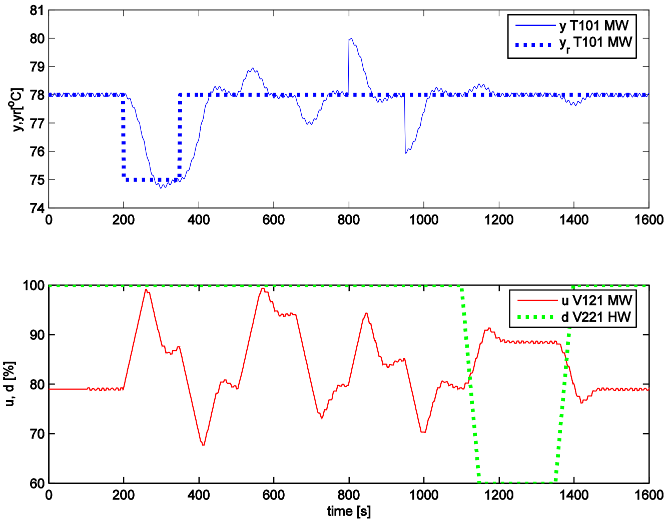

Fig. figeMPC3_sim**.** eMPC3 controller, simulation

Figure figeMPC_sigma compares sigma diagrams of the eMPC controllers eMPC1 (top-left), eMPC2 (top-right), eMPC3 (bottom-left), and eMPC5 (bottom-right). The sigma diagrams display the sensitivity functions $S(\omega)$ and $T(\omega)$ for the unconstrained region of the controllers. The corresponding transfer functions are computed from closed-loop system formulae for the system comprising the process model and the tracking controller [bibGP09]. The *sensitivity function* $S(\omega)$ is the transfer function from the measurement noise signal v to the noisy measurement y, while the complementary sensitivity function $T(\omega)$ is the transfer function from v to the noise-free output y_{nf}; more details on sensitivity functions can be found in [bibGSB08] and [bibMac02]. The solid and dashed lines represent the nominal $T(\omega)$ and $S(\omega)$ for the nominal model, while the dotted lines show a family of sensitivity functions obtained with a set of modified true models, serving as an indication of robustness to the anticipated modelling errors. The latter include modified process gain and an additional sample of computational delay.

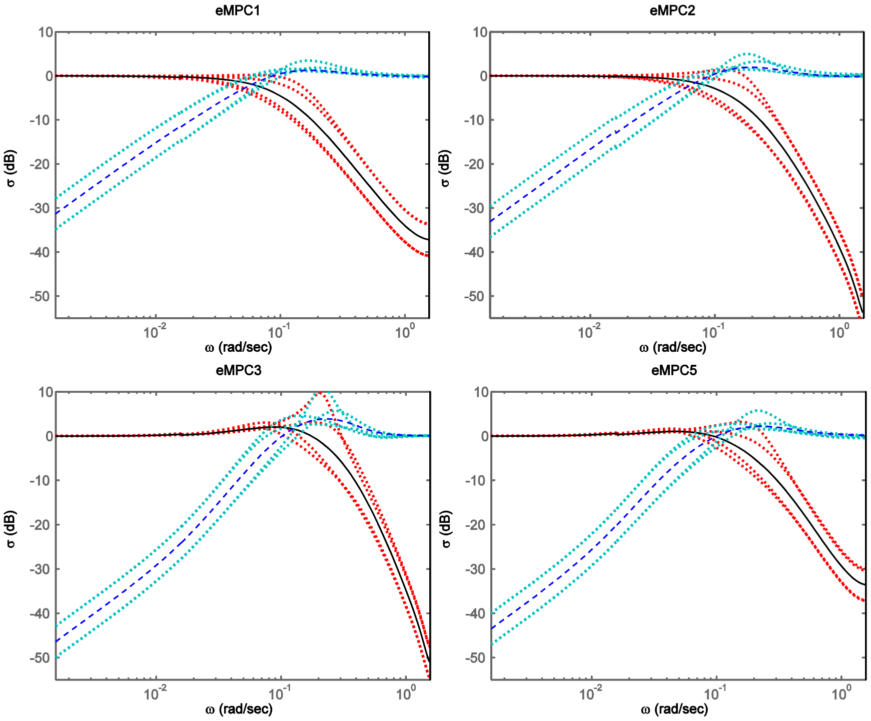

Fig. figeMPC_sigma. Sigma diagrams of nominal sensitivity functions $S(\omega)$ (dashed) and $T(\omega)$ (solid); $S(\omega)$ and $T(\omega)$ with a set of true models (dotted). Top-left: eMPC1; top-right: eMPC2; bottom-left: eMPC3; bottom-right: eMPC5.

In sigma diagrams, high bandwidth of $T(\omega)$ at medium frequencies (MF) indicates a fast tracking response. Fast roll-off of $T(\omega)$ at high frequencies (HF) indicates efficient suppression of measurement noise. A peak in the MF range of $T(\omega)$ is a sign of tracking overshoot. Low $S(\omega)$ in the low frequency range (LF) indicates efficient rejection of input disturbances. A peak in the MF range of $S(\omega)$ suggests an underdamped input-disturbance response. For example, one can notice lower $S(\omega)$ at LF with eMPC3 and eMPC5, which are both based on the disturbance model structure with the integrator at the input. Also, faster HF roll-off in

$T(\omega)$ may be noticed with eMPC2 and eMPC3, which are both based on the second-order model.

eMPC Control with Output Constraints

We now investigate the ability of eMPC to suppress disturbance transients that push the output temperature y above the set-point 78 ºC to more than the forced cooling threshold value 80 ºC by applying output constraints in the eMPC controller. The disturbances that are the most relevant in this case have slow dynamics, as they are mostly caused by V221 and other valves in the heating network. Therefore, the system response to disturbances at d is the most important. However, we focused upon very similar disturbances at u due to their better repeatability in experiments.

Figure figyconstr_exp compares responses to step-wise disturbances injected at u, starting at 200 s and 350 s, in three plant experiments:

- The solid line shows the response of the controller eMPC1, which does not consider output constraints.
- The dashed line refers to the response with the controller eMPC4 with the same tuning parameters as eMPC1, but with the additional hard output constraint $y \leq 79$, which is otherwise violated without constraints handling. However, the difference in the peak value of y compared to eMPC1 is minimal, and setting the constraint any lower results in its violation and therefore the infeasibility of the control problem.
- The dash-dot line shows the response with the controller eMPC5 based on a different disturbance model with the disturbance integrator at the process input, with no handling of output constraints. Suppression of input disturbances is improved considerably.

The effect of the output constraints may be further investigated by examining the open-loop predictions in the samples immediately following the step disturbances and the components of the MPC value function J^*.

With a tightly-tuned univariable MPC controller (with very low R_{du}), J^* is dominated by the tracking cost, which penalises excursions of y from the set-point y_r in the least-squares sense. Therefore, it also serves the purpose of avoiding y constraints almost as much as possible. In theory, by applying the y constraints it might be possible to transform the disturbance response so that the peak does not violate the constraints despite an overall increase in the tracking cost. In practice, this is difficult to achieve because of the restricted control horizon and because predictions of unmeasured disturbances may not be accurate.

With an MPC controller detuned for slower response (with higher R_{du}), J^* includes a more considerable portion of the control move cost. With such tuning, the optimum obtained without considering the y constraints is likely to violate the limit due to the slow controller reaction to the disturbance. The constrained optimum may provide a feasible solution with a faster controller response, at a higher J^*

value due to a higher control move cost. Therefore, less aggressive tuning may act as a "crush zone" for the *y* constraints, and the controller may react to a disturbance much faster in case the predicted values of *y* approach the constraints than it would otherwise.

Due to competition with the original PID control, our eMPC controllers are tuned relatively tightly. In Figure figyconstr_exp a slightly shorter delay in the response of *u* to the disturbance at 200 s can be noticed with eMPC4 than with eMPC1, and there is also a slight reduction in the *y* excursion. However, the performance improvement due to the improved disturbance model of eMPC5 is considerably more noticeable in our case.

A practical eMPC implementation would need to address the infeasibility problem in case of a violation of the output constraints. For applications where output constraints are user-specified values rather than physical constraints, soft output constraints are appropriate. With soft constraints, suitably penalised violation cost terms are added to the cost function *J*, instead of actually including the corresponding constraints in the CFTOC problem formulation. However, a straightforward implementation of soft constraints tends to increase the computational complexity over the feasible limit. Implementation is possible with the following complexity-reduction measures [bibGer11]:

- move blocking, for the sake of improved conditioning of computation;
- sparse placement of output constraints over the prediction horizon, in order to decrease the number of regions and avoid large numbers of very small regions (*nano-regions*);
- numerical improvement of the mpQP algorithm.

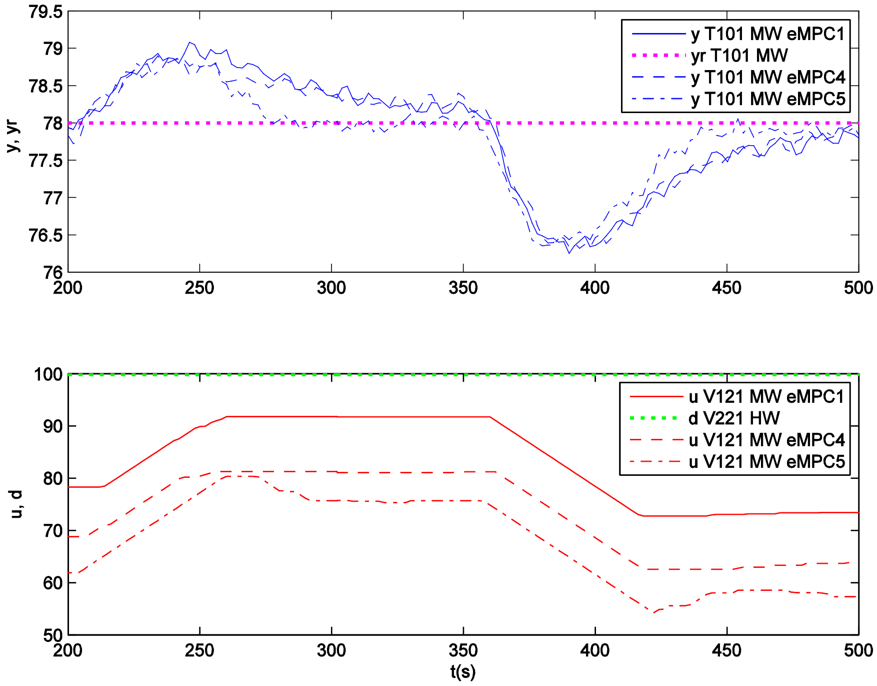

Fig. figyconstr_exp. Suppression of step-disturbances at the process input by controllers eMPC1 (solid), eMPC4 (dashed), and eMPC5 (dash-dot), experiment

Problems and limitations in applying eMPC

Crush zone. MPC enables improved control performance in the presence of constraints on process signals, compared to conventional controllers. However, when setting constraints that the controller should respect, one must also mind the "crush zones". Crush zones are spare degrees of freedom in the optimisation of the MPC cost-function that the controller may use when approaching constraints. Such crush zones may be in the form of:

a) redundant control inputs, which may occasionally deviate from the optimal or default value;
b) less important controlled outputs, which may be temporarily dropped from the control objective;
c) less aggressive tuning, which may be tightened in an emergency;
d) soft constraints, which may be used to find an acceptable compromise among several undesirable violations by specifying appropriate penalties in violation costs. However, in the absence of feasible alternatives, soft

constraints may merely prevent the occurrence of control-problem infeasibility without reducing constraint violations.

Therefore, the constraints-handling advantage of MPC is most notable with multivariable processes, where items a) and b) are available. With a relatively tightly tuned univariable process, such as the one presented here, almost no such crush zone is available. That is why only a slight performance improvement is achievable by constraints handling.

Tuning and disturbance models. Certain performance improvements with eMPC are achievable by tuning the feedback action for efficient and robust feedback performance. The appropriate methods are mostly available as the legacy of LQG control. The output-step-disturbance approach of the traditional industrial MPC methods typically offers simple design for robust but not the most efficient feedback performance. The feedback performance of an MPC controller is based on the structure and the parameters of the disturbance model (for unmeasured disturbances). Estimation methods for automated tuning of disturbance model parameters are available [bibORR06]. However, the controller must typically ensure a reasonable response to several different types of disturbances and also to set-point changes, and maintain them for a certain range of modelling errors; therefore, tuning is often a study in the art of compromise.

Partition size. The number of local controllers in the partition corresponds to all feasible combinations of the constraints. In on-line MPC, this is not a relevant consideration, as the regions need not be stored (although this may affect calculation time with a QP solver). With eMPC this is very important; not only due to storage requirements, but also because the on-line region search algorithm may take longer than on-line QP calculation for sizeable problems [bibZJM11]. It is important to keep the number of parameters and decision variables low, and place constraints sparingly.

Short-sightedness. Relatively fast sampling is needed for efficient disturbance rejection at the bottom level of the control hierarchy. In order to keep the partition size manageable with eMPC, one cannot afford long horizons. The two requirements together typically result in short-sighted controllers unable to account for output constraints violations, which are likely to occur after the end of the prediction horizon. Infinite horizon MPC algorithms may help construct a meaningful cost function with short N and have favourable theoretical properties, but they do not help to react early to disturbances that lead to constraints violations. However, the short-sightedness problem may be worked around by using blocking techniques for decision variables and by sparse placement of output constraints.

Tracking implementation. With eMPC, special care is required in order to make the off-line computational demand manageable. For example, variable reference signals increase computational complexity. Therefore, there is an advantage in handling fixed reference signals with coordinate substitutions, and in handling reference signals, which have a small number of discrete positions, via controller switching. Furthermore, various offset-free tracking implementations are possible, with two main approaches being *tracking-error integration* and *disturbance estimation*. The former may allow simpler implementation, but the resulting performance may be suboptimal.

Partitioning reliability. The available academic multi-parametric quadratic programming algorithms [bibKGB05, bibBem06] used to compute the explicit MPC solutions are not well-suited to practical tracking controllers, and may in some cases produce incomplete controller partitions. Problems appear in terms of poorly conditioned computations and in growing numbers of increasingly small regions (nano-regions) in specific partition areas. They stem from dense placement of constraints due to short sampling time and from the increased number of mpQP parameters that appear in tracking controllers due to model augmentation. Problem conditioning may be improved by applying move-blocking and by sparse placement of state/output constraints over the prediction horizon. To improve the reliability of partitioning, systematic treatment of numerical issues and degeneracies arising in several parts of the mpQP solver and its sub-functions is required [bibGer11]. This includes the choice of external LP and QP solvers and their threshold values.

It should be pointed out that QP solver reliability may also be an issue in online MPC, and that the problems may not be detected until the state enters a specific problematic area. For example, the QP solver of Matlab's Optimization Toolbox is prone to failing to find an existing feasible solution and to entering a cycling loop which never converges to the correct set of active constraints.

Conclusions

The experimental case study showed that eMPC is implementable for applications at the physical control level, which are currently dominated by PID control. However, eMPC's systematic constraints handling ability may not result in an obvious improvement regarding violations of output constraints if a suitable degree of freedom is not provided.

In our application, certain improvements in feedback control performance were achieved using eMPC compared to PI control, primarily regarding the suppression of slow process disturbances and measurement noise. However, it must be admitted that the PI controller achieves remarkably good performance especially when taking into account its simplicity, and the improvements achieved with eMPC were not considered substantial enough to warrant a control system redesign in this application.

Acknowledgments

This work was supported in part by the EC (CONNECT, COOP-CT-2006-031638) and the Slovenian Research Agency (P2-0001). The authors are grateful for the technical assistance of INEA d.o.o. and JP CČN Domžale-Kamnik d.o.o.

References

[bibQB03] Qin SJ, Badgwell TA (2003) A survey of industrial model predictive control technology. *IEEE*, 733–764

[bibMac02] Maciejowski JM (2002) *Predictive Control with Constraints.* Harlow UK: Prentice Hall

[bibPDBBM00] Pistikopoulos EN, Dua V, Bozinis NA, Bemporad A, Morari M (2000) On-line optimization via off-line parametric optimization tools. *Computers and Chemical Engineering*, 24:183–188

[bibBMDP02] Bemporad A, Morari M, Dua V, Pistikopoulos EN (2002) The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:1:3–20

[bibPGD07] Pistikopoulos EN, Georgiadis MC, Dua V (eds) (2007) *Multi-Parametric Model-Based Control.* Weinheim: Wiley-VCH

[bibBao02] Baotić M (2002) An efficient algorithm for multi-parametric quadratic programming. *Technical Report AUT02-05*. ETH Zürich, Institut für Automatik.

[bibGBTM04] Grieder P, Borrelli F, Torrisi F, Morari M (2004) Computation of the constrained infinite time linear quadratic regulator. *Automatica*, 40:701–708

[bibTJB03a] Tøndel P, Johansen TA, Bemporad A (2003a) An algorithm for multiparametric quadratic programming and explicit MPC solutions. *Automatica*, 39:489–497

[bibTJB03b] Tøndel P, Johansen TA, Bemporad A (2003b) Further results on multi-parametric quadratic programming. *Proc. 42th IEEE CDC*, Hawaii, 3173–3178

[bibTJB03c] Tøndel P, Johansen TA, Bemporad A (2003c) Evaluation of piecewise affine control via binary search tree. *Automatica*, 39:945–950

[bibMR03] Mayne DQ, Raković SV (2003) Optimal Control of Constrained Piecewise Affine Discrete-Time Systems. *Journal of Computational Optimization and Applications*, 25:167–191

[bibSKJTJ06] Spjøtvold J, Kerrigan EC, Jones CN, Tøndel P, Johansen TA (2006) On the facet-to-facet property of solutions to convex parametric quadratic programs. *Automatica*, 42:2209–2214

[bibSTJ07] Spjøtvold J, Tøndel P, Johansen TA (2007) A continuous selection and unique polyhedral representation of solutions to convex multiparametric quadratic programs. *J. Optimization Theory and Applications*, 134:177–189

[bibJM06] Jones CN, Morari M (2006) Multiparametric linear complementarity problems. *Proc. 45th IEEE Conference on Decision and Control*, 2006

[bibGSB08] Gerkšič S, Strmčnik S, van den Boom TJJ (2008) Feedback action in predictive control: an experimental case study. *Control Engineering Practice*, 16:321–332

[bibGP09] Gerkšič S, Pregelj B (2009) Disturbance rejection tuning of a tracking multi-parametric predictive controller. *Proc. IEEE International Conference on Industrial Technology*, Gippsland, 65–70

[bibKGB05] Kvasnica M, Grieder P, Baotić M (2005) *Multi-parametric toolbox* [Online]. Available at: http://control.ee.ethz.ch/ mpt/

[bibBem06] Bemporad A (2006) *Hybrid toolbox for real-time applications, user's guide*. University of Siena, technical report

[bibZJM11] Zeilinger MN, Jones CN, Morari M (2011) Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Optimization. *IEEE Transactions on Automatic Control*, 56:7:1524–1534.

[bibGJK04] Grancharova A, Johansen TA, Kocijan J (2004) Explicit model predictive control of gas-liquid separation plant via orthogonal search tree partitioning. *Computers and Chemical Engineering*, 28:2481–2491

[bibPG08] Pregelj B, Gerkšič S (2008) Tracking implementations in multi-parametric predictive control. In: Cunha JB (ed.). *Proc. 8th Portuguese Conference on Automatic Control CONTROLO 2008*, Vila Real: Universidade de Trás-os-Montes e Alto Douro, 944–949

[bibMB02] Muske KR, Badgwell TA (2002) Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12:617–632

[bibPR03] Pannocchia G, Rawlings JB (2003) Disturbance models for offset-free model predictive control. *AIChE Journal*, 49:426–437

[bibGAM08] González AH, Adam EJ, Marchetti JL (2008) Conditions for offset elimination in receding horizon controllers: A tutorial analysis. *Chemical Engineering and Processing*, 47:2184–2194

[bibKRM10] Kvasnica M, Rauova I, Fikar M (2010) Automatic Code Generation for Real-Time Implementation of Model Predictive Control. *Proc. IEEE International Symposium on Computer-Aided Control System Design*, Yokohama, 8-10 September 2010, 993–998

[bibMR93] Muske KR, Rawlings JB (1993) Model predictive control with linear models. *AIChE Journal*, 39:262–287

[bibSak03] Sakizlis V (2003) *Design of model based controllers via parametric programming*. PhD thesis, Imperial College, London

[bibSDPP04] Sakizlis V, Dua V, Perkins JD, Pistikopoulos EN (2004) Robust model-based tracking control using parametric programming. *Computers and Chemical Engineering*, 28:195–207

[bibPLR06] Pannocchia G, Laachi N, Rawlings JB (2006) A candidate to replace PID control: Siso-constrained lq control. *AIChE Journal*, 51:1178–1189

[bibRR99] Rao CV, Rawlings JB (1999) Steady states and constraints in model predictive control. *AIChE Journal*, 45:1266–1278

[bibLAAC08] Limon D, Alvarado I, Alamo T, Camacho EF (2008) MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44:2382–2387

[bibVPS99] Vrančić D, Peng Y, Strmčnik S (1999) A new PID controller tuning method based on multiple integrations. *Control Engineering Practice*, 7:5:623-633

[bibMac89] Maciejowski JM (1989) *Multivariable Feedback Design*. Wokingham UK: Addison-Wesley

[bibORR06] Odelson BJ, Rajamani MR, Rawlings JB (2006) A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42:303–308

[bibGer11] Gerkšič S (2011) Improving reliability of partition computation in explicit MPC with MPT Toolbox. *Proc. 18th IFAC World Congress 2011*, Milano IT, 9260–9265.

[bibYALMIP04] Löfberg J (2004) YALMIP: A Toolbox for Modeling and Optimization in MATLAB. *Proc. CACSD Conference*, Taipei. Available at: http://users.isy.liu.se/johanl/yalmip

[bibTJB03c] Tøndel P, Johansen TA, Bemporad A (2003) Evaluation of piecewise affine control via binary search tree. *Automatica*, 39:5:945-950

Abstract

In explicit model predictive control (eMPC), the bulk of the computational load of classic MPC is performed during the off-line design stage, which enables the controller to be implemented on standard industrial automation equipment and for processes with fast dynamics, however the computational complexity restricts applicability to small-scale control problems. The approach is appealing for a niche of control applications, but practical applications have been scarce so far. We describe one possible offset-free tracking setup that allows implementation of eMPC with relatively fast sampling and reasonably long horizons for practical applications. The applicability of eMPC is illustrated by an experimental case study of cooling-water temperature control in a biogas-fuelled combined-heat-and-power production unit, where eMPC replaces a pre-existing single-loop PID controller, with the aim of reducing unnecessary excursions of the cooling water temperature away from its set-point in the critical range near output constraints. eMPC controllers were designed using local linear analysis and tested both on a simplified simulation model and experimentally on the CHP unit. The performance improvements due to tuning of eMPC feedback action and due to the constraints-handling ability were examined, and several implementation issues related to the practical implementation of eMPC are discussed.