processes

MDPI

Article

Offset-Free Economic MPC Based on Modifier Adaptation: Investigation of Several Gradient-Estimation Techniques

Marco Vaccari [1], Dominique Bonvin [2], Federico Pelagagge [1] and Gabriele Pannocchia [1,*]

[1] Department of Civil and Industrial Engineering, University of Pisa, Largo Lazzarino 2, 56126 Pisa, Italy; marco.vaccari@ing.unipi.it (M.V.); federico.pelagagge@phd.unipi.it (F.P.)
[2] Laboratoire d'Automatique, EPFL, 1015 Lausanne, Switzerland; dominique.bonvin@epfl.ch
* Correspondence: gabriele.pannocchia@unipi.it

Abstract: Various offset-free economic model predictive control schemes that include a disturbance model and the modifier-adaptation principle have been proposed in recent years. These schemes are able to reach plant optimality asymptotically even in the presence of plant–model mismatch. All schemes are affected by a major issue that is common to all modifier-adaptation formulations, namely, plant optimality (note that convergence per se does not require perfect plant gradients) requires perfect knowledge of static plant gradients, which is a piece of information not known in most practical applications. To address this issue, we present two gradient-estimation techniques, one based on Broyden's update and the other one on linear regression. We apply these techniques for the estimation of either the plant gradients or the modifiers directly. The resulting economic MPC schemes are tested in a simulation and compared on two benchmark examples of different complexity with respect to both convergence speed and robustness to measurement noise.

Keywords: economic model predictive control; real-time optimization; modifier adaptation; gradient estimation; Broyden's update; linear regression

check for updates

Citation: Vaccari, M.; Bonvin, D.; Pelagagge F.; Pannocchia, G. Offset-Free Economic MPC Based on Modifier Adaptation: Investigation of Several Gradient-Estimation Techniques. *Processes* **2021**, *9*, 901. https://doi.org/10.3390/pr9050901

Academic Editor: Leyla Özkan and Alejandro Marquez Ruiz

Received: 26 April 2021
Accepted: 18 May 2021
Published: 20 May 2021

1. Introduction

Economic model predictive control (eMPC) [1] has been proposed to overcome the traditional separation between control and optimization layers in industrial process operations [2,3]. eMPC merges standard MPC with the economic optimization layer that is characteristic of real-time optimization (RTO) [2]. In particular, this combination is obtained by substituting the MPC tracking objective with economic performance [4]. There are many similarities between MPC and eMPC, with one of the main challenges being optimality under plant–model mismatch [3]. Note that the tracking-MPC (tMPC) and RTO communities have dealt with this challenge for many years, and two main solutions have been established. The first one concerns tMPC schemes and consists of augmenting the nominal model with a disturbance model so as to generate a steady-state offset correction to set-point changes [5,6]. A comprehensive review about offset-free tMPC can be found in [7]. On the other hand, RTO formulations have incorporated the so-called modifier-adaptation (MA) idea [8,9]. The essence of the MA strategy is to force the KKT optimality conditions of the modified model to match those of the plant upon convergence by using zeroth- and first-order correction terms [10].

To handle plant–model mismatch in eMPC schemes, a methodology that combines the augmented model used in tMPC [7] and the MA approach used in RTO [8] was first proposed in [11] and labeled "offset-free eMPC" (OF-eMPC). Alternative OF-eMPC formulations have been proposed in recent years [11–13]. These formulations can overcome plant–model mismatch and let the controlled plant reach plant optimality upon convergence. This is done with the help of zeroth- and first-order modifier terms that exploit an alternative version of MA, labeled "output MA" (MAy). Instead of modifying the cost function and the constraints as in traditional MA [8], MAy sets the modifier terms on the

outputs. A recent comprehensive analysis of MAy and its convergence properties can be found in [14].

Despite its ability to converge to plant optimality in the presence of plant–model mismatch, OF-eMPC suffers from a major caveat to proper MA schemes. Indeed, these schemes require accurate knowledge of static plant gradients with respect to inputs. This represents a major challenge, especially when dealing with noisy measurements. Even the most recent MA investigations, which strive for feasibility improvement [15], faster convergence [16], applicability to periodic systems [17], or large-scale systems [18], are still based on the assumption that the plant gradients are perfectly known. Nevertheless, several approaches for estimating plant gradients have been proposed, as detailed next.

The gradient estimation techniques found in literature can be divided into two classes [10]. The first class, and by far the largest, includes *steady-state* perturbation methods that rely on steady-state data (e.g., [19–21]). These methods are the ones that better replicate the concept of steadiness of RTO iterations. On the other hand, it might be advantageous with slow processes to use transient data and estimate steady-state gradients on the go [22–24]. In this case, we speak of *dynamic* perturbation methods, which exhibit a compromise between convergence speed and accuracy. Concretely, gradient approximations are computed using either a finite-difference approach (see [10] for an overview of available approaches) or some type of system identification [21,24,25]. Exploiting the so-called directional modifier-adaptation concept [20], recent works have shown how privileged directions can ease gradient estimation—thereby making the approach more reactive to plant–model mismatch [26,27]—and how these directions can be computed efficiently [28]. Jeong et al. [29] recently compared the performance of different gradient estimation methods with MA schemes. These methods include multivariate linear regression, partial least-squares regression, and principal component analysis. It is shown that the use of latent-variable space can result in fast convergence and stability near plant optimality.

It is worth noting that researchers have also tried to avoid working with plant gradient estimation. Navia et al. [30] formulated a nested optimization problem, in which the modifiers are computed via gradient-free optimization in the outer loop. Although the nested implementation was proved to be robust in the presence of noisy measurements and capable of reaching plant optimality, it was rather slow. Another modifier-estimation technique based on optimization has been proposed recently in [31]. In this case, the limiting factor becomes the choice of weighting factors in the optimization problem and the speed at which plant optimality can be reached.

This paper aims at investigating alternative solutions to the gradient estimation problem. In particular, we propose two main techniques to estimate static plant gradients, namely, a Broyden's update and linear regression. Note that these techniques have already been used to estimate plant gradients and modifiers in the RTO context, but not in the control context, that is, using fast sampling time and dynamic optimization. Another important advantage over our previous work [21,25] is not requiring repeated local perturbations of the process. Moreover, taking inspiration from the aforementioned "plant gradient-free" methodologies [30,31], we also use the same two techniques to estimate the modifiers directly.

This work represents an extension to [32], but it also goes further by comparing several gradient-estimation techniques. This is important to steer future research on how MA can best be combined with eMPC.

The paper is organized as follows. The problem definition and the constituents of OF-eMPC are detailed in Section 2. Several gradient-estimation techniques based on either plant gradient estimation or modifier estimation are discussed in Section 3, while Section 4 presents two OF-eMPC algorithms. The resulting OF-eMPC schemes are then compared via a numerical example and a standard benchmark case study in Section 5. Finally, conclusions are presented in Section 6.

2. Preliminaries

The OF-eMPC algorithm described in this paper deals with a general nonlinear plant and cost. Let us detail the main concepts and techniques.

2.1. Plant and Cost Specifications

The plant to be optimized is expressed by the following discrete-time nonlinear dynamic system:

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

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. The unknown functions $f_p : \mathbb{R}^{n_{x_p}} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_{x_p}}$ and $h_p : \mathbb{R}^{n_{x_p}} \to \mathbb{R}^{n_y}$ are assumed to be differentiable. The measured plant outputs y_p at time $k \in \mathbb{Z}$ are noted $y_{p,k}$. Furthermore, the inputs and outputs are bounded,

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

with $u_{\min}, u_{\max}, y_{\min}$, and $y_{\max}$ being the corresponding lower and higher bounds.

Plant (1) is said to be optimized when the known and differentiable function $\ell_e : \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is minimized. Hence, the triple $(\bar{x}_p^\star, \bar{u}^\star, \bar{y}_p^\star)$ that defines the economic optimum equilibrium of Plant (1) can be computed as:

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

subject to

$$x = f_p(x, u) \tag{4}$$

$$y = h_p(x) \tag{5}$$

$$u_{\min} \leq u \leq u_{\max} \tag{6}$$

$$y_{\min} \leq y \leq y_{\max}. \tag{7}$$

We introduce the following assumption about Problem (3)–(7):

Assumption 1 (Plant optimality)**.**
- *Problem (3)–(7) is feasible;*
- *The triple $(\bar{x}_p^\star, \bar{u}^\star, \bar{y}_p^\star)$ is the unique solution to Problem (3)–(7).*

Remark 1 (Unknown plant optimum)**.** *Given that the functions f_p and h_p in Equation (1) are typically unknown, the solution triple of Problem (3)–(7) is also unknown.*

2.2. Nominal and Augmented Models

We consider the following nominal process model:

$$\begin{aligned} x^+ &= f(x, u) \\ y &= h(x), \end{aligned} \tag{8}$$

in which x and $x^+ \in \mathbb{R}^{n_x}$ denote the current and successor states, respectively. It is assumed that the functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ are known and differentiable.

Remark 2 (Plant–model mismatch)**.** *The functions f and h in Equation (8) differ from the plant functions f_p and h_p in Equation (1) due to plant–model mismatch.*

Augmenting Model (8) linearly with the integral states $d \in \mathbb{R}^{n_d}$, called *disturbances*, has proven to be very effective for improving the performance of tMPC under plant–model

mismatch [5,6,33–35]. In the general formulation, the effects of the disturbances on the states and outputs are modeled using the matrices $B_d \in \mathbb{R}^{n_x \times n_d}$ and $C_d \in \mathbb{R}^{n_y \times n_d}$:

$$\begin{aligned} x^+ &= F(x,u,d) = f(x,u) + B_d d \\ d^+ &= d \\ y &= H(x,d) = h(x) + C_d d. \end{aligned} \tag{9}$$

The resulting augmented 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 and consistent with f and h, that is, $F(x,u,0) \to f(x,u)$, $H(x,0) \to h(x)$. To guarantee that the augmented model can represent the plant without steady-state offset, we introduce the following assumption (see [35] (Remark 8)):

Assumption 2 (Observability)**.** *The augmented system* (9) *is observable.*

2.3. State and Disturbance Estimation

At each time step, the estimation procedure includes two steps, namely, prediction and filtering. First, using the augmented model (9), one computes the *predicted values* $\hat{x}_k, \hat{d}_k$, and $\hat{y}_k$ as

$$\begin{aligned} \hat{x}_k &= F(\hat{x}^*_{k-1}, u_{k-1}, \hat{d}^*_{k-1}) \\ \hat{d}_k &= \hat{d}^*_{k-1} \\ \hat{y}_k &= H(\hat{x}_k, \hat{d}_k), \end{aligned} \tag{10}$$

where $\hat{x}^*_{k-1}$ and $\hat{d}^*_{k-1}$ are the *estimates* of x_{k-1} and d_{k-1} at time $k-1$. Then, using output measurements, these predicted values are filtered appropriately to give the estimates $\hat{x}^*_k, \hat{d}^*_k$, and $\hat{y}^*_k$.

There are different types of estimators. For example, the so-called moving-horizon estimator (MHE) processes past output measurements in an optimization framework [36]. In contrast, a Luenberger observer or Kalman filter uses only the current measurements $y_{p,k}$ to estimate the augmented states from their predicted values. This is the type of estimator used in this study and is detailed next.

The prediction errors at time k are expressed as

$$\epsilon_k = y_{p,k} - \hat{y}_k \tag{11}$$

and used to estimate the augmented states as follows:

$$\begin{aligned} \hat{x}^*_k &= \hat{x}_k + K_x \epsilon_k \\ \hat{d}^*_k &= \hat{d}_k + K_d \epsilon_k. \end{aligned} \tag{12}$$

Note that the matrices $K_x \in \mathbb{R}^{n_x \times n_y}$ and $K_d \in \mathbb{R}^{n_d \times n_y}$ must be chosen to form an asymptotically stable observer, which implies $n_y = n_d$ and K_d to be invertible [5,35].

2.4. Output Modifiers

The output-correction scheme adopted in this study follows the MAy formulation that was originally developed in the RTO literature [14] and firstly applied in the eMPC framework by [11]. The correction is based on zeroth- and first-order terms that represent the steady-state deviations between plant and model outputs. In our implementation, we use the disturbance estimates $\hat{d}^*$ as zeroth-order corrections. The first-order modifier matrix at time k is noted $\Lambda_k \in \mathbb{R}^{n_y \times n_u}$, with $\Lambda_0 = 0$. Its evolution over time is described by the following filtering equation:

$$\Lambda_k = (1-\sigma)\Lambda_{k-1} + \sigma\Big(\mathrm{D}_u g_p(\bar{u}_{k-1}) - \mathrm{D}_u g(\bar{u}_{k-1})\Big), \tag{13}$$

where σ is a scalar filter constant $\in (0,1]$, the operators $\mathrm{D}_u(\cdot)$ are the derivatives of the considered output functions (either g_p or g) with respect to the inputs u, and $\bar{u}_{k-1}$ are

the input steady-state targets computed at the previous iteration by the target calculation described in the next subsection.

The plant steady-state input-to-output maps $g_p : \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ are unknown, and $\mathrm{D}_u g_p(\bar{u}_{k-1})$ must be estimated experimentally from input and output measurements. The model steady-state input-to-output maps $g : \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ are calculated from Model (8) upon imposing the target inputs $\bar{u}_{k-1}$ as follows:

$$\begin{cases} \bar{x}_{k-1} = f(\bar{x}_{k-1}, \bar{u}_{k-1}) \\ \bar{y}_{k-1} = h(\bar{x}_{k-1}) \end{cases} \implies \bar{y}_{k-1} = g(\bar{u}_{k-1}). \tag{14}$$

Remark 3 (Model gradients)**.** *Since a linear disturbance model is used, the derivatives of Models* (8) *and* (9) *with respect to u are identical.*

2.5. Target Calculation with Modifiers

The equilibrium triple $(\bar{x}_k, \bar{u}_k, \bar{y}_k)$ at time k is calculated via steady-state optimization. The modifiers described in Section 2.4 correct the outputs so as to eliminate any first-order plant–model mismatch. The following steady-state target problem is solved at each iteration:

$$(\bar{x}_k, \bar{u}_k, \bar{y}_k) = \underset{(x,u,y)}{\arg\min}\ \ell_e(y, u) \tag{15}$$

subject to

$$x = F(x, u, \hat{d}_k^*) \tag{16}$$

$$y = H(x, \hat{d}_k^*) + \Lambda_k(u - \bar{u}_{k-1}) \tag{17}$$

$$u_{\min} \leq u \leq u_{\max} \tag{18}$$

$$y_{\min} \leq y \leq y_{\max}. \tag{19}$$

The modifier terms $\Lambda_k(u - \bar{u}_{k-1})$ in Equation (17) ensure KKT matching between Problems (3)–(7) and (15)–(19) [11,14].

2.6. Economic MPC with Modifiers

The finite-horizon optimal control problem (FHOCP) typical of eMPC is modified similarly to the method in Equation (17). Defining the generic state and input sequences $\boldsymbol{x} := \{\chi_0, \chi_1, \ldots, \chi_N\}$ and $\boldsymbol{u} := \{\nu_0, \nu_1, \ldots, \nu_{N-1}\}$, the optimal problem solved over N steps at each decision time k reads

$$(\boldsymbol{x}_k^\star, \boldsymbol{u}_k^\star) = \underset{\boldsymbol{x},\boldsymbol{u}}{\arg\min} \sum_{i=0}^{N-1} \ell_e(\gamma_i, \nu_i) \tag{20}$$

subject to

$$\chi_0 = \hat{x}_k^* \tag{21}$$

$$\chi_{i+1} = F(\chi_i, \nu_i, \hat{d}_k^*) \tag{22}$$

$$\gamma_i = H(\chi_i, \hat{d}_k^*) + \Lambda_k(\nu_i - \bar{u}_{k-1}) \tag{23}$$

$$u_{\min} \leq \nu_i \leq u_{\max} \tag{24}$$

$$y_{\min} \leq \gamma_i \leq y_{\max} \tag{25}$$

$$\chi_N = \bar{x}_k. \tag{26}$$

Assumption 3 (FHOCP)**.**

- *Problem* (20)–(26) *is feasible*
- *The ordered pair* $(\boldsymbol{x}_k^\star, \boldsymbol{u}_k^\star)$ *is the unique solution to Problem* (20)–(26)

The inputs implemented in the plant are the first ones of the optimal sequence $\boldsymbol{u}_k^\star$, that is,

$$u_k = \boldsymbol{u}_k^\star[1]. \tag{27}$$

We remark that, since the modifier terms in Equations (17) and (23) serve the same goal, for consistency purposes, the steady-state input targets used in Problem (20)–(26) are $\bar{u}_{k-1}$ and not $\bar{u}_k$. Moreover, for ensuring closed-loop stability of eMPC at nominal conditions, the following assumption is introduced:

Assumption 4. *Steady-state operation is assumed to be more profitable than oscillating behavior, that is, the system is dissipative with respect to the stage cost $\ell_e(\cdot)$ [37].*

Furthermore, to be able to deal with plant–model mismatch, we also need the following assumption:

Assumption 5. *The closed-loop control sequence generated by the FHOCP (20)–(26) converges.*

Note that Assumption 5 is consistent with the convergence requirements of tracking offset-free MPC [7], modifier-adaptation schemes [9,14], and offset-free economic MPC [12].

3. Gradient or Modifier Estimation

The issue limiting the practical applicability of MA methods (and thus, also of MAy), is the need to gather appropriate information about the plant–model mismatch. In particular, it is necessary to accurately estimate the steady-state plant gradients $\mathrm{D}_u g_p(\cdot)$ for the OF-eMPC algorithm in Section 2 to work properly. As a matter of fact, convergence to plant optimality is only ensured with perfect gradient estimation [38].

3.1. Basic Idea

This section compares two methodologies based on steady-state perturbation methods. We also compare the performance when one uses only measurements taken at the current iteration versus data collected at several (previous) iterations.

In Section 2.4, the first-order modifiers Λ_k are updated at every sampling time. This works fine if the plant measurements correspond to steady-state conditions; however, when dealing with transient measurements, the system must reach some sort of "steadiness" before updating Λ_k. For this reason, with all the methods investigated next, the modifiers are updated every M sampling times, where $M \sim \frac{\tau_{st}}{\tau}$ is a tuning parameter, with τ_{st} being the time to reach steady state and τ being the sampling time. This way, we let the system reach quasi-stationary conditions between two successive modifier updates, thereby satisfying the necessity of operating with (near) steady-state measurements. This implies a time-scale separation between the control and gradient-estimation tasks. Since Λ_k is updated recursively, it helps to start the recursion with a fair estimate of Λ. In this initialization phase, we apply n_u appropriate input perturbations and collect quasi-stationary output measurements, with which we can construct an initial nonzero Λ_0. The next two subsections will address plant gradient estimation and modifier adaptation, respectively.

3.2. Plant Gradient Estimation

We propose two techniques for the estimation of plant output gradients.

3.2.1. Broyden's Update for Plant Gradient Estimation

Broyden's update offers a recursive way of updating the estimated gradients using current and past measurements [10,39,40]. The advantage of Broyden's update is that it does not require additional perturbations; however, being a numerical method, sufficient variation between two consecutive inputs is needed for the scheme not to fail. The technique is a standard *secant method* in nonlinear programming for updating estimates of first-order derivatives, such as Jacobian matrices [41].

At the kth iteration, $\mathrm{D}_u g_{p,k}$, which are estimates of $\mathrm{D}_u g_p(\bar{u}_{k-1})$, are updated by defining

$$\Delta Y_k := y_{p,k} - y_{p,k-M} \tag{28}$$

$$\Delta U_k := u_{k-1} - u_{k-M-1} \tag{29}$$

and using Broyden's formula as follows:

$$\mathrm{D}_u g_{p,k} = \mathrm{D}_u g_{p,k-M} + \frac{\Delta Y_k - \mathrm{D}_u g_{p,k-M} \Delta U_k}{\Delta U_k^T \Delta U_k} \Delta U_k^T. \tag{30}$$

The gradients of the model output functions $g(\cdot)$ can be computed from their definition (14) using the implicit function theorem:

$$\mathrm{D}_u g(\cdot) = \mathrm{D}_x h(x) \Big[(I - \mathrm{D}_x f(x,u))^{-1} \mathrm{D}_u f(x,u) \Big]. \tag{31}$$

Then, the gradient differences are calculated as follows:

$$\Delta g_k = \mathrm{D}_u g_{p,k} - \mathrm{D}_u g_k. \tag{32}$$

When applying Equation (30), care must be taken to avoid ill-conditioning when $\Delta U_k \to 0$. Hence, the step given by Equation (30) is performed only if $\|\Delta U_k\| \geq \rho_u$, where ρ_u is a chosen threshold.

3.2.2. Linear Regression for Plant Gradient Estimation

This method relies on output measurements at $n_s + 1$ steady-state operating points corresponding to the inputs $u_k, u_{k-M}, \ldots, u_{k-n_s M}$, with $n_s \geq n_u$. Using such a data set, the plant output gradients are computed as linear interpolation if $n_u = n_s$ or linear regression if $n_s > n_u$. These gradients are often called *simplex* gradients [42]. A similar approach is used in [43]. In the present work, we use n_s past points, with $n_s > n_u$, to estimate the n_u-dimensional gradients since the resulting *redundancy* helps deal with measurement noise [20].

Hence, at the kth iteration, we set $n_{s_k} = \min\left\{n_s, \frac{k}{M} - 1\right\}$ to construct the following matrices:

$$\mathcal{U}_k = \Big[u_k - u_{k-M} \; \ldots \; u_k - u_{k-n_{s_k} M} \Big] \in \mathbb{R}^{n_u \times n_{s_k}} \tag{33}$$

$$\mathcal{Y}_{p,k} = \Big[y_{p,k} - y_{p,k-M} \; \cdots \; y_{p,k} - y_{p,k-n_{s_k} M} \Big] \in \mathbb{R}^{n_y \times n_{s_k}}. \tag{34}$$

The length of the sets $\mathcal{U}_k$ and $\mathcal{Y}_{p,k}$ increases at each iteration from n_u, when only $n_u + 1$ operation points are available, to n_s. Defining

$$\mathcal{Y}_{p,k} = \mathrm{D}_u g_{p,k} \mathcal{U}_k, \tag{35}$$

the simplex gradients are computed as the least-squares solution to Equation (35):

$$\mathrm{D}_u g_{p,k} = \mathcal{Y}_{p,k} \mathcal{U}_k^{\dagger} = \mathcal{Y}_{p,k} \mathcal{U}_k^T (\mathcal{U}_k \mathcal{U}_k^T)^{-1}. \tag{36}$$

The matrix $\mathcal{U}_k^{\dagger}$ in Equation (36) can be poorly conditioned if the successive points do not extend in all directions of the input space [42]. This ill-conditioning can lead to erroneous gradient estimates; hence, as with Broyden's update, Step (36) is performed only if $\|\mathcal{U}_k\|_1 \geq \rho_u$. Then, the gradient differences are calculated as in (32).

3.3. Modifier Estimation

In this section, we propose to use the two techniques developed in Section 3.2 to estimate the *differences* between the plant and model gradients directly, that is, Δg_k. In fact, 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 made for linear interpolation in [10, Figure 3], but this is beyond the scope of the present paper.

3.3.1. Broyden's Update for Modifier Estimation

Analogously to Section 3.2.1, we define the following quantities:

$$\delta y_k := y_{p,k} - h(\hat{x}_k) \tag{37}$$

$$\Delta E_k := \delta y_k - \delta y_{k-M}, \tag{38}$$

which leads to Broyden's update:

$$\Delta g_k = \Delta g_{k-M} + \frac{\Delta E_k - \Delta g_{k-M} \Delta U_k}{\Delta U_k^T \Delta U_k} \Delta U_k^T. \tag{39}$$

It should be noted that when applying Equation (39), the same issue mentioned for Equation (30) of avoiding ill-conditioning must be considered.

3.3.2. Linear Regression for Modifier Estimation

Analogously to Equation (34), at the kth iteration, we construct the following matrix:

$$\Delta \mathcal{E}_k = \left[\delta y_k - \delta y_{k-M} \; \ldots \; \delta y_k - \delta y_{k-n_{s_k} M} \right] \in \mathbb{R}^{n_y \times n_{s_k}}, \tag{40}$$

with δy_k defined in Equation (37). The matrix of differences between plant and model gradients is computed as the least-squares solution to the following regression problem:

$$\Delta \mathcal{E}_k = \mathrm{D}_u g_{p,k} \mathcal{U}_k - \mathrm{D}_u g_k \mathcal{U}_k = \Delta g_k \mathcal{U}_k, \tag{41}$$

for which the solution is

$$\Delta g_k = \Delta \mathcal{E}_k \mathcal{U}_k^{\dagger} = \Delta \mathcal{E}_k \mathcal{U}_k^T (\mathcal{U}_k \mathcal{U}_k^T)^{-1}. \tag{42}$$

As above, we need to consider the ill-conditioning problem occurring when $\mathcal{U}_k \to 0$.

3.4. Role of Disturbance Model

The role played by the disturbance model is not the same for plant gradient estimation and modifier estimation. This issue is discussed next.

3.4.1. Plant Gradient Estimation

In the case of plant gradient estimation, the computation of $\mathrm{D}_u g_{p,k}$ in Equation (30) or Equation (36) depends on the disturbance model that has been selected:

- Using a *linear* disturbance model, as in (9), implies an equivalence between the nominal and augmented models for computing $\mathrm{D}_u g(\cdot)$, as in (31), since the disturbances d would not appear in any of the derivatives involved, that is, $\mathrm{D}_x F(\cdot), \mathrm{D}_x H(\cdot)$, and $\mathrm{D}_u F(\cdot)$. Hence, the disturbance variables are not involved in the gradient calculation.
- In contrast, using a *nonlinear* disturbance model, it would not be possible to discard the disturbance variables when calculating gradients. Moreover, a nonlinear disturbance model may be more challenging to implement; however, if implemented correctly, the first-order modifier terms might become superfluous (see [11] for an example).

Hence, in the case of plant gradient estimation, the most straightforward option is to work with a linear disturbance model and use first-order modifiers.

3.4.2. Modifier Estimation

The driving terms ΔE_k in Broyden's update (39) and $\Delta \mathcal{E}_k$ in the linear regression update (42) use the output errors δy_k (the differences between the plant and *nominal*

model outputs) in place of the prediction errors ϵ_k (the differences between the plant and *augmented* model outputs). As a matter of fact, using

$$\Delta E_k = \epsilon_k - \epsilon_{k-M} = \left(y_{p,k} - H(\hat{x}_k, \hat{d}_k)\right) - \left(y_{p,k-M} - H(\hat{x}_{k-M}, \hat{d}_{k-M})\right) \tag{43}$$

or

$$\Delta \mathcal{E}_k = \left[\epsilon_k - \epsilon_{k-M} \ \ldots \ \epsilon_k - \epsilon_{k-n_{s_k}M}\right] \in \mathbb{R}^{n_y \times n_{s_k}} \tag{44}$$

to compute Δg_k in Equation (39) or Equation (42) would not work. The implicit secant equation behind the Broyden update, namely, $\epsilon_k - \epsilon_{k-M} = \Delta g_k \Delta U_k$, does not hold true because $\hat{d}_k \neq \hat{d}_{k-M}$. The same problem appears when dealing with the linear regression update and Equation (41). Hence, the nominal model (8) is used for estimation, while the augmented model (9) is used for optimization to ensure convergence.

4. Offset-Free Economic MPC Algorithms

This sections formulates algorithmically the two OF-eMPC schemes based on Broyden's update and linear regression. The way the algorithms are initialized is presented first.

4.1. Algorithm Initialization

To improve convergence speed, the OF-eMPC schemes require a good initial estimate for Δg, which translates into an appropriate Λ_0. This can be achieved by perturbing each input individually around the current operating point and calculating the corresponding gradient elements, which requires n_u input perturbations. Following each perturbation, one must wait M sampling times for the system to approach a steady state.

Next, we detail this initialization phase. Let k_0 denote the iteration at which the plant first reaches steady state, with the inputs u_{k_0}. The following control law is then used during the initialization phase, for $j = 1, \ldots, n_u$:

$$u_k = u_{k_0} + s_j e_j \qquad \text{for} \qquad (j-1)M \leq k - k_0 < jM, \tag{45}$$

where s_j is the amplitude of a step of duration M in the direction e_j, with e_j being a unit vector in input space. One sees that the jth component of u is perturbed individually by the amount of $s_j e_j$ during M iterations. The initialization phase ends at iteration $k_f := k_0 + n_u M$.

The jth column of the plant gradient to be used in Equation (30) is estimated as

$$\mathrm{D}_u g_{p,k_f,j} = \frac{y_{p,k_0+jM} - y_{p,k_0}}{s_j}, \tag{46}$$

while the jth column of the gradient differences to be used in Equation (39) is

$$\Delta g_{k_f,j} = \frac{\left(y_{p,k_0+jM} - y_{p,k_0}\right) - \left(h(\hat{x}_{k_0+jM}) - h(\hat{x}_{k_0})\right)}{s_j}. \tag{47}$$

Note that, unlike Broyden's update, the linear regression method in Equation (36) or in Equation (42) does not require Jacobian initialization, albeit the initialization phase is still used.

These estimated plant gradients or gradient differences are used to compute Λ_k as follows:

$$\Lambda_k = \begin{cases} 0 & \text{for } k_0 < k < k_f \quad \text{(initialization phase)} & (48) \\ \Lambda_{k-1} & \text{if mod } (k - k_f, M) \neq 0 \qquad k > k_f & (49) \\ (1-\sigma)\Lambda_{k-1} + \sigma \Delta g_k & \text{if mod } (k - k_f, M) = 0 \qquad k \geq k_f. & (50) \end{cases}$$

4.2. OF-eMPC Algorithm Using Broyden's Update

The block diagram and algorithm describing the OF-eMPC scheme with Broyden's update are given in Figure 1 and Algorithm 1. For the sake of brevity, the initialization phase is not shown, and only the behavior for $k > k_f$ is displayed.

As can be seen in Figure 1, gradient estimates are computed after $k_0 + n_u M$ iterations. 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 gradient differences based on little informative data, that is, to avoid ill-conditioning of the updating Equation (30) or Equation (39). Hence, the parameter ρ_u can be seen as a tuning parameter for the performance of the gradient estimation method.

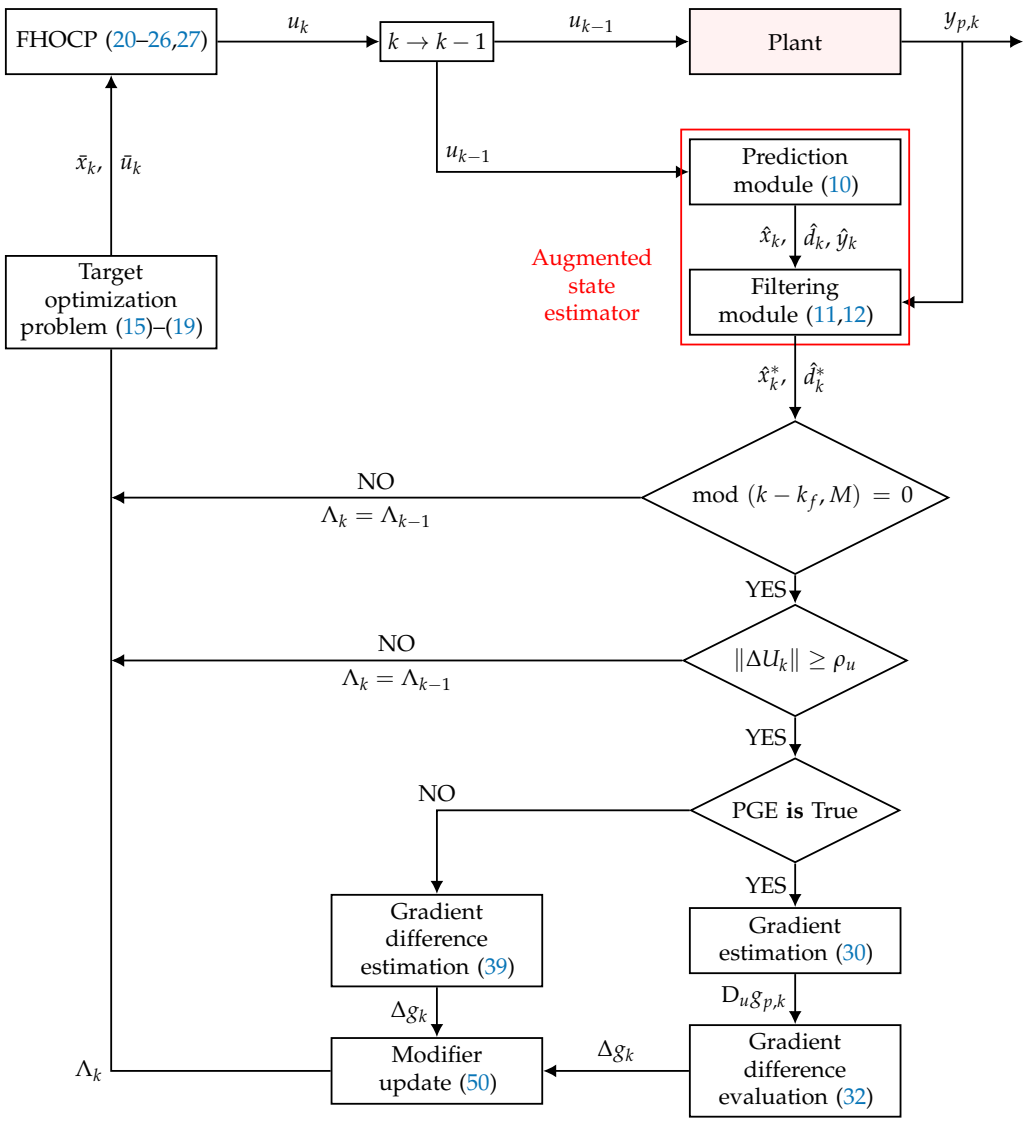

Figure 1. Block diagram for the eMPC algorithm described in Sections 2, 3.2.1, and 3.3.1 using Broyden's update formula, for $k > k_f$. The acronym PGE stands for Plant Gradient Estimation.

Algorithm 1 Offset-free eMPC algorithm using Broyden's update.

1: Set $k = k_f$ and initialize u_k using current plant values, $\hat{x}_k^*$, $\hat{d}_k^*$ from (12) and Λ_k from (50), with $\Lambda_{k-1} = 0$.
2: Inject inputs u_k into plant and wait for next sampling instant.
3: Update time index $k := k + 1$.
4: Read $y_{p,k}$ from plant.
5: Predict successor quantities using (10).
6: Evaluate prediction errors using (11).
7: Estimate states and disturbances using (12).
8: **if** $\mod(k - k_f, M) = 0$ **then**
9: **if** Plant Gradient Estimation **is** True **then**
10: Define ΔU_k and ΔY_k as in (29) and (28).
11: **else**
12: Define ΔU_k and ΔE_k as in (29) and (38).
13: **end if**
14: **if** $\|\Delta U_k\| \geq \rho_u$ **then**
15: **if** Plant Gradient Estimation **is** True **then**
16: Evaluate plant gradients using (30).
17: Evaluate gradient differences using (32).
18: **else**
19: Evaluate gradient differences using (39).
20: **end if**
21: Update modifier matrix using (50).
22: **else**
23: Do not update modifier matrix.
24: **end if**
25: **else**
26: Do not update modifier matrix.
27: **end if**
28: Solve (15)–(19) to obtain targets $(\bar{x}_k, \bar{u}_k, \bar{y}_k)$.
29: Solve FHOCP (20)–(26), set inputs u_k as per (27).
30: **Output:** u_k, $\hat{x}_k^*$, $\hat{d}_k^*$, and Λ_k.
31: **Goto** 2.

4.3. OF-eMPC Algorithm Using Linear Regression

The block diagram and algorithm describing the OF-eMPC scheme that uses linear regression are given in Figure 2 and Algorithm 2. Here, the initialization phase is not shown, and only the behavior for $k > k_f$ is displayed.

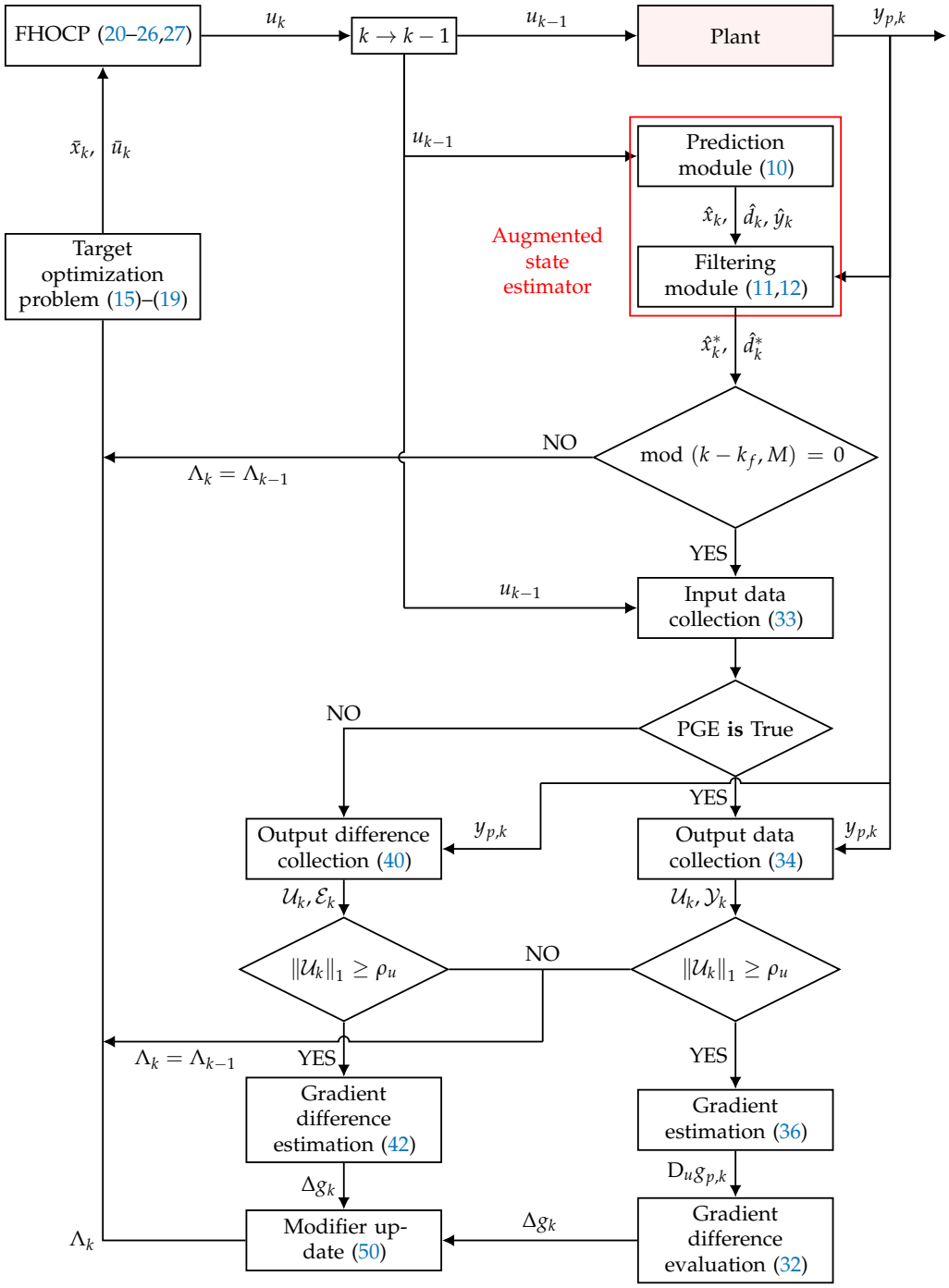

Figure 2. Block diagram for the eMPC algorithm described in Sections 2, 3.2.2, and 3.3.2 using the linear regression solution, for $k > k_f$. The acronym PGE stands for Plant Gradient Estimation.

Algorithm 2 Offset-free eMPC algorithm using linear regression

1: Set $k = k_f$ and initialize u_k using current plant values, $\hat{x}_k^*$, $\hat{d}_k^*$ from (12) and Λ_k from (50), with $\Lambda_{k-1} = 0$.
2: Inject inputs u_k into plant and wait for next sampling instant.
3: Update time index $k := k + 1$.
4: Read $y_{p,k}$ from plant.
5: Predict successor quantities using (10).
6: Evaluate prediction errors using (11).
7: Estimate states and disturbances using (12).
8: **if** $\mod (k - k_f, M) = 0$ **then**
9: Update input matrix $\mathcal{U}_k$ as in (33).
10: **if** Plant gradient estimation **is** True **then**
11: Update output matrix $\mathcal{Y}_k$ as in (34).
12: **else**
13: Update output difference matrix $\mathcal{E}_k$ as in (40).
14: **end if**
15: **if** $\|\mathcal{U}_k\|_1 \geq \rho_u$ **then**
16: **if** Plant gradient estimation **is** True **then**
17: Evaluate plant gradients using (36)
18: Evaluate gradient differences using (32).
19: **else**
20: Evaluate gradient differences using (42).
21: **end if**
22: Update modifier matrix using (50).
23: **else**
24: Do not update modifier matrix.
25: **end if**
26: **else**
27: Do not update modifier matrix.
28: **end if**
29: Solve (15)–(19) to obtain targets $(\bar{x}_k, \bar{u}_k, \bar{y}_k)$.
30: Solve FHOCP (20)–(26), set inputs u_k as in (27).
31: **Output:** u_k, $\hat{x}_k^*$, $\hat{d}_k^*$, and Λ_k.
32: **Goto** 2.

4.4. Ill-Conditioning

Ill-conditioning of Δg_k in Equations (30), (36), (39), or (36) results from too small a value of ΔU_k or $\mathcal{U}_k$. When ΔU_k or $\mathcal{U}_k$ tends to zero, the aforementioned Δg_k can no longer be calculated. This happens when two measurements corresponding to (almost) the same steady state are used in ΔU_k or $\mathcal{U}_k$, that is, with $u_{k-1} \approxeq u_{k-M-1}$. Hence, before calculating Δg_k, we compare the value of ΔU_k or $\mathcal{U}_k$ with the threshold value ρ_u, which depends on the input–output sensitivity and the size of measurement noise. This is an engineering choice that the designer has to make based on the process at hand.

We underline that, if modifier update does not take place due to ill-conditioning, and thus the previous values of the gradients or modifiers are used, Problems (15) and (20) are still solved as part of OF-eMPC. On the other hand, if the inputs u get "stuck" far away from plant optimality, the system requires additional excitation for accurate estimation of plant gradients or modifiers.

5. Simulation Results

Two examples taken from the literature are used to compare the performance of five eMPC schemes, all using the same nominal model, cost function, and sampling time. There is significant plant–model mismatch, and the augmented model (9) reads

$$\begin{aligned} x^+ &= f(x,u) \\ d^+ &= d \\ y &= h(x) + d. \end{aligned} \tag{51}$$

The estimator of Section 2.3 is the deadbeat Kalman filter with the parameters $K_x = \mathbf{0}$ and $K_d = \mathrm{I}$. In addition, all eMPC controllers use the target calculation (15) and the FHOCP (20). The five controllers are the following:

1. eMPC0 assumes the plant gradients to be perfectly known, i.e., $\mathrm{D}_u g_{p,k} = \mathrm{D}_u g_p(\bar{u}_{k-1})$. This controller is not practically implementable but is used as reference.
2. eMPC1 uses the Broyden's gradient update described in Section 3.2.1.
3. eMPC2 estimates the plant gradient using linear regression as per Section 3.2.2. The number of data points in the regression is $n_s = 4$.
4. eMPC3 uses the Broyden's modifier update described in Section 3.3.1.
5. eMPC4 estimates the modifiers using linear regression as per Section 3.3.2. The number of data points in the regression is also $n_s = 4$.

Note that, in the presence of plant–model mismatch, the use of the disturbance model (51) by itself is not sufficient to eliminate offset, as reaching plant optimality requires perfect knowledge of plant gradients [25,32]. When not otherwise specified, $M = 15$ is chosen for all controllers. Moreover, the filter constant for updating the modifiers in Equation (13) is $\sigma = 0.5$, and the tolerance threshold for avoiding numerical ill-conditioning is $\rho_u = 5 \cdot 10^{-4}$.

5.1. CSTR with Competitive Reactions

The first example is a continuous stirred-tank reactor (CSTR), in which two consecutive reactions take place:

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C. \tag{52}$$

The objective is to maximize the profit expressed as the difference between the revenues generated by the desired product B and the cost of the raw material A. The following system of ordinary differential equations describes the reactor dynamics:

$$\begin{aligned} \dot{x}_1 &= \tfrac{u}{V}(c_{A0} - x_1) - k_1 x_1 \\ \dot{x}_2 &= \tfrac{u}{V}(c_{B0} - x_2) + k_1 x_1 - k_2 x_2, \end{aligned} \tag{53}$$

where x_1 and x_2 represent the molar concentrations of A and B in the reactor, the corresponding feed concentrations are c_{A0} and c_{B0}, and the input u is the feed flow rate regulated through a valve. For the sake of simplicity, the reactor is assumed to have a constant volume V and to be isothermal. Both states are assumed to be measured. The reactor parameters are taken from [25]. Optimal steady-state performance is computed by solving the following optimization problem:

$$\max_{u,x} p(u,x) \tag{54}$$

subject to

$$\frac{u}{V}(c_{A0} - x_1) - k_1 x_1 = 0 \tag{55}$$

$$\frac{u}{V}(c_{B0} - x_2) + k_1 x_1 - k_2 x_2 = 0 \tag{56}$$

$$0 \leq x_1 \leq c_{A0} \tag{57}$$

$$0 \leq x_2 \leq c_{A0}, \tag{58}$$

with the running profit

$$p(u, x) = \beta_B u\, x_2 - \beta_A u\, c_{A0}\,. \tag{59}$$

Optimal operation corresponds to $u_s^\star = 1.043\,\mathrm{m}^3/\mathrm{min}$, $x_{1,s}^\star = 0.511\,\mathrm{kmol/m}^3$, and $x_{2,s}^\star = 0.467\,\mathrm{kmol/m}^3$ with the maximal profit $p(u^\star, x^\star) = 0.906$ €/min.

5.1.1. Model

We assume plant–model mismatch and design the controller with erroneous values of the two kinetic parameters: the model values known by the controller are $\tilde{k}_1 = 0.5\,\mathrm{min}^{-1}$ and $\tilde{k}_2 = 0.4\,\mathrm{min}^{-1}$, meaning that, compared to the true values $k_1 = 1.0\,\mathrm{min}^{-1}$ and $k_2 = 0.05\,\mathrm{min}^{-1}$, the first reaction rate is underestimated, while the second one is overestimated. The economic cost function to be minimized in Problem (20) is

$$\ell_e(y(t_i), u(t_i)) = -\int_{t_i}^{t_i+\tau} p\big(u(t), y_2(t)\big)dt = \int_{t_i}^{t_i+\tau} \big[\beta_A u(t) c_{A0} - \beta_B u(t) y_2(t)\big]dt, \tag{60}$$

with the sampling time $\tau = 0.25\,\mathrm{min}$. Furthermore, the dynamics (53) is discretized using an implicit Euler method and the initial conditions are $x_{1,0} = x_{2,0} = 0.1\,\mathrm{kmol/m}^3$. Note that, with $\tau_{st} \simeq 4\,\mathrm{min}$, the choice $M = 15$ is quite appropriate.

5.1.2. Reactor Performance

The input and the cost function of the reactor with the five aforementioned controllers are depicted in Figure 3. We can immediately see that all schemes reach plant optimality even under the presence of plant–model mismatch. Before reaching the initialization phase at $t = 3.75\,\mathrm{min}$, the behavior is the same for the practical controllers eMPC1–eMPC4 as they all use the disturbance model (51) and no first-order correction. eMPC0 is, of course, significantly better as it uses perfect knowledge of plant gradients. Λ_k is updated for the first time at time step $k_f = k_0 + n_u M = 30$, which corresponds to $t = 7.5\,\mathrm{min}$. After initialization, we see a slight difference between the schemes that use Broyden's update (eMPC1 and eMPC3) and those that rely on linear regression (eMPC2 and eMPC4). All of them start to move towards plant optimality, with the methodologies using Broyden's update reaching steady state faster. We observe a steplike profile, since the corrections are updated every M time samples. The average convergence time for the proposed methods is about 40 min.

To speed up convergence, one can reduce the value of M, thereby looking for a compromise between convergence speed and steadiness of the collected samples. Figure 4 shows the reactor performance with the five controllers when M has been reduced from 15 to 5. For the sake of comparison, the same initial steady state was used, that is, we have $k_0 = 15$ and $k_f = 20$, with $t = 5\,\mathrm{min}$ being the first time that Λ_k is updated. All schemes reach plant optimality in about half the time of the previous case. Note also that, with all schemes, the profit (bottom panel of Figure 4) quickly reaches plant optimality. Only eMPC3 presents some erratic behavior that could result from poor numerical conditioning when Broyden's update is used on the gradient difference. However, adjusting the filtering parameter α can improve the situation.

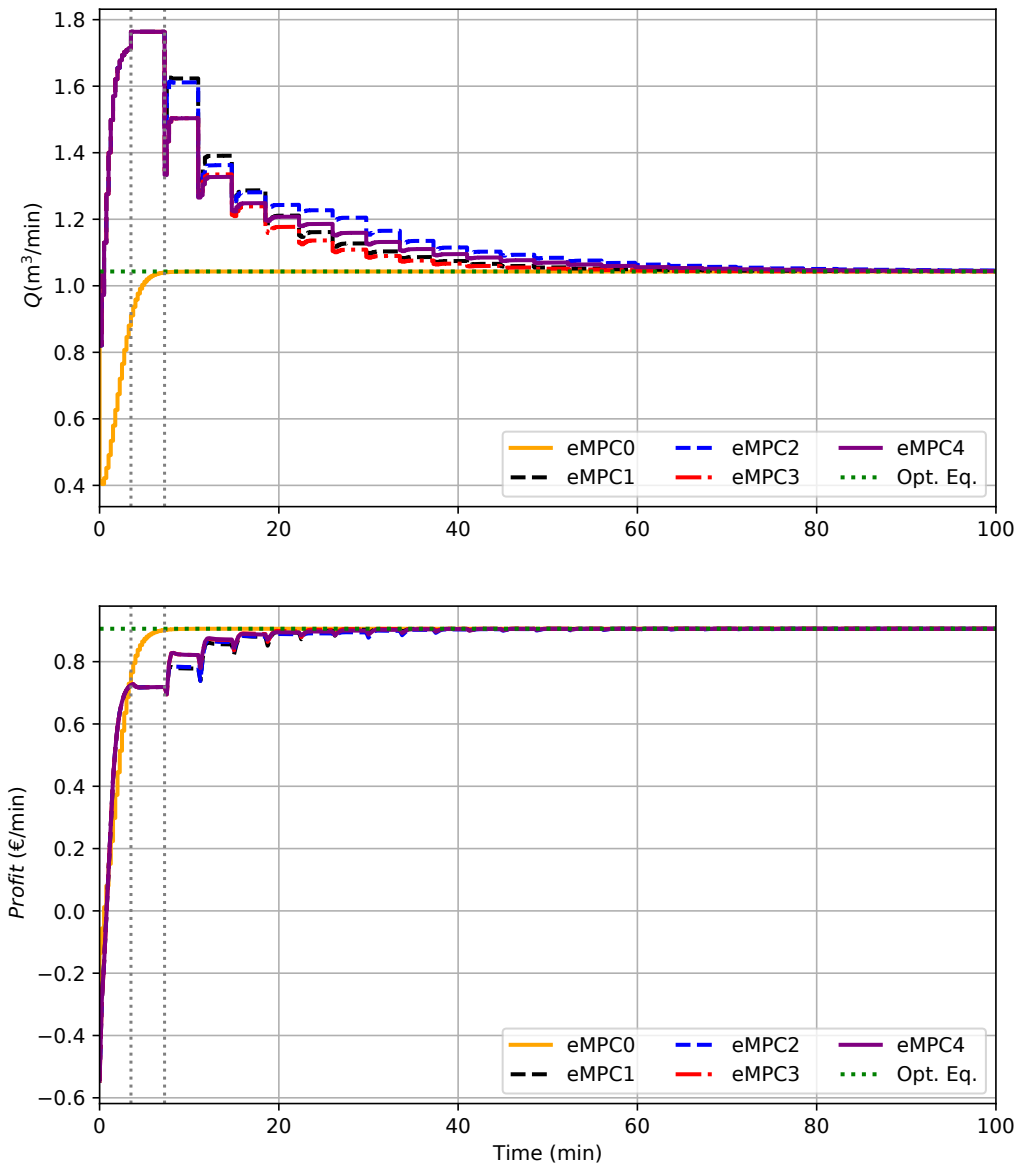

Figure 3. Reactor performance with five controllers and $M = 15$: input (**top**) and profit (**bottom**). The dotted vertical lines depict the time interval [3.75–7.5] min used for Λ-initialization.

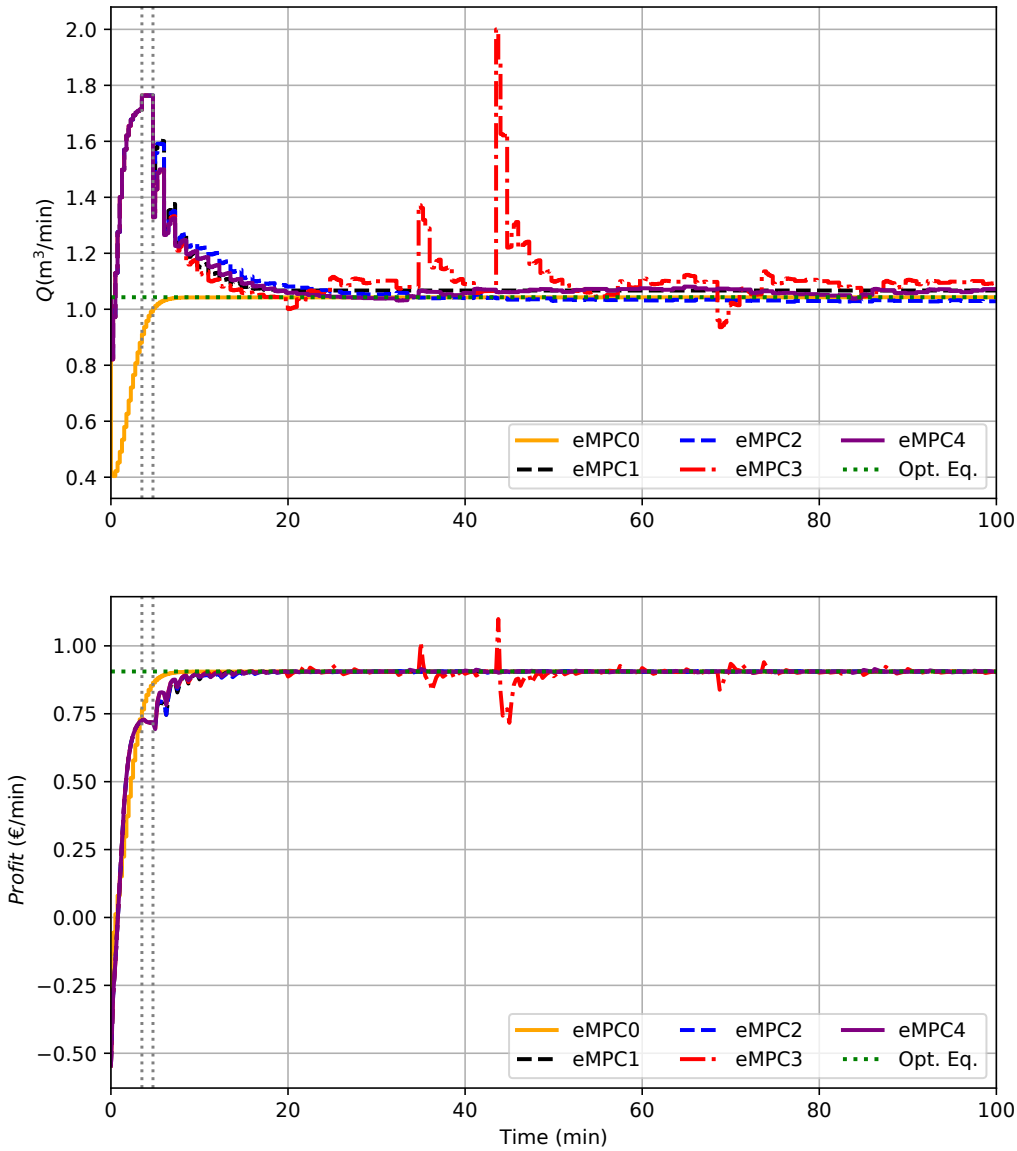

Figure 4. Reactor performance with five controllers and $M = 5$: input (**top**) and profit (**bottom**). The dotted vertical lines depict the time interval [3.75–5] min used for Λ-initialization.

5.2. Williams–Otto Reactor

5.2.1. Process

As a well-known RTO example [10,44], the Williams–Otto reactor is a nonisothermal CSTR, in which three reactions take place:

$$\begin{aligned} \mathrm{A} + \mathrm{B} &\xrightarrow{k_1} \mathrm{C} & r_1 &= k_1(T_r)c_\mathrm{A}c_\mathrm{B} \\ \mathrm{B} + \mathrm{C} &\xrightarrow{k_2} \mathrm{P} + \mathrm{E} & r_2 &= k_2(T_r)c_\mathrm{B}c_\mathrm{C} \\ \mathrm{C} + \mathrm{P} &\xrightarrow{k_3} \mathrm{G} & r_3 &= k_3(T_r)c_\mathrm{C}c_\mathrm{P}. \end{aligned} \tag{61}$$

The reactor feed consists of two components: Species A fed at the constant flow rate Q_A with molar concentration $c_{\mathrm{A}0}$, and Species B added at the variable flow rate 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 outlet flow rate Q_r is set equal to the sum of the two inlet flow rates, that is, $Q_r = Q_\mathrm{A} + Q_\mathrm{B}$, with the reactor volume V being constant. Mimicking realistic process conditions, only the molar concentrations of the two desired products are assumed to be measured, that is, $y_p = \begin{bmatrix} c_\mathrm{P} & c_\mathrm{E} \end{bmatrix}^T$. The reactor

temperature T_r is manipulated, assuming ideal cooling. The kinetic parameters depend on the reactor temperature via an Arrhenius-type law:

$$k_i(T_r) = k_{i0} \exp\left(\frac{-E_i}{T_r + 273.15}\right) \qquad \text{for } i = 1, 2, 3. \tag{62}$$

For the sake of brevity, the system dynamics are not reported here but can be found in [12]. The reactor has two inputs, namely, the temperature T_r and the flow rate Q_B. The profit to be maximized is

$$p(\cdot) = Q_r c_P p_P + Q_r c_E p_E - Q_A c_{A0} p_A - Q_B c_{B0} p_B, \tag{63}$$

where p_P, p_E, p_A, and p_B are the molar prices of products and reactants. The plant parameters can be found in [32] and in Table 1.

Table 1. William–Otto reactor: Plant parameters

Parameter	Value	Unit
k_{10}	9.9594×10^6	$m^3/(kmol \cdot min)$
k_{20}	8.66124×10^9	$m^3/(kmol \cdot min)$
k_{30}	9.9594×10^6	$m^3/(kmol \cdot min)$
E_1	6666.7	K
E_2	8333.3	K
E_3	11,111	K
c_{A0}	10	$kmol/m^3$
c_{B0}	10	$kmol/m^3$
V	2.105	m^3
Q_A	112.35	m^3/min
p_A	7.623	\$/kmol
p_B	11.434	\$/kmol
p_P	114.338	\$/kmol
p_E	5.184	\$/kmol

5.2.2. Model

The model used for control design comprises only *two* reactions:

$$\begin{array}{lcll} A + 2B & \xrightarrow{\tilde{k}_1} & P + E & \tilde{r}_1 = \tilde{k}_1(T_r) c_A c_B^2 \\ A + B + P & \xrightarrow{\tilde{k}_2} & G & \tilde{r}_2 = \tilde{k}_2(T_r) c_A c_B c_P, \end{array} \tag{64}$$

for which the kinetic parameters are reported in Table 2.

Table 2. Williams–Otto reactor: Model parameters

Parameter	Value	Unit
$\tilde{k}_{10}$	1.3134×10^8	$m^6/\left(kmol^2 \cdot min\right)$
$\tilde{k}_{20}$	2.586×10^{13}	$m^6/\left(kmol^2 \cdot min\right)$
$\tilde{E}_1$	8077.6	K
$\tilde{E}_2$	12,438.5	K

We can infer from Equation (64) that they are 5 model states,

$$x = \begin{bmatrix} c_A & c_B & c_P & c_E & c_G \end{bmatrix}^T,$$

while the plant has 6 states,

$$x_p = \begin{bmatrix} c_A & c_B & c_C & c_P & c_E & c_G \end{bmatrix}^T.$$

In addition, the inputs are bounded as follows:

$$0.18\ \mathrm{m}^3/\mathrm{min} \leq Q_B \leq\ 0.36\ \mathrm{m}^3/\mathrm{min} \tag{65}$$

$$75\ ^\circ\mathrm{C} \leq T_r \leq 100\ ^\circ\mathrm{C}. \tag{66}$$

5.2.3. Reactor Optimization

The sampling time in this example is $\tau = 2$ min, which calls for $M\tau = 30$ min between two successive modifier updates. In this case, with $\tau_{st} = 35$ min, the choice $M = 15$ is also quite appropriate. The performance of the five controllers is depicted in Figure 5, together with the optimal operating point that is unknown to the controllers. Λ-initialization is performed in the time interval of 30–90 min. The starting point for the model state vector is

$$x_0 = \begin{bmatrix} 2 & 2 & 1.1 & 1 & 0.6 \end{bmatrix}^T,$$

and the first measurement from the plant reads $y_{p,0} = \begin{bmatrix} 1.1 & 0.6 \end{bmatrix}^T$.

A behavior similar to that shown in Figure 3 can be observed, namely, (i) all schemes reach plant optimality; (ii) before the initialization phase, the practical schemes eMPC1-4 behave alike since the modifier matrix is still zero. Then, from $k_0\tau = 30$ to $k_f\tau = 90$ min, the two inputs are excited individually and sequentially as described in Section 4.1. Once a nonzero modifier matrix has been computed, we see a difference between the schemes that use plant-gradient estimation (eMPC1 and eMPC2) and those that estimate the gradient differences directly (eMPC3 and eMPC4). In this example, the schemes that rely on plant-gradient estimation do not work well: eMPC1 and eMPC2 fail to reach plant optimality; in addition, eMPC2 converges slowly, while the second input with eMPC1 gets stuck to its upper bound. In contrast, eMPC3 and eMPC4 are able to reach plant optimality. These results indicate that the two schemes based on estimating the gradient differences give the best performance.

In order to check the robustness of these last two schemes, we propose to corrupt the measurements with white noise, i.e., Plant (1) is modified as follows:

$$\begin{aligned} x_p^+ &= f_p(x_p, u) \\ y_p &= h_p(x_p) + R_v v, \end{aligned} \tag{67}$$

where $v \in \mathbb{R}^{n_y}$ is a white noise signal with covariance $R_v = 10^{-3}\,\mathrm{I}_{n_y}$. Figure 6 shows the effect of noise on the convergence performance of eMPC3 and eMPC4. It is seen that linear regression handles measurement noise better than Broyden's update since the former uses measurements stemming from several measurement points and therefore possesses a stronger filtering effect. Note that eMPC3 and eMPC4 perform very similarly right after the initialization phase when the number of measurements points is the same. As shown in the top panels of Figure 6, as time advances and more measurements become available to eMPC4, the inputs are able to move much quicker than with eMPC3. Another illustration of the improved resilience of eMPC4 with respect to measurement noise is shown by the profit profile in the bottom panel of Figure 6.

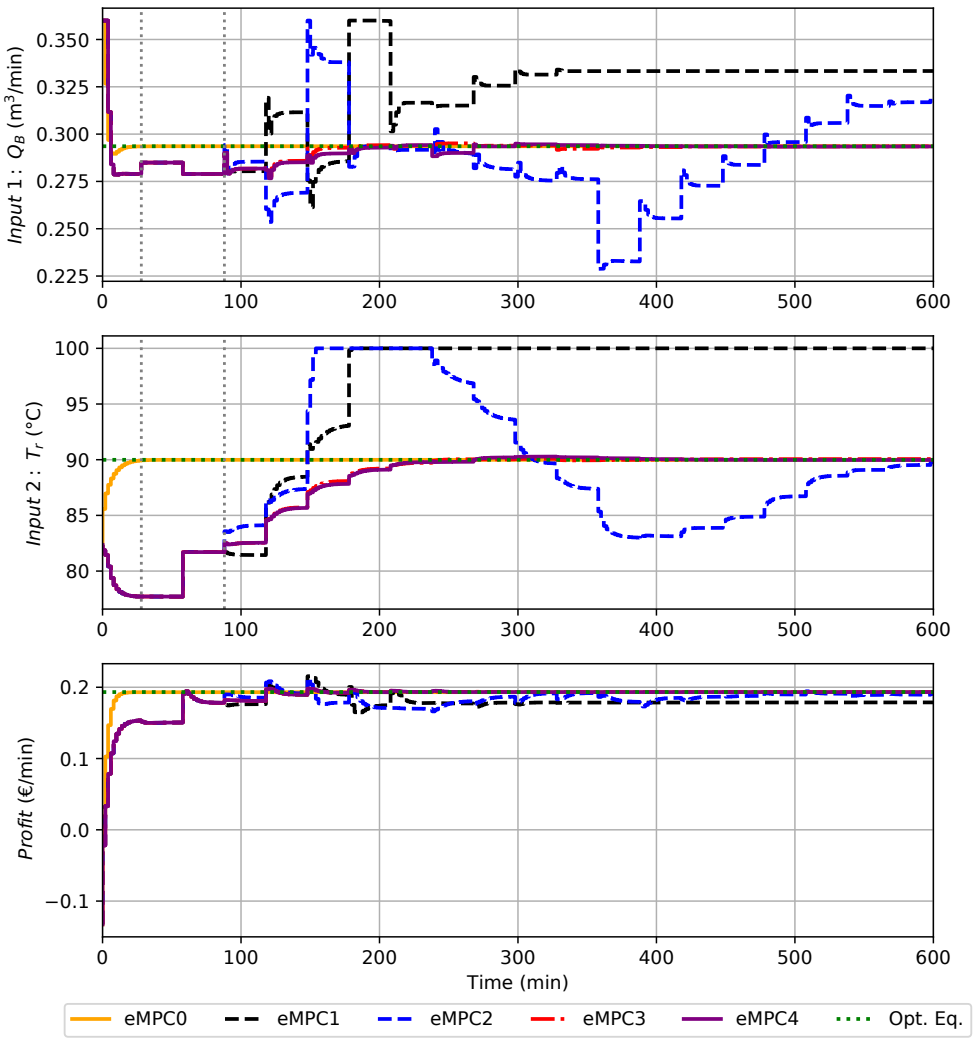

Figure 5. Performance of the Williams–Otto reactor with five controllers and no measurement noise: inputs (**top** and **middle**) and profit (**bottom**). The dotted vertical lines depict the time interval of 30–90 min used for Λ-initialization.

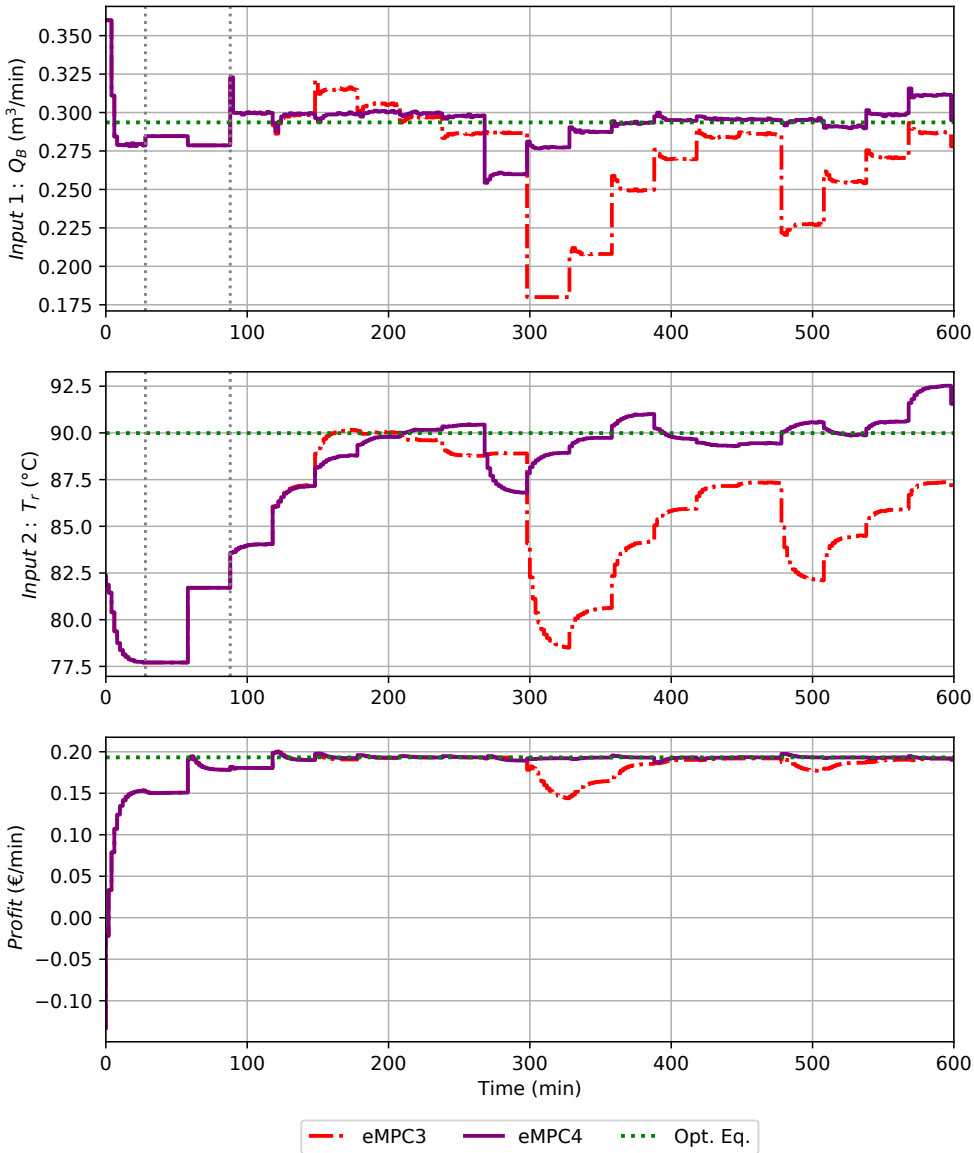

Figure 6. Performance of the Williams–Otto reactor with eMPC3 and eMPC4 in the presence of measurement noise: inputs (**top** and **middle**) and profit (**bottom**). The dotted vertical lines depict the time interval of 30–90 min used for Λ-initialization.

5.3. Final Considerations

An interesting result is the improvement in estimation quality when the modifiers are estimated directly rather than computed from the estimated plant gradients and the predicted model gradients. This is related to the nature of the estimation error. The gradient estimation error encompasses two terms resulting from truncation error and measurement noise, respectively (see [9] for a more comprehensive discussion). These two error types are quite different in nature, with the former being systematic and the latter being random. On this basis, one can consider two different ways of estimating the modifiers:

1. Estimate the plant gradients from measurements, compute the model gradients "analytically" from the model, and evaluate the modifiers as their differences, as per Section 3.2. This approach is very sensitive to truncation errors, as these errors affect the estimation of plant gradients but not of model gradients.
2. Estimate the plant gradients from measurements, estimate the model gradients from the model using the *same* numerical scheme, and evaluate the modifiers as their differences. This is equivalent to estimating the modifiers directly from measurements

(see Section 3.3). In this case, the truncation errors tend to cancel out upon computing the differences between plant and model gradients.

6. Conclusions

Different offset-free economic model predictive control schemes that combine an augmented model structure and first-order modifiers on the outputs have been proposed in recent years. These schemes proved to converge asymptotically to plant optimality, even under plant–model mismatch. The main caveat of this formulation is the requirement of perfect knowledge of static plant gradients.

This paper has investigated two different techniques to estimate plant gradients using steady-state measurements, namely, Broyden's update and linear regression. These techniques have been adapted and applied to estimate either the plant gradient or the modifiers directly. Consequently, four different approaches have been tested via two examples: in the first (single-input) example, all methods give satisfactory results, and also when the time between modifier updates is reduced to boost convergence speed. In the second (multi-input) example, only the approach that estimates the modifiers directly succeeded in reaching plant optimality. The approach was tested using both Broyden's update and linear regression. It was found that the approach that uses linear regression is clearly superior in handling measurement noise. In general, although the most effective estimation method should be chosen according to the specific case, it appears from these results that estimating the modifiers directly is more effective than estimating the plant gradient alone. Furthermore, in the presence of measurement noise, modifier estimation is best implemented via linear regression.

Note that these findings are not the end of the story, as this investigation was limited to plant gradient estimation using as many *steady-state* measurements as possible ($M = 15$ in this study). As mentioned in the introduction, one would like to be able to estimate static plant gradients using transient data, so as to speed up convergence even more (ideally, $M = 1$). Hence, a similar type of study to analyze the performance of eMPC in the presence of plant–model mismatch and the availability of *transient* measurements is very welcome. Finally, in the present paper, closed-loop stability has been assumed as per Assumption 5. However, the stability of eMPC under plant–model mismatch is still an open issue, which may motivate future research.

Author Contributions: Conceptualization, D.B. and G.P.; methodology, M.V., F.P., and G.P.; software, F.P. and M.V.; validation, M.V., D.B., and G.P.; investigation, M.V. and F.P.; data curation, M.V. and F.P.; writing—original draft preparation, M.V.; writing—review and editing, M.V., D.B., and G.P.; visualization, M.V.; supervision, G.P.; project administration, G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rawlings, J.B.; Angeli, D.; Bates, C.N. Fundamentals of economic model predictive control. In Proceedings of the 51st IEEE Conference on Decision and Control, Maui, HI, USA, 10–13 December 2012; pp. 3851–3861.
2. Morari, M.; Arkun, Y.; Stephanopoulos, G. Studies in the synthesis of control structures for chemical processes: Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures. *AIChE J.* **1980**, *26*, 220–232. [CrossRef]
3. Engell, S. Feedback control for optimal process operation. *J. Process Control* **2007**, *17*, 203–219. [CrossRef]
4. Ellis, M.; Christofides, P.D. On finite-time and infinite-time cost improvement of economic model predictive control for nonlinear systems. *Automatica* **2014**, *50*, 2561 – 2569. [CrossRef]

5. Pannocchia, G.; Rawlings, J.B. Disturbance models for offset-free model-predictive control. *AIChE J.* **2003**, *49*, 426–437. [CrossRef]
6. Muske, K.R.; Badgwell, T.A. Disturbance modeling for offset-free linear model predictive control. *J. Process Control* **2002**, *12*, 617–632. [CrossRef]
7. Pannocchia, G. Offset-free tracking MPC: A tutorial review and comparison of different formulations. In Proceedings of the European Control Conference, Linz, Austria, 15–17 July 2015; pp. 527–532.
8. Marchetti, A.G.; Chachuat, B.; Bonvin, D. Modifier-adaptation methodology for real-time optimization. *Ind. Eng. Chem. Res.* **2009**, *48*, 6022–6033. [CrossRef]
9. Marchetti, A.G.; Chachuat, B.; Bonvin, D. A dual modifier-adaptation approach for real-time optimization. *J. Process Control* **2010**, *20*, 1027–1037. [CrossRef]
10. Marchetti, A.; François, G.; Faulwasser, T.; Bonvin, D. Modifier adaptation for real-time optimization—Methods and applications. *Processes* **2016**, *4*, 55. [CrossRef]
11. Vaccari, M.; Pannocchia, G. A Modifier-Adaptation Strategy towards Offset-Free Economic MPC. *Processes* **2016**, *5*, 2. [CrossRef]
12. Faulwasser, T.; Pannocchia, G. Towards a unifying framework blending RTO and economic MPC. *Ind. Eng. Chem. Res.* **2019**, *58*, 13583–13598. [CrossRef]
13. Vergara-Diertrich, J.; Ferramosca, A.; Mirasierra, V.; Normey-Rico, J.E.; Limon, D. A Modifier-Adaptation Approach to the One-Layer Economic MPC. *IFAC-PapersOnLine* **2020**, *53*, 6957–6962. [CrossRef]
14. Papasavvas, A.; de Avila Ferreira, T.; Marchetti, A.G.; Bonvin, D. Analysis of output modifier adaptation for real-time optimization. *Comput. Chem. Eng.* **2019**, *121*, 285–293. [CrossRef]
15. Marchetti, A.G.; Faulwasser, T.; Bonvin, D. A feasible-side globally convergent modifier-adaptation scheme. *J. Process Control* **2017**, *54*, 38–46. [CrossRef]
16. Schneider, R.; Milosavljevic, P.; Bonvin, D. Accelerated and adaptive modifier-adaptation schemes for the real-time optimization of uncertain systems. *J. Process Control* **2019**, *83*, 129–135. [CrossRef]
17. Mirasierra, V.; Vergara-Diertrich, J.; Limon, D. Real-Time Optimization of Periodic Systems: A Modifier-Adaptation Approach. *IFAC-PapersOnLine* **2020**, *53*, 1690–1695. [CrossRef]
18. Papasavvas, A.; Francois, G. Internal Modifier Adaptation for the Optimization of Large-Scale Plants with Inaccurate Models. *Ind. Eng. Chem. Res.* **2019**, *58*, 13568–13582. [CrossRef]
19. Gao, W.; Wenzel, S.; Engell, S. Modifier adaptation with quadratic approximation in iterative optimizing control. In Proceedings of the European Control Conference, Linz, Austria, 15–17 July 2015; pp. 2527–2532.
20. Costello, S.; François, G.; Bonvin, D. A Directional Modifier-Adaptation Algorithm for Real-Time Optimization. *J. Process Control* **2016**, *39*, 64–76. [CrossRef]
21. Pannocchia, G. An economic MPC formulation with offset-free asymptotic performance. *IFAC-PapersOnLine* **2018**, *51*, 393–398. [CrossRef]
22. François, G.; Bonvin, D. Use of transient measurements for the optimization of steady-state performance via modifier adaptation. *Ind. Eng. Chem. Res.* **2013**, *53*, 5148–5159. [CrossRef]
23. de Avila Ferreira, T.; François, G.; Marchetti, A.G.; Bonvin, D. Use of transient measurements for static real-time optimization. *IFAC-PapersOnLine* **2017**, *50*, 5737–5742. [CrossRef]
24. Hernández, R.; Engell, S. Economics Optimizing Control with Model Mismatch Based on Modifier Adaptation. *IFAC-PapersOnLine* **2019**, *52*, 46–51. [CrossRef]
25. Vaccari, M.; Pannocchia, G. Implementation of an economic MPC with robustly optimal steady-state behavior. *IFAC-PapersOnLine* **2018**, *51*, 92–97. [CrossRef]
26. Singhal, M.; Marchetti, A.G.; Faulwasser, T.; Bonvin, D. Active directional modifier adaptation for real-time optimization. *Comput. Chem. Eng.* **2018**, *115*, 246–261. [CrossRef]
27. Marchetti, A.; de Avila Ferreira, T.; Costello, S.; Bonvin, D. Modifier adaptation as a feedback control scheme. *Ind. Eng. Chem. Res.* **2020**, *59*, 2261–2274. [CrossRef]
28. Singhal, M.; Marchetti, A.G.; Faulwasser, T.; Bonvin, D. A note on efficient computation of privileged directions in modifier adaptation. *Comput. Chem. Eng.* **2020**, *132*, 106524. [CrossRef]
29. Jeong, D.H.; Lee, C.J.; Lee, J.M. Experimental gradient estimation of multivariable systems with correlation by various regression methods and its application to modifier adaptation. *J. Process Control* **2018**, *70*, 65–79. [CrossRef]
30. Navia, D.; Briceño, L.; Gutiérrez, G.; De Prada, C. Modifier-Adaptation Methodology for Real-Time Optimization Reformulated as a Nested Optimization Problem. *Ind. Eng. Chem. Res.* **2015**, *54*, 12054–12071. [CrossRef]
31. Oliveira-Silva, E.; de Prada, C.; Navia, D. Dynamic optimization integrating modifier adaptation using transient measurements. *Comput. Chem. Eng.* **2021**, *149*, 107282. [CrossRef]
32. Vaccari, M.; Pelagagge, F.; Dominique, B.; Pannocchia, G. Estimation technique for offset-free economic MPC based on modifier adaptation. *IFAC-PapersOnLine* **2020**, *53*, 11251–11256. [CrossRef]
33. Maeder, U.; Borrelli, F.; Morari, M. Linear offset-free model predictive control. *Automatica* **2009**, *45*, 2214–2222. [CrossRef]
34. Morari, M.; Maeder, U. Nonlinear offset-free model predictive control. *Automatica* **2012**, *48*, 2059–2067. [CrossRef]
35. Pannocchia, G.; Gabiccini, M.; Artoni, A. Offset-free MPC explained: Novelties, subtleties, and applications. *IFAC-PapersOnLine* **2015**, *48*, 342–351. [CrossRef]

36. Rawlings, J.B. and Allan, D.A. Moving horizon estimation. *Encycl. Syst. Control* **2020**, 1–7. doi: 10.1007/978-1-4471-5102-9_4-2. [CrossRef]
37. Angeli, D.; Amrit, R.; Rawlings, J.B. On average performance and stability of economic model predictive control. *IEEE Trans. Autom. Control* **2012**, *57*, 1615–1626. [CrossRef]
38. Vaccari, M.; Pannocchia, G. A performance monitoring algorithm for sustained optimal operation with economic MPC. In Proceedings of the 2019 18th European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 3353–3358.
39. Roberts, P.D. Broyden derivative approximation in ISOPE optimising and optimal control algorithms. In Proceedings of the 11th IFAC Workshop on Control Applications of Optimisation, Saint Petersburg, Russia, 3–6 July 2000; pp. 283–288.
40. Rodger, E.A.; Chachuat, B. Design methodology of modifier adaptation for on-line optimization of uncertain processes. In Proceedings of the IFAC World Congress, Milan, Italy, 28 August–2 September 2011; pp. 4113–4118.
41. Dennis, J.E., Jr.; Schnabel, R.B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*; SIAM: Philadelphia, PA, USA, 1996; Volume 16.
42. Kelley, C.T. *Iterative Methods for Optimization*; SIAM: Philadelphia, PA, USA, 1999.
43. Marchetti, A.G. A new dual modifier-adaptation approach for iterative process optimization with inaccurate models. *Comput. Chem. Eng.* **2013**, *59*, 89–100. [CrossRef]
44. Williams, T.J.; Otto, R.E. A generalized chemical processing model for the investigation of computer control. *Trans. Am. Inst. Electr. Eng. Part I Commun. Electron.* **1960**, *79*, 458–473. [CrossRef]