Estimation technique for offset-free economic MPC based on modifier adaptation

Marco Vaccari * **Federico Pelagagge** * **Dominique Bonvin** ** **Gabriele Pannocchia** *,***

* *University of Pisa, Department of Civil and Industrial Engineering, Largo Lazzarino 2, 56126 Pisa (Italy)*
** *Laboratoire d'Automatique, EPFL, CH-1015, Lausanne, Switzerland*
*** *Corresponding author: gabriele.pannocchia@unipi.it*

Abstract: Economic model predictive control formulations that combine online optimizing control with offset-free methodologies such as modifier adaptation have been proposed recently. These new algorithms are able to achieve asymptotic optimal performance despite the presence of plant-model mismatch. However, there is a major requirement stemming from the modifier-adaptation part, namely, the necessity to know the static plant gradients at the sought (and therefore still unknown) steady-state operating point. Hence, for implementation purposes, the algorithms need to be enhanced with plant gradient estimation techniques. This work proposes to estimate modifiers directly, based on steady-state perturbations and using Broyden's approximation. The proposed economic MPC algorithm has been tested in simulation on the Williams-Otto reactor and provides plant optimality upon convergence.

Keywords: Economic model predictive control; Real-time optimization; Modifier adaptation; Gradient estimation; Broyden's method.

1. INTRODUCTION

In the last decade, economic MPC (eMPC) has become one of the most studied solution method to overcome the hierarchical separation between economic optimization and control in the process industries (Rawlings et al., 2012; Ellis and Christofides, 2014). However, as with model predictive control (MPC) and model-based real-time optimization (RTO), eMPC typically does not converge to plant optimality in the presence of plant-model mismatch. In order to deal with this problem, novel offset-free eMPC (OF-eMPC) algorithms have been proposed in the literature (Vaccari and Pannocchia, 2016; Pannocchia, 2018; Faulwasser and Pannocchia, 2019), by merging offset-free tracking MPC (Pannocchia, 2015) with modifier adaptation (MA) (Marchetti et al., 2009). The purpose of these formulations is to enforce that the steady state reached by the closed-loop system corresponds to the best equilibrium point for the plant. These algorithms rely on the demonstrated MA feature that first-order modifier terms can guarantee meeting the necessary conditions of optimality of the unknown plant (Marchetti et al., 2009). A particular version of MA that puts modifiers on the outputs, labeled "output MA" (MAy), is used in this study (Papasavvas et al., 2019). The major drawback of both MA and MAy is the requirement of plant gradients at steady-state conditions. This is not a trivial task, and much work in the literature has dealt with gradient estimation (see e.g. Costello et al. (2016); François and Bonvin (2013); Marchetti et al. (2016) and references therein). A recent implementation of OF-eMPC with gradient estimation is described in (Vaccari and Pannocchia, 2018). The authors detail a new method based on system identification to compute a gradient approximation to be used in MAy or other offset-free structures. Although the results are quite promising, one should keep in mind that system identification requires sufficient system excitation and also the ability to process noisy measurements. For this reason, an important contribution of the present work concerns a way of estimating first-order modifiers directly from measurements using a Broyden's update that relies on past operating points rather than local perturbations.

The paper is organized as follows. The problem definition and an OF-eMPC algorithm available in the literature are detailed in Section 2. The proposed gradient estimation technique is presented in Section 3. The resulting OF-eMPC algorithm is tested on a simulated example in Section 4. Finally, conclusions are presented in Section 5.

2. PROBLEM DEFINITION

2.1 Plant and cost specifications

Discrete-time nonlinear dynamic systems are the object of the current study:

$$\begin{aligned} x_p^+ &= f_p(x_p, u) \\ y_p &= h_p(x_p) \end{aligned} \tag{1}$$

where $x_p \in \mathbb{R}^{n_{x_p}}$, $u \in \mathbb{R}^{n_u}$ and $y_p \in \mathbb{R}^{n_y}$ are the plant states, inputs and outputs, respectively, x_p^+ are the successor states. At each time $k \in \mathbb{Z}$, the plant outputs y_p are measured and denoted by $y_{p,k}$. We assume differentiability of the functions $f_p : \mathbb{R}^{n_{x_p}} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_{x_p}}$ and $h_p : \mathbb{R}^{n_{x_p}} \to \mathbb{R}^{n_y}$.

Input and output bounds must be fulfilled at all times:

$$u_{\min} \le u \le u_{\max}, \quad y_{\min} \le y \le y_{\max} \tag{2}$$

where $u_{\min}, u_{\max}, y_{\min}$ and $y_{\max}$ are the corresponding bounds.

The economically optimal steady state of Plant (1) is defined by solving the following problem:

$$(\bar{x}_p^\star, \bar{u}^\star, \bar{y}_p^\star) = \arg\min_{x,u,y} \ell_e(y,u) \tag{3a}$$

subject to

$$x = f_p(x,u) \tag{3b}$$
$$y = h_p(x) \tag{3c}$$
$$u_{\min} \le u \le u_{\max} \tag{3d}$$
$$y_{\min} \le y \le y_{\max} \tag{3e}$$

where $\ell_e : \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is the cost function and $(\bar{x}_p^\star, \bar{u}^\star, \bar{y}_p^\star)$ is the optimal equilibrium point of Plant (1).

Assumption 1. The cost function $\ell_e(y,u)$ is continuously differentiable.

It is important to underline that, even if Problem (3) is assumed to be feasible and its solution unique, $(\bar{x}_p^\star, \bar{u}^\star, \bar{y}_p^\star)$ is unknown due to plant-model mismatch as discussed next.

2.2 Model and augmented model

In order to design an eMPC algorithm, the following nominal process model is used:

$$\begin{aligned} x^+ &= f(x,u) \\ y &= h(x) \end{aligned} \tag{4}$$

where x and $x^+ \in \mathbb{R}^{n_x}$ denote the current and successor states. The functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$, which are assumed to be differentiable, differ from their plant counterparts f_p and h_p due to plant model-mismatch.

The general form of a (linearly) augmented model for offset-free tracking MPC algorithms is (Pannocchia et al., 2015):

$$\begin{aligned} x^+ &= F(x,u,d) = f(x,u) + B_d d \\ d^+ &= d \\ y &= H(x,d) = h(x) + C_d d \end{aligned} \tag{5}$$

where $d \in \mathbb{R}^{n_d}$ are the so-called *disturbances*, $B_d \in \mathbb{R}^{n_x \times n_d}$ and $C_d \in \mathbb{R}^{n_y \times n_d}$ are matrices used to model the disturbance effects. The functions $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_x}$ and $H : \mathbb{R}^{n_x} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_y}$ are assumed to be continuous. The following assumption regarding system observability is imposed on the $(n_x + n_d)$ augmented states (Pannocchia et al., 2015).

Assumption 2. The augmented system (5) is observable.

It has to be noted that Assumption 2 imply the observability of the nominal model (4) (Pannocchia et al., 2015, Remark 8).

2.3 State and disturbance estimation

Let $\hat{x}_{k-1}^*$ and $\hat{d}_{k-1}^*$ denote the estimates of x_{k-1} and d_{k-1} obtained using the output measurements at time $k-1$. Furthermore, $\hat{x}_k, \hat{d}_k$ and $\hat{y}_k$, are the predicted values of x_k, d_k and y_k obtained at time k using the augmented model (5) and the best available values of the states, inputs and disturbances at time $k-1$, that is:

$$\begin{aligned} \hat{x}_k &= F(\hat{x}_{k-1}^*, u_{k-1}, \hat{d}_{k-1}^*) \\ \hat{d}_k &= \hat{d}_{k-1}^* \\ \hat{y}_k &= H(\hat{x}_k, \hat{d}_k) \end{aligned} \tag{6}$$

Hence, defining the prediction errors at time k as:

$$\varepsilon_k = y_{p,k} - \hat{y}_k \tag{7}$$

the filtering relations for the augmented states can be written as:

$$\begin{aligned} \hat{x}_k^* &= \hat{x}_k + K_x \varepsilon_k \\ \hat{d}_k^* &= \hat{d}_k + K_d \varepsilon_k \end{aligned} \tag{8}$$

where the matrices $K_x \in \mathbb{R}^{n_x \times n_y}$ and $K_d \in \mathbb{R}^{n_d \times n_y}$ are chosen to form an asymptotically stable observer, which requires $n_y = n_d$ and K_d to be invertible (Pannocchia et al., 2015).

2.4 Computation of modifiers

We will use the concept of output modifiers that is borrowed from the RTO literature (Marchetti et al., 2009; Papasavvas et al., 2019). In this study, these modifiers express the differences between the plant and model outputs at steady state. Let $\Lambda_k \in \mathbb{R}^{n_y \times n_u}$ be the modifier matrix at time k. This matrix, initialized as $\Lambda_0 = 0$, is updated at each decision time as follows:

$$\Lambda_k = (1-\sigma)\Lambda_{k-1} + \sigma\left(\mathrm{D}_u g_p\left(\bar{u}_{k-1}\right) - \mathrm{D}_u g\left(\bar{u}_{k-1}\right)\right) \tag{9}$$

where σ is a scalar first-order filter constant $\in (0,1]$, $g_p : \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ and $g : \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ are the plant and model steady-state input-to-output maps, calculated for Plant (1) and Model (4) or (5), respectively [1]. The operator $\mathrm{D}_u(\cdot)$ represents the derivative of the considered function with respect to the variable u, and $\bar{u}_{k-1}$ the input steady-state targets computed at the previous iteration and available at iteration k (see next subsection).

2.5 Target calculation with modifiers

A steady-state target calculation is required at time k to compute the equilibrium triple $(\bar{x}_k, \bar{u}_k, \bar{y}_k)$, considering the current disturbance estimate $\hat{d}_k^*$. However, because of plant-model mismatch, the outputs are corrected using first-order modifier terms, which gives the following target problem to be solved at each iteration:

$$(\bar{x}_k, \bar{u}_k, \bar{y}_k) = \underset{(x,u,y)}{\arg\min}\ \ell_e(y,u) \tag{10a}$$

subject to:

$$x = F(x,u,\hat{d}_k^*) \tag{10b}$$
$$y = H(x,\hat{d}_k^*) + \Lambda_k\left(u - \bar{u}_{k-1}\right) \tag{10c}$$
$$u_{\min} \le u \le u_{\max} \tag{10d}$$
$$y_{\min} \le y \le y_{\max} \tag{10e}$$

The modifier terms in Eq. (10c) enforce KKT matching between Problems (3) and (10) (Papasavvas et al., 2019).

2.6 Economic MPC with modifiers

Let $\boldsymbol{x}{:=}\{\chi_0, \chi_1, \ldots, \chi_N\}$ and $\boldsymbol{u}{:=}\{\nu_0, \nu_1, \ldots, \nu_{N-1}\}$ denote some generic state and input sequences, respectively. Then, a finite-horizon optimal control problem (FHOCP), modified as per Eq. (10c), is solved at each decision time k:

$$(\boldsymbol{x}_k^\star, \boldsymbol{u}_k^\star) = \underset{\boldsymbol{x},\boldsymbol{u}}{\arg\min} \sum_{i=0}^{N-1} \ell_e(\gamma_i, \nu_i) \tag{11a}$$

subject to

$$\chi_0 = \hat{x}_k^* \tag{11b}$$
$$\chi_{i+1} = F(\chi_i, \nu_i, \hat{d}_k^*) \tag{11c}$$
$$\gamma_i = H(\chi_i, \hat{d}_k^*) + \Lambda_k(\nu_i - \bar{u}_{k-1}) \tag{11d}$$
$$u_{\min} \le \nu_i \le u_{\max} \tag{11e}$$
$$y_{\min} \le \gamma_i \le y_{\max} \tag{11f}$$
$$\chi_N = \bar{x}_k \tag{11g}$$

As usual in MPC, assuming that Problem (11) is feasible, only the first inputs of the optimal sequence $\boldsymbol{u}_k^\star$ are implemented:

$$u_k = \boldsymbol{u}_k^\star[1] \tag{12}$$

Remark 3. The modifier terms in Eq. (11d) serve the same purpose as those in Eq. (10c) and, for consistency with the latter, $\bar{u}_{k-1}$ and not $\bar{u}_k$ are used as steady-state input targets.

[1] Since a linear disturbance model is used, derivatives of Model (4) or (5) with respect to u are identical.

3. GRADIENT ESTIMATION

The OF-eMPC algorithm in Section 2 requires the knowledge of the steady-state plant gradients $\mathrm{D}_u g_p(\cdot)$. Since these gradients cannot be measured directly, they must be inferred from typically noisy and transient output measurements. Various methods have been proposed in the literature (François and Bonvin, 2013; Costello et al., 2016; de Avila Ferreira et al., 2017). These approaches can be classified as either *steady-state* perturbation methods that normally use only steady-state data, or *dynamic* perturbation methods that use transient data (Marchetti et al., 2016).

3.1 Basic idea

In this work, we propose a methodology based on steady-state perturbation methods, that is, we use steady-state data for gradient estimation. Since, for any input change, one needs to wait for the plant to reach the new steady-state conditions, these methods can be particularly slow.

Broyden's update offers a way of estimating gradients from current and past measurements (Roberts, 2000; Rodger and Chachuat, 2011; Marchetti et al., 2016). Although no additional perturbations are required, the input changes from one iteration to the next must be sufficiently exciting for the scheme to work. The technique is a standard *secant method* in nonlinear programming for updating estimates of first-order derivatives, such as Jacobian matrices (Dennis Jr and Schnabel, 1996).

The modifier terms Λ_k described in Section 2, are updated at each sampling time using transient data. However, due to the necessity of the proposed estimation algorithm to operate with steady-state measurements, the system must arrive at quasi-stationary conditions between two successive modifier updates. Hence, the interval between consecutive modifier updates should be at least of the order of the system settling time τ_{st}, which corresponds to have a time-scale separation between the control and gradient-estimation tasks. That is, modifier update is provided every M sampling times of the eMPC scheme, where $M \sim \frac{\tau_{st}}{\tau}$ is a tuning parameter and τ is the eMPC sampling time. Dedicated input perturbations can be performed to initialize the proposed scheme with a non-zero Λ_0 as a way to speed up convergence. These perturbations are required in order to collect (n_u+1) quasi-stationary output measurements.

3.2 Broyden's method for modifier estimation

We propose to update the modifier matrix every M iterations by computing the *differences* between the plant and model gradients. For this, let us define:

$$\delta g_k := \mathrm{D}_u g_{p,k} - \mathrm{D}_u g_k \tag{13a}$$
$$\Delta U_k := u_{k-1} - u_{k-M-1} \tag{13b}$$
$$\delta y_k := y_{p,k} - h(\hat{x}_k) \tag{13c}$$
$$\Delta E_k := \delta y_k - \delta y_{k-M} \tag{13d}$$

where $\mathrm{D}_u g_{p,k}$ and $\mathrm{D}_u g_k$ are the estimates available at iteration k of $\mathrm{D}_u g_p(\bar{u}_{k-1})$ and $\mathrm{D}_u g(\bar{u}_{k-1})$. One can update the gradient differences using Broyden's formula as:

$$\Delta g_k = \Delta g_{k-M} + \frac{\Delta E_k - \Delta g_{k-M}\Delta U_k}{\Delta U_k^T \Delta U_k}\Delta U_k^T \tag{14}$$

Finally, the modifier matrix is updated as follows:

$$\Lambda_k = \begin{cases} \Lambda_{k-1} & \text{if } \mod(k-k_{in},M) \neq 0 \quad (15a) \\ (1-\sigma)\Lambda_{k-1} + \sigma\Delta g_k & \text{if } \mod(k-k_{in},M) = 0 \quad (15b) \end{cases}$$

where k_{in} is the time at which the initialization starts (this is detailed in the next section). It should be noted that, when applying Eq. (14), care must be taken to avoid ill-conditioning when $\Delta U_k \to 0$. Hence, the step given by Eq. (14) is performed only if $\|\Delta U_k\| \geq \rho_u$, where ρ_u is a chosen threshold.

Remark 4. One could consider a different approach, in which the estimates $D_u g_{p,k}$ of the plant gradients are obtained via Broyden's method, while the exact model gradients $\mathrm{D}_u g(\bar{u}_{k-1})$ are used. However, experience has shown that the direct estimation of modifiers is often preferred over the estimation of the individual gradients since the plant and the model are approximated using the *same* numerical scheme. A graphical explanation can be derived similarly to the one in (Marchetti et al., 2016, Figure 3) made for linear interpolation.

Remark 5. The driving term ΔE_k of the Broyden update (14) is defined in Eq. (13d) using the errors between the plant outputs and the nominal model outputs as defined in Eq. (13c) instead of the prediction errors ε_k defined in Eq. (7) using the augmented model. Note that using $\Delta E_k = \varepsilon_k - \varepsilon_{k-M} = (y_{p,k} - H(\hat{x}_k, \hat{d}_k)) - (y_{p,k-M} - H(\hat{x}_{k-M}, \hat{d}_{k-M}))$ to update Δg_{k-M} in Eq. (14) would not work. The tacit secant equation that is behind the Broyden update, that is $\varepsilon_k - \varepsilon_{k-M} = \Delta g_k \Delta U_k$, does not hold true because $\hat{d}_k \neq \hat{d}_{k-M}$. In fact, whenever the system reaches an equilibrium (constant inputs and outputs), it follows that the prediction errors ε_k in Eq. (7) go to zero due the use of the augmented Model (5) (Pannocchia et al., 2015). Hence, since steady-state conditions are supposed to be reached after M iterations, this would give $\varepsilon_k - \varepsilon_{k-M} \approx 0$ and Δg_{k-M} would not be updated.

3.3 Algorithm initialization

To increase convergence speed, the scheme needs a good initial value for Λ, that is, a good initial value for Δg. A simple approach for this consists in perturbing each input individually around the current operating point to get an estimate of the corresponding gradient elements. Hence, this approach requires n_u input perturbations to be carried out and, for each perturbation, we must wait for steady state, that is, for M sampling times.

To implement this initialization, the closed-loop system is brought to steady state, reached at time k_{in} with the inputs $u_{k_{in}}$. Then, for $j = 1, \cdots, n_u$, the following control law is used:

$$u_k = u_{k_{in}} + s_j e_j \quad \text{if} \quad (j-1)M \leq k - k_{in} < jM \tag{16}$$

where s_j is the amplitude of a step of duration M in the direction e_j, with e_j a unit vector in input space. Hence, the term $s_j e_j$ perturbs the j^{th} component of $u_{k_{in}}$ individually during M iterations. The j^{th} column of the estimated gradient differences can be computed as:

$$\Delta g_{k_{in}+N_{in},j} = \frac{(y_{p,k_{in}+jM} - y_{p,k_{in}}) - (h(\hat{x}_{k_{in}+jM}) - h(\hat{x}_{k_{in}}))}{s_j} \tag{17}$$

which is used to compute $\Lambda_{k_{in}+N_{in}}$ in Eq. (9).

3.4 Summary of the algorithm

To detail the new algorithm, a block diagram and an algorithm are presented in Figure 1 and Algorithm 1, respectively. As can be seen in Figure 1, the gradient difference estimation is modified only after $N_{in}+M$ iterations. According to Algorithm 1, the modifier matrix is updated every M time steps, but only when the difference between two successive inputs is not too close to zero. This is needed to avoid calculating a gradient

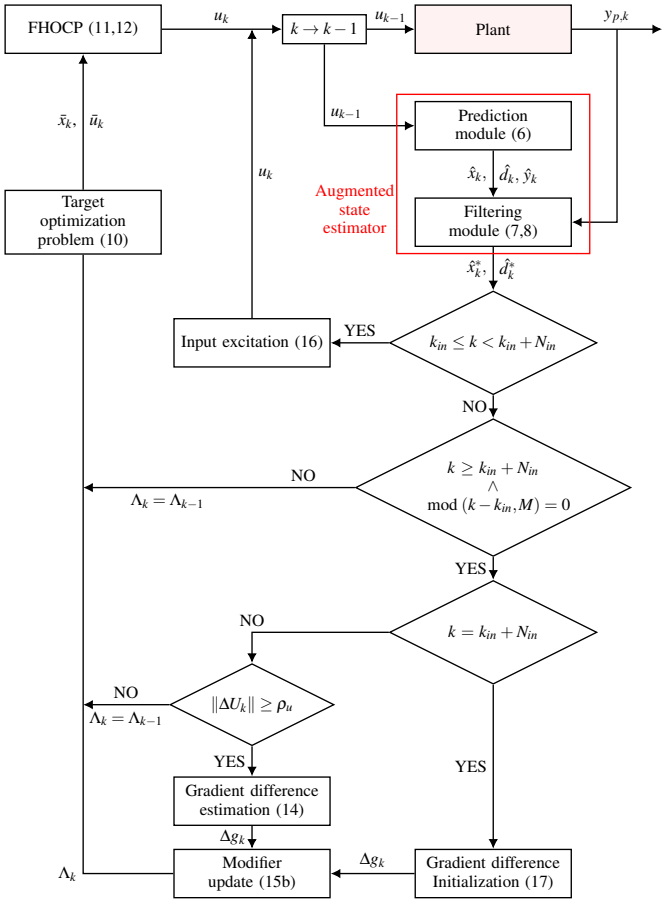

Fig. 1. Block diagram for the eMPC algorithm described in Sections 2 and 3.

difference based on not enough informative data, that is, avoid ill-conditioning of Eq. (14). Hence, the value of the parameter ρ_u can be seen also as a tuning parameter for the performance of the gradient estimation method.

4. CASE STUDY: WILLIAMS-OTTO REACTOR

This section presents a case study to validate the proposed gradient-estimation methodology. The output disturbance model is used, that is, $B_d = 0$, $C_d = \mathrm{I}$, and the estimator is a deadbeat Kalman filter with $K_x = \mathbf{0}$, $K_d = \mathrm{I}$.

The controllers studied are as follows:

- eMPC0 is the economic MPC that uses only the output disturbance model.
- eMPC1 is the offset-free economic MPC defined in Section 2 with *plant gradients assumed known*;
- eMPC2 uses the gradient differences approximation described in Section 3.2.

Moreover, the filter constant for updating the modifiers in Eq. (9) is $\sigma = 0.5$, while, unless differently specified, the number of iterations between two consecutive modifier updates is $M = 15$.

Process. The Williams-Otto reactor is a well-known process control example that is often used as a benchmark for RTO (Williams and Otto, 1960; Marchetti et al., 2016). It consists of a non-isothermal CSTR, in which the following three reactions take place:

$$\begin{aligned} \mathrm{A}+\mathrm{B} &\xrightarrow{k_1} \mathrm{C} \qquad r_1 = k_1(T_r)\, c_\mathrm{A} c_\mathrm{B} \\ \mathrm{B}+\mathrm{C} &\xrightarrow{k_2} \mathrm{P}+\mathrm{E} \qquad r_2 = k_2(T_r)\, c_\mathrm{B} c_\mathrm{C} \\ \mathrm{C}+\mathrm{P} &\xrightarrow{k_3} \mathrm{G} \qquad r_3 = k_3(T_r)\, c_C c_\mathrm{P} \end{aligned} \tag{18}$$

Species A is fed at the constant flowrate Q_A with molar concentration $c_{\mathrm{A}0}$, while species B is added at the variable flowrate Q_B with molar concentration $c_{\mathrm{B}0}$. The desired products are P and E, while C and G are intermediate and undesired products, respectively. The reactor temperature T_r is assumed manipulated, thus reflecting an ideal cooling system, while the reactor outlet flowrate Q_r is set equal to the sum of the two inlet flowrates, that is, $Q_r = Q_\mathrm{A} + Q_\mathrm{B}$. The reactor volume V_r remains constant. Moreover, only the molar concentration of the two desired products are assumed to be measured, that is, $y_p = [c_\mathrm{P}\ c_\mathrm{E}]^T$. The kinetic constants follow an Arrhenius-type law,

$$k_i(T_r) = k_{i0} \exp\left(\frac{-E_i}{T_r + 273.15}\right) \text{ for } i = 1,2,3 \tag{19}$$

The system dynamics are reported in (Faulwasser and Pannocchia, 2019). The system is characterized by two inputs, that is, $u = [Q_\mathrm{B}\ T_r]^T$.

The process economics is defined by the following running cost:

$$\ell_e(\cdot) = Q_\mathrm{A} c_{\mathrm{A}0} p_\mathrm{A} + Q_\mathrm{B} c_{\mathrm{B}0} p_\mathrm{B} - Q_r c_\mathrm{P} p_\mathrm{P} - Q_r c_\mathrm{E} p_\mathrm{E} \tag{20}$$

where $p_\mathrm{A}, p_\mathrm{B}, p_\mathrm{P}$, and p_E are the molar prices of reactants and products. The plant parameters are given in Table 1.

Algorithm 1 Offset-free eMPC algorithm

Initialize u_0, $\hat{x}^*_0$, $\hat{d}^*_0$ and Λ_0; set $k = 1$.
Input: u_{k-1}, $\hat{x}^*_{k-1}$, $\hat{d}^*_{k-1}$ and Λ_{k-1}.
Read $y_{p,k}$ from the plant.
Predict successor quantities using (6).
Evaluate prediction errors using (7).
Evaluate state and disturbance estimates using (8).
if $k_{in} \le k < k_{in} + N_{in}$ **then**
    Set u_k using (16).
else
    if mod $(k - k_{in}, M) = 0$ and $k \ge k_{in} + N_{in}$ **then**
        if $k = k_{in} + N_{in}$ **then**
            Initialize the gradient differences using (17).
            Update the modifier matrix using (15b).
        else
            Define ΔU_k and ΔE_k as in (13b) and (13d).
            if $\|\Delta U_k\| \ge \rho_u$ **then**
                Evaluate the gradient difference using (14).
                Update the modifier matrix using (15b).
            else
                Do not update the modifier matrix.
            end if
        end if
    else
        Do not update the modifier matrix.
    end if
    Solve (10) to obtain the targets $(\bar{x}_k, \bar{u}_k, \bar{y}_k)$.
    Solve FHOCP (11), set the input u_k as in (12).
end if
Output: u_k, $\hat{x}^*_k$, $\hat{d}^*_k$ and Λ_k.
Inject the inputs u_k into the plant.
Update time index $k := k + 1$

Table 1. William-Otto Reactor: Plant Parameters

Parameter	Value	Unit
k_{10}	9.9594×10^6	$\mathrm{dm}^3/(\mathrm{mol}\cdot\mathrm{min})$
k_{20}	8.66124×10^9	$\mathrm{dm}^3/(\mathrm{mol}\cdot\mathrm{min})$
k_{30}	9.9594×10^6	$\mathrm{dm}^3/(\mathrm{mol}\cdot\mathrm{min})$
E_1	6666.7	K
E_2	8333.3	K
E_3	11111	K
c_{A0}	10	$\mathrm{mol}/\mathrm{dm}^3$
c_{B0}	10	$\mathrm{mol}/\mathrm{dm}^3$
V_r	2105	dm^3
Q_{A}	112.35	$\mathrm{dm}^3/\mathrm{min}$
p_{A}	7.623	\$/kmol
p_{B}	11.434	\$/kmol
p_{P}	114.338	\$/kmol
p_{E}	5.184	\$/kmol

Table 2. Williams-Otto Reactor: Model Parameters

Parameter	Value	Unit
k_{10}^*	1.3134×10^8	$\mathrm{dm}^6/(\mathrm{mol}^2\cdot\mathrm{min})$
k_{20}^*	2.586×10^{13}	$\mathrm{dm}^6/(\mathrm{mol}^2\cdot\mathrm{min})$
E_1^*	8077.6	K
E_2^*	12438.5	K

Model. The model used for control design comprises only two reactions:

$$\begin{aligned} \mathrm{A}+2\mathrm{B} &\xrightarrow{k_1^*} \mathrm{P}+\mathrm{E} \quad r_1^* = k_1^*(T_r)c_{\mathrm{A}}c_{\mathrm{B}}^2 \\ \mathrm{A}+\mathrm{B}+\mathrm{P} &\xrightarrow{k_2^*} \mathrm{G} \quad r_2^* = k_2^*(T_r)\,c_{\mathrm{A}}c_{\mathrm{B}}c_{\mathrm{P}} \end{aligned} \tag{21}$$

for which the kinetic parameters are reported in Table 2. With these parameter values, the plant settling time is of the order of $\tau_{st} = 25$ min.

Let us underline that the model state vector has five components, $x = [c_{\mathrm{A}}\ c_{\mathrm{B}}\ c_{\mathrm{P}}\ c_{\mathrm{E}}\ c_G]^T$, and differs from the plant state vector that has six components, $x_p = [c_{\mathrm{A}}\ c_{\mathrm{B}}\ c_{\mathrm{C}}\ c_{\mathrm{P}}\ c_{\mathrm{E}}\ c_{\mathrm{G}}]^T$. The following input constraints should be met at all times:

$$180\ \mathrm{dm}^3/\mathrm{min} \le Q_{\mathrm{B}} \le\ 360\ \mathrm{dm}^3/\mathrm{min} \tag{22}$$

$$75\ ^\circ\mathrm{C} \le T_{\mathrm{r}} \le 100\ ^\circ\mathrm{C} \tag{23}$$

Optimization results. The sampling time used for control is $\tau = 2$ min, which means that the time between two successive modifier updates is $M\tau = 30$ min. The performance of controllers eMPC0, eMPC1 and eMPC2 are depicted in Figure 2, together with the optimal equilibrium, which is unknown to the controllers. The simulation was performed on a PC with CPU Intel i5 7200u. The resulting computational cost required for a single iteration is near 0.15 s, and the time needed for the update of the modifier matrix is only 1% of that time. Furthermore, since updating is only required every M iterations, computational cost is not an issue for this problem.

eMPC0 is reported for comparison and, as expected, does not converge to the plant optimum. On the contrary, eMPC1 converges to plant optimality and does it very quickly because of the availability of perfect gradients. Hence, the real interesting case is eMPC2. Figure 2 has three distinct time zones that are separated by two vertical grey lines. In the first part, from the beginning to time $k_{in}\tau = 30$ min, with the modifiers equal to zero, eMPC2 reaches a steady state that is not the plant optimum. In the middle part, from $k_{in}\tau = 30$ to $(k_{in}+N_{in})\tau = 90$ min, modifier initialization is performed by perturbing the two inputs individually, one after the other. Then, the modifiers are updated every $M\tau = 30$ min, as described in Section 3.2,

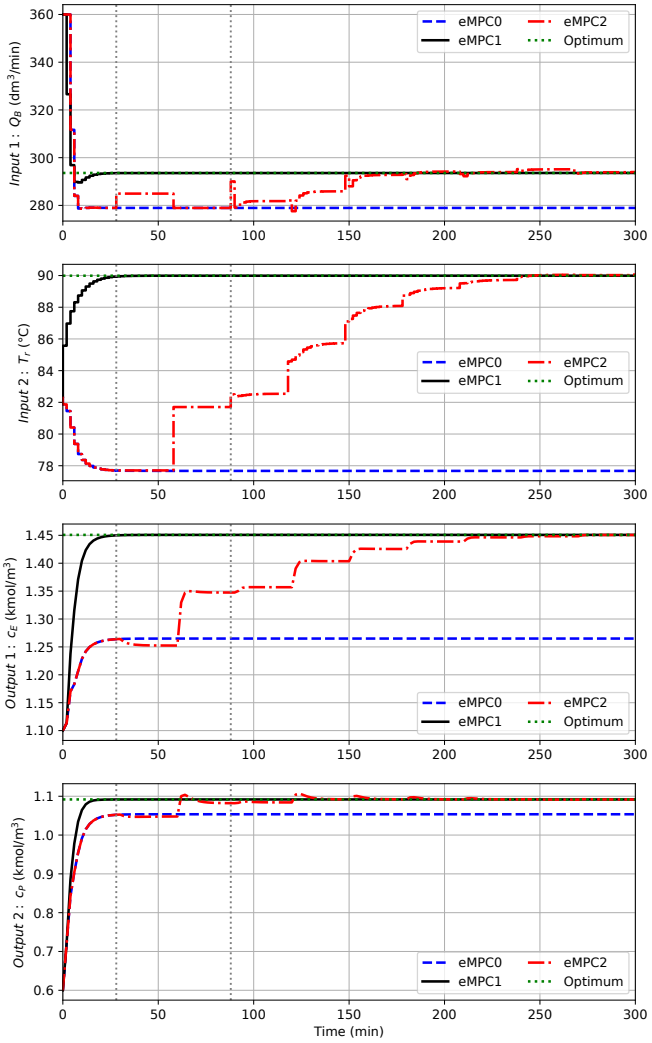

Fig. 2. Closed-loop results with eMPC0, eMPC1 and eMPC2: inputs (top) and outputs (bottom). The two grey vertical lines delimitate the time interval for modifier initialization. *Optimum* denotes optimal values.

which enforces convergence to plant optimality. Compared to eMPC1, eMPC2 estimates the static gradients but at the cost of slow convergence. A potential remedy to this slow convergence is to decrease M.

Figure 3 shows the effect of M on the convergence speed of eMPC2. $M \ge 10$ allows reaching steady-state conditions between modifier updates, which results in good estimate of the static gradients. The time needed for convergence is reduced using a smaller value of M, however at the price of no longer using steady-state plant measurements. As a result, the gradient differences estimated via Broyden's update are inaccurate. This effect is clearly visible in Figure 3 for $M = 3$. The best overall performance seems to be obtained using $M = 5$.

5. CONCLUSIONS

The two main ingredients of offset-free eMPC are the standard augmented system structure and the first-order modifiers for the outputs. This formulation proved to asymptotically converge to plant optimality despite model uncertainty. A fundamental requirement of the proposed scheme is the knowledge of plant

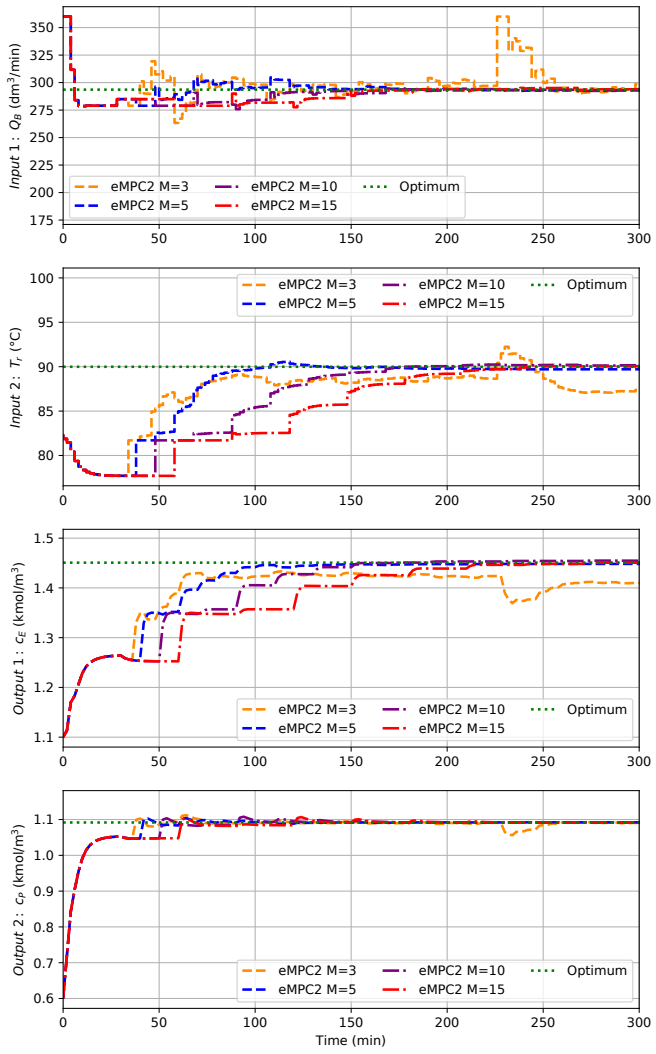

Fig. 3. eMPC2 performance with different values of M.

gradients at steady state. In this work, a modifier estimation technique based on steady-state perturbations and Broyden's gradient approximation has been proposed and used successfully.

A representative example with plant-model mismatch has been selected for validating the proposed OF-eMPC implementation. Estimating the modifiers directly appears to be very effective if steady-state measurements are used. However, the methodology was found less efficient with respect to convergence time. For this reason, the time between modifier updates has been reduced. Results show that, even if the measurements are not taken at steady state, the closed-loop system is still able to reach plant optimality, and the convergence time can be significantly reduced.

Further studies will focus on the influence of various tuning parameters to possibly enhance the convergence behavior. Furthermore, the influence of measurement noise on the gradient estimation will be analyzed. Moreover, convergence conditions of the algorithm and KKT matching upon convergence, are some of the properties that need to be investigated and proved.

REFERENCES

Costello, S., François, G., and Bonvin, D. (2016). A directional modifier-adaptation algorithm for real-time optimization. *Journal of Process Control*, 39, 64–76.

de Avila Ferreira, T., François, G., Marchetti, A.G., and Bonvin, D. (2017). Use of transient measurements for static real-time optimization. *IFAC-PapersOnLine*, 50(1), 5737–5742.

Dennis Jr, J.E. and Schnabel, R.B. (1996). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, volume 16. Society for Industrial Applied Mathematics.

Ellis, M. and Christofides, P.D. (2014). On finite-time and infinite-time cost improvement of economic model predictive control for nonlinear systems. *Automatica*, 50(10), 2561 – 2569.

Faulwasser, T. and Pannocchia, G. (2019). Towards a unifying framework blending RTO and economic MPC. *Industrial & Engineering Chemistry Research*, 58, 13583–13598.

François, G. and Bonvin, D. (2013). Use of transient measurements for the optimization of steady-state performance via modifier adaptation. *Industrial & Engineering Chemistry Research*, 53(13), 5148–5159.

Marchetti, A., François, G., Faulwasser, T., and Bonvin, D. (2016). Modifier adaptation for real-time optimization – Methods and applications. *Processes*, 4(4), 1–35.

Marchetti, A.G., Chachuat, B., and Bonvin, D. (2009). Modifier-adaptation methodology for real-time optimization. *Industrial & Engineering Chemistry Research*, 48(13), 6022–6033.

Pannocchia, G. (2015). Offset-free tracking MPC: A tutorial review and comparison of different formulations. In *European Control Conference*, 527–532.

Pannocchia, G. (2018). An economic MPC formulation with offset-free asymptotic performance. *IFAC-PapersOnLine*, 51(18), 393–398.

Pannocchia, G., Gabiccini, M., and Artoni, A. (2015). Offset-free MPC explained: Novelties, subtleties, and applications. *IFAC-PapersOnLine*, 48(23), 342–351.

Papasavvas, A., de Avila Ferreira, T., Marchetti, A.G., and Bonvin, D. (2019). Analysis of output modifier adaptation for real-time optimization. *Computers and Chemical Engineering*, 121, 285–293.

Rawlings, J.B., Angeli, D., and Bates, C.N. (2012). Fundamentals of economic model predictive control. In *51st IEEE Conference on Decision and Control*, 3851–3861.

Roberts, P.D. (2000). Broyden derivative approximation in ISOPE optimising and optimal control algorithms. In *11th IFAC Workshop on Control Applications of Optimisation*, 283–288. St Petersburg, Russia.

Rodger, E.A. and Chachuat, B. (2011). Design methodology of modifier adaptation for on-line optimization of uncertain processes. In *IFAC World Congress*, 4113–4118. Milano, Italy.

Vaccari, M. and Pannocchia, G. (2016). A modifier-adaptation strategy towards offset-free economic MPC. *Processes*, 5(1), 2.

Vaccari, M. and Pannocchia, G. (2018). Implementation of an economic MPC with robustly optimal steady-state behavior. *IFAC-PapersOnLine*, 51(20), 92–97.

Williams, T.J. and Otto, R.E. (1960). A generalized chemical processing model for the investigation of computer control. *Trans. of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 79(5), 458–473.