Computers and Chemical Engineering 95 (2016) 10–20

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

# Economic model predictive control of chemical processes with parameter uncertainty

Omar Santander, Ali Elkamel, Hector Budman*

*Chemical Engineering Department, University of Waterloo, Waterloo, ON N2L3G1, Canada*

ARTICLE INFO

*Article history:*
Received 7 March 2016
Received in revised form 20 August 2016
Accepted 23 August 2016
Available online 27 August 2016

*Keywords:*
Real time optimization
Economic nonlinear predictive control

ABSTRACT

This work proposes an EMPC (Economic Model Predictive Control) algorithm that integrates RTO (Real Time Optimization) and EMPC objectives within a single optimization calculation. Robust stability conditions are enforced on line through a set of constraints within the optimization problem.

A particular feature of this algorithm is that it constantly calculates a set point with respect to which stability is ensured by the aforementioned constraints while searching for economic optimality over the horizon. In contrast to other algorithms reported in the literature, the proposed algorithm does not require terminal constraints or penalty terms on deviations from fixed set points that may lead to conservatism.

Changes in model parameters over time are also compensated for through parameter updating. The latter is accomplished by including the parameters' values as additional decision variables within the optimization problem.

Several case studies are presented to demonstrate the algorithm's performance.



## 1. Introduction

Chemical Plants are designed with the task of transforming raw materials into more valuable products. These transformations must occur in the most efficient way in order to attain different goals such as maximization of product yield, minimization of the amount of contaminants or by-products, minimization of the energy employed in the process etc. Furthermore, these transformations have to be carried out under economical, physical and environmental constraints and they must be robust to variations in process settings like temperature, input flows and pressures or variations in raw material quality. To achieve these goals advanced model based controllers such as MPC are widely used since they can optimally deal with multivariable interactions while accounting for process constraints.

The conventional hierarchical control structure (see (Findeisen et al., 1980; Luyben et al., 1990)) implemented in most process industries involves an RTO (real time optimization) (Naysmith and Douglas, 1995) level above a multivariate control level realized by an MPC or other multivariable control strategy followed by lower level single-input single-output controllers (e.g. PIDs) to effect control of actuators. The RTO is generally executed to maximize a steady state economic cost with respect to steady state values of process variables that are used as set points in the lower level multivariable control strategy. Thus, the RTO provides targets (set-points) and the multivariable controller (e.g. MPC) controls the system around these targets. Although this hierarchical strategy has resulted in good performance in industrial applications there is an opportunity for improvement since chemical processes are rarely at steady state. Hence, the steady state set points calculated by the RTO and enforced by the MPC controller may not be optimal during transient scenarios.

There are several additional drawbacks related to this two layer structure. Often the RTO and MPC layers employ different models, with RTO commonly using a detailed steady state model whereas MPC generally uses simplified dynamic models which steady state values may not exactly match those calculated by the RTO algorithm. Hence the set points computed by the RTO may be sometimes unreachable by the MPC layer. Moreover, the frequency of calculation is typically different for the two layers: MPC is optimized at every sampling period whereas RTO is optimized once a new steady state has been reached. Thus, the RTO's sampling period is typically in the order of hours or even days whereas for MPC it is in the order of minutes-seconds (Ellis et al., 2014). Since industrial processes are subjected to continuous disturbances the process may never reach a steady state.

The fact that the steady state does not always correspond to the optimal economic operation (Budman and Silveston, 2008; Huang

* Corresponding author.
*E-mail address:* hbudman@uwaterloo.ca (H. Budman).

http://dx.doi.org/10.1016/j.compchemeng.2016.08.010
0098-1354/

et al., 2011, 2012; Limon et al., 2014; Budman et al., 1996) has motivated Economic Model Predictive Control (EMPC) (Ellis and Christofidies, 2013, 2014; Angeli et al., 2012). EMPC maintains many of the strengths of MPC such as the use of dynamic MIMO models, the explicit handling of constraints, feedback etc (Morari and Lee, 1999; Rawlings, 2000; Grune and Pannek, 2011). However, in contrast with conventional MPC, it directly optimizes an economic cost instead of the typical quadratic stage cost that penalizes tracking errors with respect to set-points in controlled and manipulated variables. To ensure stability most EMPC algorithms previously reported (Amrit et al., 2011; Diehl et al., 2011; Angeli et al., 2009; Rawlings et al., 2012) used terminal constraints based on a particular steady state value but these may lead to conservative results. The need to avoid terminal constraints to reduce conservatism has been identified by Heidarinejad et al. (Heidarinejad et al., 2012) that proposed a two-stage algorithm to control and optimize the system in each stage respectively. In addition to the economic cost, most previously reported EMPC methods used tracking terms in the objective function that penalize deviations in controlled and manipulated variables with respect to the chosen steady state. The calculation of the steady state has to be done off-line by the RTO level. Also, robustness of EMPC to bounded disturbances has been studied in (Heidarinejad et al., 2012) but robustness to model variations has not been explicitly studied.

In this work we propose a robust nonlinear EMPC algorithm. The motivation was to propose an EMPC algorithm that avoids some of the assumptions and constraints used in previously reported EMPC methods that may contribute to conservatism. Towards that goal the proposed algorithm has the following properties:

1 Terminal constraints (or periodic constraints) are not needed.
2 The cost is strictly an economic cost without additional terms (such as discounted costs used in previously reported studies) related to the deviations of the controlled and manipulated variables with respect to final optimal set points.
3 Robust stability is solved on-line using a polytopic model that captures model error. By solving the problem on-line potential conservatism that arises from the use of worst bounds in combination with worst inputs is avoided (Amrit et al., 2011; Diehl et al., 2011; Angeli et al., 2009; Rawlings et al., 2012).
4 Parameter updating is introduced to cope with parameter errors. Both the set point and the updating parameter value are introduced as additional decision variables in the economic optimization.

Since the proposed algorithm assumes the set-point to be a decision variable the need for an RTO calculation level is eliminated. The price for by-passing the RTO level is that stability has to be assessed with respect to a time varying set-point. Furthermore, a robust stability condition that accounts for model parametric uncertainty is computed and enforced online.

Towards that goal the EMPC algorithm is solved as an optimization problem over a receding horizon in which the nonlinear dynamic system behavior is represented by a polytopic model where its vertices are described by different operation points (Operation window) (Ellis et al., 2014; Huang et al., 2011; Kothare et al., 1996). A set of especial constraints is added so as to assure robust stability.

The organization of this paper is as follows.

In Section 2, the notation, assumptions, proposed algorithm and asymptotic robust stability are described. Simulation examples proving the performance of the algorithm are shown and discussed in Section 3 followed by conclusions in Section 4.

## 2. Definitions and methodology

### 2.1. Definitions and assumptions

**Definition 2.1.** (Positive definite function)

A function $\rho()$ is positive definite with respect to $\boldsymbol{x} = \boldsymbol{a}$ if:
*It is continuous*, $\rho(\boldsymbol{a}) = 0$ *and* $\rho(\boldsymbol{x}) > 0$ *for all* $\boldsymbol{x} \neq \boldsymbol{a}$

**Definition 2.2.** (Class $\mathfrak{K}$ function).

A function $\Upsilon() : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a class $\mathfrak{K}$ if it is continuous, zero at the origin and strictly increasing.

**Lemma 2.1.** Given a positive definite function $\rho(\boldsymbol{x})$ defined on a compact set $D$ containing the origin, there exists a class $\mathfrak{K}$ function $\Upsilon()$ such that:

$$\rho(\boldsymbol{x}) \geq \Upsilon(|\boldsymbol{x}|), \quad \forall \boldsymbol{x} \quad \in \quad \mathbb{D}$$

**Definition 2.3.** (Positive invariant set).

A set $\mathbb{A}$ is positive invariant for the discrete nonlinear system $\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k))$ if:

$$\boldsymbol{x}(k) \in \mathbb{A} \;\; \textit{and} \;\; \boldsymbol{x}(k+1) \in \mathbb{A}$$

**Definition 2.4.** (Asymptotic stability).

The steady state $\boldsymbol{xs}$ of a nonlinear discrete system $\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k))$ is asymptotically stable on $\mathbb{X}$, where $\mathbb{X}$ has $\boldsymbol{xs}$ in its interior, if there exist a $\Upsilon()$ such that for any $\boldsymbol{x} \in \mathbb{X}$, all solutions $\Phi(k;\boldsymbol{x})$ satisfy:

$$\Phi(k;\boldsymbol{x}) \in \mathbb{X},$$

$$|\Phi(k;\boldsymbol{x}) - \boldsymbol{xs}| \leq \Upsilon(|\boldsymbol{x} - \boldsymbol{xs}|, k) \quad \forall \mathrm{k} \in \mathbb{I}_{>0}$$

Where $\mathbb{I}_{>0}$ represents positive integers.

**Definition 2.5.** (Lyapunov function)

A function $V : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is said to be Lyapunov function for the nonlinear discrete system in the set $\mathbb{X}$ if there exists $\Upsilon i()$, where $i \in \{1, 2, 3\}$ such that for any $\boldsymbol{x} \in \mathbb{X}$

$$\Upsilon 1(|\boldsymbol{x}|) \leq V(\boldsymbol{x}) \leq \Upsilon 2(|\boldsymbol{x}|);$$

$$V(\mathfrak{x}(k+1)) - V(\boldsymbol{x}(k)) \leq -\Upsilon 3(|\boldsymbol{x}|)$$

**Lemma 2.2.** (Lyapunov function and asymptotic stability). Consider a set $\mathbb{X}$ that is positive invariant for the nonlinear discrete system $\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k))$. The steady state $\boldsymbol{xs}$ is an asymptotically stable equilibrium point for the system if and only if there exists a Lyapunov function $V$ on $\mathbb{X}$ such that $V$ satisfies the properties described above (Definition 2.5).

**Assumptions.**

i) The stage cost to be used by the EMPC algorithm and the nonlinear model are continuous.
ii) There is weak controllability. Therefore, there exists $\Upsilon() : \mathbb{R} \to \mathbb{R}_{\geq 0}$ so that for each $\boldsymbol{x} \in \mathbb{X}$ there exists a feasible $\boldsymbol{u}$ trajectory [u(1), u(2)...u(N)] with:

$$\sum_{k=1}^{N} |\boldsymbol{u}(k) - \boldsymbol{us}| \leq \Upsilon(|\boldsymbol{x} - \boldsymbol{xs}|)$$

Where $\boldsymbol{us}$ is the input vector corresponding to $\boldsymbol{xs}$.

### 2.2. Proposed EMPC algorithm

#### 2.2.1. Definitions

Define deviation variables as follows:

$$\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{xs} \tag{1}$$

$$\boldsymbol{u}' = \boldsymbol{u} - \boldsymbol{us} \tag{2}$$

Where $\boldsymbol{x}'$, $\boldsymbol{x} and \boldsymbol{xs}$ represent the deviation state vector, the absolute state vector and a vector of steady states' solution respectively. For the case of the input ($\boldsymbol{u}$) the definitions are the same (e.g. $\boldsymbol{u}'$ is the deviation input vector).

The deviation input (manipulated variable) vector will be calculated by a time-varying state feedback control law:

$$\boldsymbol{u}' = \boldsymbol{K}\boldsymbol{x}' \tag{3}$$

Where $\boldsymbol{K}$ is a time-varying controller gain calculated from an optimization problem defined below.

The nonlinear dynamic discrete system is defined as:

$$\boldsymbol{x}(k+1) = f\left(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\theta}(k)\right) \tag{4}$$

Where $\boldsymbol{\theta}(k)$ represents a vector of time-varying parameters. In (4), $\boldsymbol{x}(k+1)$ represents the state vector at the next time step; the sampling time T is used for discretization. In addition, the vector of states and corresponding set points (steady states) are $\boldsymbol{x} \epsilon \mathbb{X} \subseteq \mathbb{R}^n$ and $\boldsymbol{xs} \epsilon \mathbb{X}s \subset \mathbb{X}$, for the manipulated variables, $\boldsymbol{u} \epsilon \mathbb{U} \subset \mathbb{R}^m$ and $\boldsymbol{us} \epsilon \mathbb{U}s \subset \mathbb{U}$, and for the model parameters $\boldsymbol{\theta} \epsilon \mathbb{Q} \subset \mathbb{R}^q$. Where $\boldsymbol{us}$ is defined from

$$0 = f\left(\boldsymbol{xs}(k), \boldsymbol{us}(k), \boldsymbol{\theta}(k)\right).$$

The economic stage cost to be optimized is defined as:

$$L = l(\boldsymbol{x}(k), \boldsymbol{u}(k)) \tag{5}$$

#### 2.2.2. Polytopic model representation used for enforcing robust stability

To address robust stability the nonlinear system given in (4) is approximated by linearized models around different operating points and can be represented in deviation variables as follows:

$$\boldsymbol{x}'(k+1) = \boldsymbol{A}\left(\boldsymbol{\theta}\right)\boldsymbol{x}'(k) + \boldsymbol{B}\left(\boldsymbol{\theta}\right)\boldsymbol{u}'(k) \tag{6}$$

From (3) and (6):

$$\boldsymbol{x}'(k+1) = \boldsymbol{A_{CL}}\left(\boldsymbol{\theta}, k\right)\boldsymbol{x}'(k) \tag{7}$$

Where $\boldsymbol{A_{CL}}\left(\boldsymbol{\theta}, k\right) = \boldsymbol{A}\left(\boldsymbol{\theta}\right) + \boldsymbol{B}\left(\boldsymbol{\theta}\right)\boldsymbol{K}(k)$

The plant model is then represented by a convex set $\mathbb{A}_{\mathbb{T}}$ of linear plants with M vertices:

$$\mathbb{A}_{\mathbb{T}} = \left\{\boldsymbol{A_{CL1}}\left(\boldsymbol{\theta}, k\right), \boldsymbol{A_{CL2}}\left(\boldsymbol{\theta}, k\right) \ldots \boldsymbol{A_{CLM}}\left(\boldsymbol{\theta}, k\right)\right\} = \sum_{i=1}^{M} \zeta_i \left[A_{CLi}\left(\boldsymbol{\theta}, k\right)\right]; \quad \forall \zeta_i \geq 0, \sum_{i=1}^{M} \zeta_i = 1 \tag{8}$$

Thus the representation (8), describes the plant model by a polytope of matrices. Each element of the polytope, $\boldsymbol{A_{CLi}}\left(\boldsymbol{\theta}\right)$, can be derived from the linearization defined in (7) performed at different operating conditions that define a window of operation of the controller as proposed in (Al-Gherwi et al., 2011). Using (8) it follows that the plant model can be approximated as follows:

$$\boldsymbol{x}'(k+1) = \sum_{i=1}^{M} \zeta_i [A_{CLi}(\boldsymbol{\theta}, k)]\boldsymbol{x}'(k) \tag{9}$$

A candidate Lyapunov function with properties as stated in Definition 2.5 is (Ellis et al., 2014; Diehl et al., 2011; Rawlings et al., 2012):

$$V(k) = \boldsymbol{x}'(k)^T \boldsymbol{P}(\boldsymbol{k})\boldsymbol{x}'(k) \tag{10}$$

Then, to enforce robust stability:

$$V_{max}(k+1) - V(k) < 0 \tag{11}$$

To ensure that this condition is satisfied for the worst case corresponding to a particular model described by the polytopic model defined in (9), $V_{max}(k+1)$ is derived from a maximization problem with respect to the parameter $\zeta$ corresponding to one particular model out of the family of models defined in (9) as follows:

$$V_{max}(k+1) = \underset{\zeta}{\text{argmin}} \left[-\boldsymbol{x}'(k+1)^T \boldsymbol{P}(\boldsymbol{k})\boldsymbol{x}'(k+1)\right]$$

s.t.

$$\boldsymbol{x}'(k+1) = \sum_{i=1}^{M} \zeta_i [A_{\boldsymbol{CLi}}(\boldsymbol{\theta}, k)]\boldsymbol{x}'(k)$$

$$\sum_{i=1}^{M} \zeta_i = 1$$

$$0 \leq \zeta_i \leq 1$$

Condition (11) is introduced as a constraint in the proposed EMPC algorithm described in the following section.

#### 2.2.3. Proposed robust EMPC formulation

The proposed robust EMPC algorithm is defined by the following optimization problem:

$$\min_{\boldsymbol{us}, \boldsymbol{K}, \boldsymbol{P}, \boldsymbol{\theta}} \sum_{k=1}^{N} l(\boldsymbol{x}(k), \boldsymbol{u}(k)) \tag{12}$$

s.t.

$$\boldsymbol{x}(0) = \boldsymbol{xo} = \boldsymbol{x_{meas}} \tag{12.a}$$

$$\boldsymbol{x}(k+1) = f\left(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\theta}(k)\right) \tag{12.b}$$

$$\boldsymbol{u_l} \leq \boldsymbol{u} \leq \boldsymbol{u_h} \tag{12.c}$$

$$0 = f\left(\boldsymbol{xs}(k), \boldsymbol{us}(k), \boldsymbol{\theta}(k)\right) \tag{12.d}$$

$$\boldsymbol{P}_{max} \geq \boldsymbol{P}(k) > 0 \tag{12.e}$$

$$(\boldsymbol{x}(k) - \boldsymbol{xs}(k))^T \boldsymbol{P}(k-1)(\boldsymbol{x}(k) - \boldsymbol{xs}(k)) < \alpha(\boldsymbol{x}(k-1) - \boldsymbol{xs}(k-1))^T \boldsymbol{P}(k-1)(\boldsymbol{x}(k-1) - \boldsymbol{xs}(k-1)) \quad 0 < \alpha < 1 \tag{12.f}$$

$$(\boldsymbol{x}(k) - \boldsymbol{xs}(k))^T \boldsymbol{P}(k)(\boldsymbol{x}(k) - \boldsymbol{xs}(k)) \leq (\boldsymbol{x}(k) - \boldsymbol{xs}(k))^T \boldsymbol{P}(k-1)(\boldsymbol{x}(k) - \boldsymbol{xs}(k)) \tag{12.g}$$

$$V_{max} = \underset{\zeta}{\text{argmin}} \left[-[\boldsymbol{x}(k+1) - \boldsymbol{xs}(k)]^T \boldsymbol{P}(k)[\boldsymbol{x}(k+1) - \boldsymbol{xs}(k)]\right] \tag{12.h}$$

s.t.

$$\begin{gathered} \boldsymbol{x}(k+1) - \boldsymbol{xs}(k) = \sum_{i=1}^{M} \zeta_i [A_{\boldsymbol{CLi}}(\boldsymbol{\theta}, k)][\boldsymbol{x}(k) - xs(k)] \\ \boldsymbol{A_{CLi}}(\boldsymbol{\theta}, k) = \boldsymbol{A_i}(\boldsymbol{\theta}) + \boldsymbol{B_i}(\boldsymbol{\theta})\boldsymbol{K}(k) \\ \sum_{i=1}^{M} \zeta_i = 1 \quad 0 \leq \zeta_i \leq 1 \\ V_{max} - [\boldsymbol{x}(k) - \boldsymbol{xs}(k)]^T \mathrm{P}(k)[\boldsymbol{x}(k) - \boldsymbol{xs}(k)] < 0 \end{gathered} \tag{12.i}$$

Where N represents the finite horizon and $\boldsymbol{xo}$ is the initial state respectively which is also chosen as the current measured value from the process ($\boldsymbol{x_{meas}}$) to introduce feedback. The steady state and parameters are kept constant for the entire horizon length N, implying that $\boldsymbol{us}(1) = \boldsymbol{us}(2) = \ldots \boldsymbol{us}(N)$ and $\boldsymbol{\theta}(1) = \boldsymbol{\theta}(2) = \ldots \boldsymbol{\theta}(N)$. $\boldsymbol{A_i}$ and $\mathbf{B}_i$ are obtained from linearization of the nonlinear system given by (4) around different operating conditions as explained in Section 2.2.2. The vector $\boldsymbol{u}s$ is defined by the steady state Eq. 12.d. In Eq. (12.f) α is a tuning parameter that determines the rate of convergence of the Lyapunov function (see Theorem 1) that increases as α is smaller.

It should be noticed that the steady state may change at each execution of the EMPC controller since the manipulated variables' vector at steady state **u**s is a decision variable. This is a key novelty in our proposed algorithm since previously proposed approaches have used a set point obtained from an off-line steady state optimization. This dynamic optimization problem is solved in a receding horizon fashion, implying that the first element of the solution vector $\boldsymbol{u}(k) = \boldsymbol{u}'(k) + \boldsymbol{u}s(k) = [\boldsymbol{u}(1,k), \boldsymbol{u}(2,k), \boldsymbol{u}(3,k)\ldots\boldsymbol{u}(N,k)]$, namely $\boldsymbol{u}(1,k)$, and the computed $\boldsymbol{\theta}(1,k)$ are applied to the dynamic discrete system. Then, after applying this control action, the vector of states $\boldsymbol{x}(k+1,k)$ is used at the next step for re-solving the problem when the first element of the vector $\boldsymbol{x}(k+1,k)$ is measured and becomes available.

Thus the control action is:

$$\boldsymbol{u}(k)^* = \boldsymbol{u}(1,k) \tag{13}$$

The constraints 12.f-12.i are used to ensured robust stability with respect to the polytopic model described in Section 2.2.2 (constraint 12.i) and time variations in $\boldsymbol{P}(k)$ (constraint 12.g) and the set point $\boldsymbol{x}s(k)$ (constraint 12.f). The robust stability and feasibility of the proposed algorithm are discussed below.

### 2.3. Robust stability and feasibility of the proposed algorithm

**Theorem 1.** (Stability of the EMPC algorithm described in (12):

The EMPC algorithm described by the optimization problem (12) is robustly stable:

$$\lim_{t\to\infty} (\boldsymbol{x}(tk) - \boldsymbol{x}s(k)) = 0 \tag{14}$$

**Proof.** Noticing that the LHS of 12.f is equal to the RHS of 12.g implies

$$\begin{aligned}&(\boldsymbol{x}(k) - \boldsymbol{x}s(k))^T \boldsymbol{P}(k)(\boldsymbol{x}(k) - \boldsymbol{x}s(k))\\ &< \alpha(\boldsymbol{x}(k-1) - \boldsymbol{x}s(k-1))^T \boldsymbol{P}(k-1)(\boldsymbol{x}(k-1) - \boldsymbol{x}s(k-1))\end{aligned} \tag{15}$$

Since $\boldsymbol{P}$(k) is positive definite and bounded according to (12.d) and $0 < \alpha < 1$ then the function defined in (10) is:

$$V(k) = (\boldsymbol{x}(k) - \boldsymbol{x}s(k))^T \boldsymbol{P}(k)(\boldsymbol{x}(k) - \boldsymbol{x}s(k)) = \boldsymbol{x}'(k)^T \boldsymbol{P}(k)\boldsymbol{x}'(k) \tag{16}$$

Then, under the assumptions and definition in Lemma 2.2, *V(k)* is a Lyapunov function according to Definition 2.5 and thus $\lim_{k\to\infty} \|x(k) - \boldsymbol{x}s(k)\| = 0$. Then, from (12.h) and (12.i) it follows that for the worst case with respect to the polytopic model in (8):

$$\underset{\zeta}{\operatorname{argmin}}[-\boldsymbol{x}'(k+1)^T \boldsymbol{P}(k)\boldsymbol{x}'(k+1)] < V(k) \tag{17}$$

Since *V(k)* is a Lyapunov function, from (17), $\lim_{k\to\infty} \|x(k+1) - xs(k)\| = 0$ is ensured for all models included in the polytopic model given in (8). Then, since from Eq. (15) also $\lim_{k\to\infty} \|x(k) - xs(k)\| = 0$, it can be concluded that $\boldsymbol{x}(k+1) = \boldsymbol{x}(k) = xs(k)$ and accordingly the set point also converges $xs(k+1) = xs(k)$.

A key difference of the proposed algorithm is that stability is ensured on-line rather than in an off-line fashion as done in previous approaches by using dissipativity arguments. A main feature of this on-line approach is that it does not require restrictive terminal conditions and the use of fixed steady state solutions as required in the off-line stability proofs previously reported. On the other hand, the on-line imposition of the stability constraints requires significantly longer computation time as further discussed in the section of Results.

**Remark 1.** (Feasibility of the algorithm in (12)).

The feasibility of the steady state constraint (12.d) is ensured if there is at least one possible steady state that complies with the input constraints (12.c).

In terms of the feasibility of the additional constraints a distinction has to be made between open loop stable or open loop unstable systems.

For open loop stable systems, the eigenvalues of all the open loop models included in the polytopic model in (8) are stable. Then, constraint (12.i) can always be satisfied for $\boldsymbol{K}(k) = 0$, (see Eq. (7)). If 12.i is satisfied in open loop, then (12.f) can be satisfied at least for the particular case **x**$s(k)$=**x**$s(k-1)$ since it can be easily verified that 12.i and 12.f are identical for the case of constant set point $\boldsymbol{x}s(k)$. Constraint 12.g can be always trivially satisfied at least for the case **P**(k) = **P**(k − 1).

For open loop unstable systems there are two options. First, a simple stabilizing controller can be implemented and the set point of this controller can then be regulated by the proposed EMPC algorithm. Alternatively, an off-line maximization of an admissible region of initial conditions can be calculated using Linear Matrix Inequalities ((Boyd et al., 1994)) but this is beyond the scope of the current study.

### 2.4. Penalization of steady state in the cost function to speed up convergence

Due to the longer computation times required for the on-line enforcement of the stability related constraints and since the optimization was executed with fmincon in the Matlab environment, the prediction horizon N could not be chosen to be very large. It is expected that the use of faster hardware and software (e.g. IPOPT (Watcher and Biegler, 2006)) or the use of distributed strategies will permit the use of longer prediction horizons.

A disadvantage related to the choice of smaller N is that the system mostly penalizes the economic cost over a short term horizon. Then, as the deviation variables approach zero, the updating rate of the set point as per the constraint given by Eq. (12.f) becomes very small thus slowing down the convergence to the steady state economic optimum. To speed-up convergence, a steady state economic cost term is added with a corresponding weight.

Considering the definitions of deviation variables, the stage cost with the addition of the steady state cost term is as follows:

$$\min_{\boldsymbol{u}s,\boldsymbol{K},\boldsymbol{P}} \sum_{k=1}^{N} l(\boldsymbol{x}'(k), \boldsymbol{u}'(k)) + Ql(\boldsymbol{x}s(k), \boldsymbol{u}s(k)) \tag{18}$$

$Q$ is a tunning parameter which has values of $Q \geq 1$. Note that for linear cost functions when $Q = 1$, the stage cost reduces to the stage cost in (5) since the components of the cost in each of the terms in (18) that are dependent on $\boldsymbol{x}s$ and $\boldsymbol{u}s$ cancel out.

Although a systematic approach for choosing Q is beyond the scope of the current work, simulations are presented in the case studies with different values of Q to illustrate its effect on the closed loop responses. It should be noticed that in contrast with previously reported algorithms, the added penalty term is only necessary to speed up convergence to the optimal steady state but it is not necessary for enforcing stability.

### 2.5. Parameter convergence

Parameter convergence in case that the parameters are updated for the general formulation (12) can be also analytically proven by simple arguments as given by the following proposition.

**Proposition 1.** Let assumptions (i) and (ii) hold. If the dynamic optimization problem (12) is considered, at steady state $\boldsymbol{\theta}_{plant} = \boldsymbol{\theta}_{model}$

**Proof.** Considering the steady state equation, $0 = f\left(\boldsymbol{x}s(k), \boldsymbol{u}s(k), \boldsymbol{\theta}(k)\right)$, that can be equivalently written as:

$$\boldsymbol{x}s(k) = g\left(\boldsymbol{u}s(k), \boldsymbol{\theta}(k)\right) \tag{19}$$

Also:

$$\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), \boldsymbol{x}s(k), \boldsymbol{K}) \tag{20}$$

From Theorem 1 the deviation variables,

$$\boldsymbol{x}'(k+1) = \boldsymbol{x}(k+1) - \boldsymbol{x}s(k+1) \quad \text{and} \quad \boldsymbol{x}'(k) = \boldsymbol{x}(k) - \boldsymbol{x}s(k) \tag{21}$$

converge to zero and upon convergence as shown in Theorem 1:

$$\boldsymbol{x}(k) = \boldsymbol{x}(k+1) = \boldsymbol{x}s(k) = \boldsymbol{x}_{meas} \tag{22}$$

Since $\boldsymbol{x}_{meas}$ is the first value in the prediction vector as explained after (12), then $\boldsymbol{\theta}_{model} = \boldsymbol{\theta}_{plant}$, because this is the only possibility for which the measured value $\boldsymbol{x}_{meas}$ will equate the model predictions assuming that the model structure is correct.

### 2.6. Optimal economic steady state

For the purpose of comparisons in the case studies the best steady state can be calculated as follows. If $\boldsymbol{u}s$ and $\boldsymbol{x}s$ represents the best steady state pair then $0 = f\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_0\right)$ defines the steady state corresponding to a particular set of parameters $\boldsymbol{\theta}_0$. Accordingly, the optimal steady state problem may be stated as:

$$\min_{\boldsymbol{u}} l(\boldsymbol{x}s, \boldsymbol{u}s)$$

s.t.

$$0 = f\left(\boldsymbol{x}s, \boldsymbol{u}s, \boldsymbol{\theta}_0\right) \tag{23}$$

$$\boldsymbol{u_l} \leq \boldsymbol{u} \leq \boldsymbol{u_h}$$

Where $\boldsymbol{u} = \boldsymbol{u}s$ *and* $\boldsymbol{x} = \boldsymbol{x}s$. It is assumed that there is a unique solution. The result of this optimization problem is used in the case studies to test whether the dynamic EMPC algorithm converges to the best economic steady state.

## 3. Case studies (Simulation examples)

As explained in the previous section a key goal of the proposed algorithm was to avoid the use of terminal constraints or discounted cost which can contribute to conservatism while ensuring robust stability with respect to parametric uncertainty. In the presence of uncertainty the algorithm is capable to update model parameters' values. Both the set points and parameters' values are used as decision variables for maximization of the economic cost.

Three case studies are presented to illustrate these properties. The first two case studies involve an isothermal CSTR where parallel reactions are occurring. In the first case study the parameters are assumed to be accurately known a priori whereas in the second case study one of the parameters is not known a priori and therefore it has to be updated on-line by the proposed algorithm. The third case study deals with a William-Otto reactor represented by a system of 6 nonlinear differential equations.

In the first case study the objective was to show the basic implementation of the algorithm and to compare it to a previously reported technique that uses a discounted cost and terminal constraints. From that comparison the goal is to assess whether the proposed algorithm can reduce conservatism in certain scenarios. Also, the impact of the weight Q on the steady state optimum was investigated.

The second study where initial parameter error is assumed, the objective was to test robust stability in the presence of this error and convergence of the parameter to its actual value. In the third case study, the objective was to test the capability of the algorithm to deal with a system of larger dimensions than the one used in case studies 1 and 2.

### 3.1. Case study 1

The proposed EMPC is applied to a reactor system example presented in (Rawlings et al., 2012). The reactor consists of an isothermal CSTR with parallel reactions.

The dimensionless nonlinear mass balance equations are described by:

$$\frac{dx_1}{dt} = u_1 - x_1 - \sigma_1 x_1 x_2 \tag{24}$$

$$\frac{dx_2}{dt} = u_2 - x_2 - \sigma_1 x_1 x_2 - \sigma_2 x_2 x_3 \tag{25}$$

$$\frac{dx_3}{dt} = -x_3 + \sigma_1 x_1 x_2 - \sigma_2 x_2 x_3 \tag{26}$$

$$\frac{dx_4}{dt} = -x_4 + \sigma_2 x_2 x_3 \tag{27}$$

Where $x_1, x_2, x_3, x_4$ represent the concentrations of $P_0$, B,$P_1$,$P_2$ respectively and $u_1, u_2$ represent the flow rates of $P_0$ and B. For this example the parameters $\sigma_1$ and $\sigma_2$ are assumed constant and have a value of 1 and 0.4 respectively. The inputs $u_i$ have upper and lower bounds as follows:

$$0 \leq u_1 \leq 1 \tag{28}$$

$$0 \leq u_2 \leq 10 \tag{29}$$

The best economic steady state for these bounds computed by the off-line optimization given in (23) is as follows:

$$\boldsymbol{x}s = [0.3874, 1.5811, 0.3752, 0.2373]^T \text{and} \boldsymbol{u}s = [1, 2.4310]^T$$

As mentioned above this steady state value is given for comparison with the steady state value reached by the proposed algorithm upon convergence but the algorithm uses time varying values of $\boldsymbol{x}s$ in the calculation of control actions.

We assume the same sampling time (T = 0.1) as described in (Rawlings et al., 2012).

For this particular formulation, the economic stage cost is obtained from the assumption that profit is directly related to the concentration of $P_1$ which is the most valuable product. Thus, the cost to be maximized is:

$$l = -x_3$$

The original dynamic optimization problem can be stated as follows:

$$\min_{\boldsymbol{u}s, \boldsymbol{K}, \boldsymbol{P}} \sum_{k=1}^{N} -x_3 \tag{30}$$

s.t.

$$\begin{aligned}
&\boldsymbol{x}(0) = \boldsymbol{x}o \\
&\boldsymbol{x}(k+1) = f\left(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\theta}(k)\right) \\
&0 = f\left(\boldsymbol{x}s(k), \boldsymbol{u}s(k), \boldsymbol{\theta}(k)\right) \\
&0 \leq u_1 \leq 1 \\
&0 \leq u_2 \leq 10
\end{aligned}$$

**Table 1**
Performance comparison between our formulation and alternative.

| EMPC | Average Cost | % Improvement |
|---|---|---|
| Q1 | 0.37739 | 0.00958 |
| Q2 | 0.37759 | 0.06392 |
| EMPC-A | 0.37735 | 0 |

Constraints 12.e-12.i

As explained in Section 2.4, to speed up convergence to the best economic optimum an economic steady state cost multiplied by a tuning parameter Q is added as follows:

$$\min_{\boldsymbol{us},\boldsymbol{K},\boldsymbol{P}} \sum_{k=1}^{N} -\left(x'_3 + Qxs_3\right) \tag{31}$$

s.t.
$\boldsymbol{x}(0) = \boldsymbol{x}o$
$\boldsymbol{x}(k+1) = f\left(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\theta}(k)\right)$
$0 = f\left(\boldsymbol{xs}(k), \boldsymbol{us}(k), \boldsymbol{\theta}(k)\right)$
$0 \le u_1 \le 1$
$0 \le u_2 \le 10$
Constraints 12.e-12.i

The EMPC formulation was analyzed for two cases with different values of $Q$ ($Q1$=45, $Q2$=6). The horizon had a value of 15 for both cases. The problem was solved in Matlab using the function *fmincon* for optimization and *ode45* and *fsolve* to solve the differential equations and the steady state values respectively. The algorithm presented in (Rawlings et al., 2012) that uses terminal constraints and a modified cost to ensure stability is also implemented for comparison with our formulation that does not use terminal constraints and/or modified cost. In the discussion below the algorithm reported in (Rawlings et al., 2012) will be referred thereafter as EMPC-A. The values of terminal constraints and parameters in the stage cost for EMPC-A were taken from that manuscript. The dynamic optimization problem was solved using the same horizon (15). The Matlab functions *fmincon* was used for nonlinear optimization and *ode45* for integrating the nonlinear differential equations. The deviation for each state was [-0.0845, −0.585, 0.077, 0.2635] and it was chosen arbitrarily but large enough to clearly illustrate the features of the proposed algorithm. The suboptimal steady state used for defining deviation variables at the beginning of the simulation was [0.3845, 1.5850, 0.3730, 0.2365].

Fig. 1 shows the closed loop state profiles for an initial condition (sub optimal), different values of $Q$ ($Q1$ = 45, $Q2$ = 6) and the alternative formulation (Rawlings et al., 2012). In the figures, the variables ending in 's' represent the steady state, 'A' represents the EMPC-A, 't' represents time and 'BSS' refers to the best steady state. The deviation from steady state assumed at time=0 can be viewed as equivalent to the introduction of a disturbance.

From Fig. 1 is evident that the closed loop behavior of our algorithm with $Q1$ and the EMPC-A exhibit very similar responses for all the states but the dynamic behaviour is significantly different with $Q2$ for states $x_1$ and $x_2$. However, regardless of the value of $Q$ the closed loop responses for $x_3$ converges to the same value that corresponds to the best economic steady state calculated from (23).

Fig. 2 shows the closed loop inputs profiles for the two values of $Q$ and for EMPC-A. The flow rate (u1) is identical in all the simulations. On the other hand, while for *Q1*, u2 is similar for the proposed algorithm and for EMPC-A, u2 is different for *Q2* as compared to EMPC-A. Thus, the results with the proposed algorithm can be significantly different as compared to EMPC-A.

Table 1 shows the economic performance in terms of Average Cost i.e. the average of the objective function over the entire simulation time for the proposed algorithm and for EMPC-A. The profit

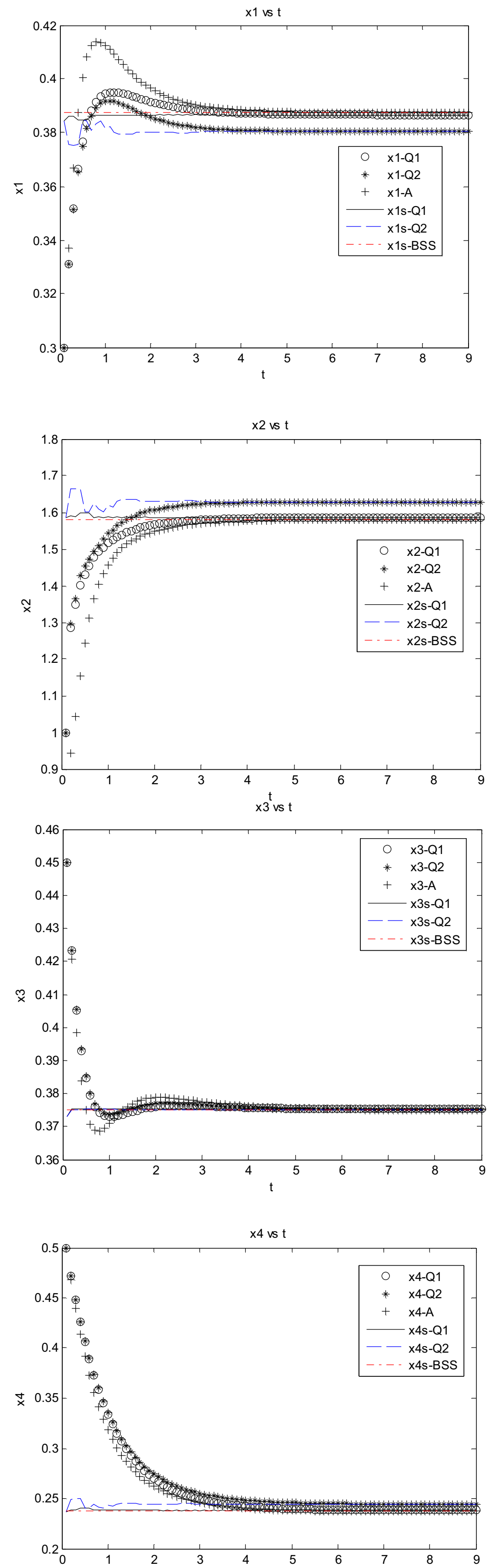
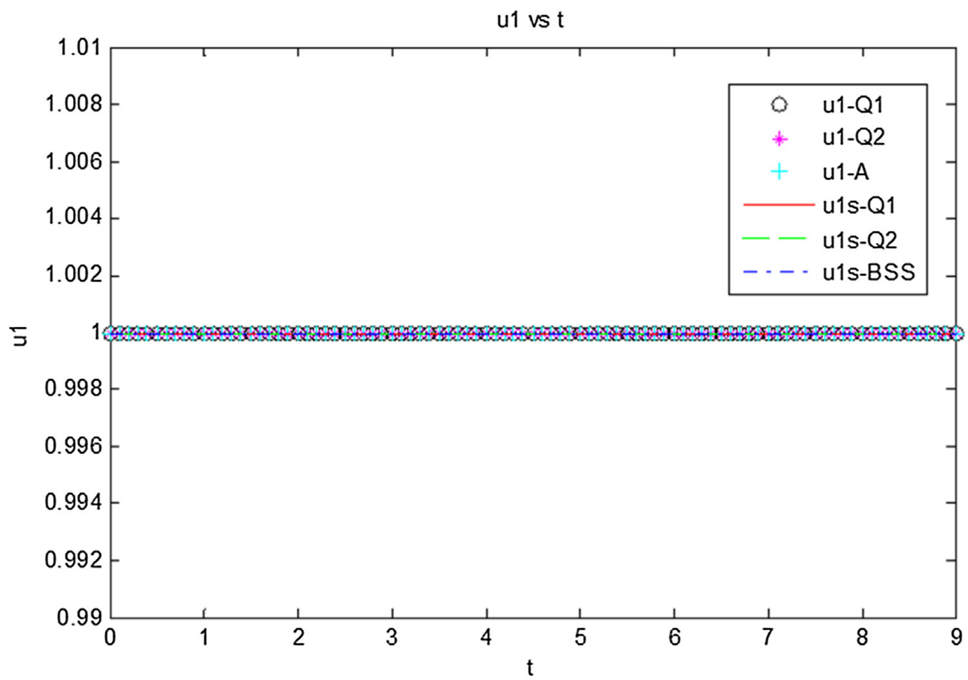



**Fig. 1.** Evolution of state variables with time for case study 1 (Legend $x_i$-A refers to state i obtained with the EMPC-A algorithm that was used for comparison, Legend $x_i$-BSS refers to the best steady state value of state i obtained with the optimization described in Eq. (23), Legends $x_i$-Qi refers to the state 1 using weight Qi to penalize the steady state economic cost as shown in Eq. (18)).



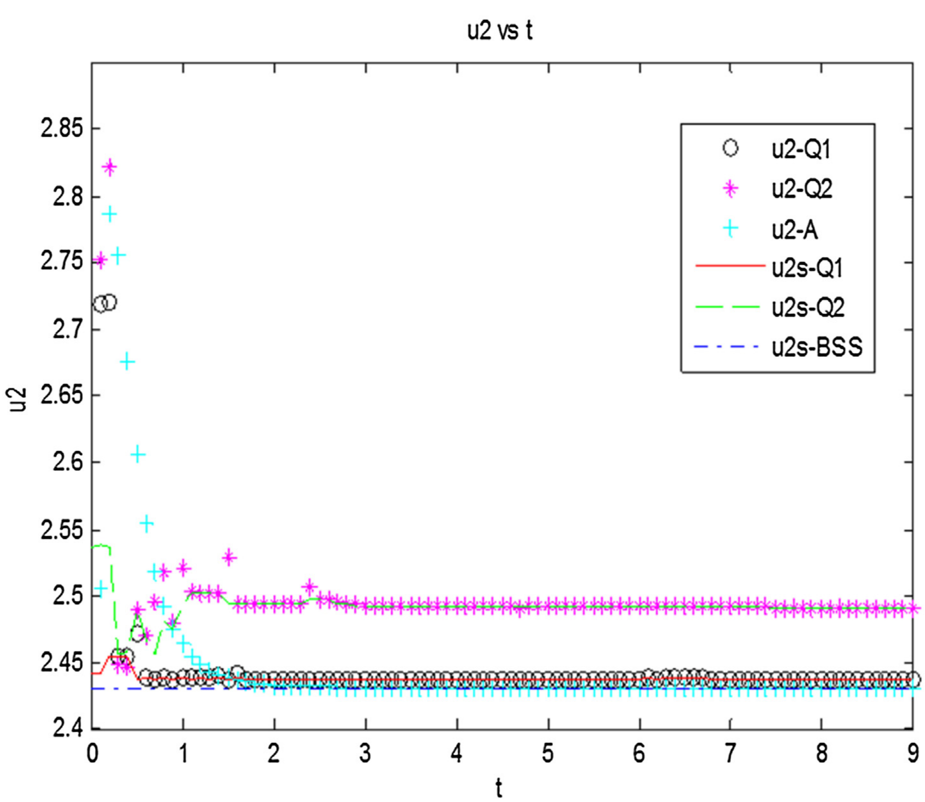




**Fig. 2.** Evolution of input (manipulated) variables with time (Legend $u_i$-A refers to input i obtained with the EMPC-A algorithm that was used for comparison, Legend $u_i$-BSS refers to the best steady state value of input i obtained with the optimization described in Eq. (23), Legends $u_i$-Qi refers to the state i using weight Qi to penalize the steady state economic cost as shown in Eq. (18)).

obtained with our approach in these examples are slightly higher or similar to the EMPC-A.

For example, from Table 1, it was expected that for $Q1$, where the steady state is strongly penalized in the cost, our results will be very similar to EMPC-A since the latter involves penalty terms on the deviations from the final steady state. On the other hand for $Q2$, where the steady state term in the cost function is penalized by a lower value, our approach is slightly better. This slight improvement is due to the fact that in our approach the steady state values are used as additional decision variables thus adding degrees of freedom for optimization.

Finally, the online closed loop behavior of the Lyapunov function ($\boldsymbol{x}'(k)^T \boldsymbol{P}(k) \boldsymbol{x}'(k)$) for the different values of $Q$ is shown in Fig. 3 to illustrate compliance with the robust stability related constraints.

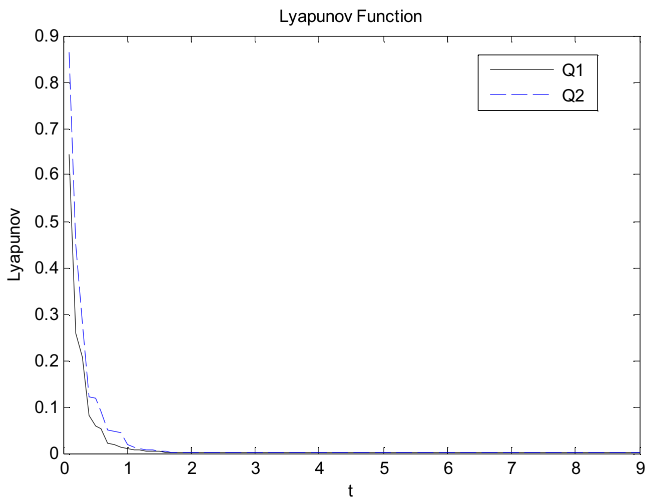



**Fig. 3.** Evolution of the Lyapunov function with time for case 1 using the proposed algorithm with weight Q1 to penalize the steady state cost.

This figure also confirms that the polytopic model used to formulate the robust stability condition (constraint 12.h) is able to properly approximate the original nonlinear system by a set of models that are obtained by linearization as explained in Section 2.2.2.

### *3.2. Case study 2: nonlinear reactors with parameter updating*

The problem treated in this case study was the same as in case study 1. However, in this study it was assumed that the exact value of $\sigma_1$ was unknown a priori and instead it was bounded as follows:

$$1 \le \sigma_1 \le 1.03 \tag{32}$$

The simulations were conducted with the tuning parameter $Q1 = 45$. The resulting dynamic optimization problem is as follows:

$$\min_{\boldsymbol{us},\boldsymbol{K},\boldsymbol{P},\sigma_1} -(x'_3 + Q1xs_3)\sum_{k=1}^{N} \tag{33}$$

s.t.
$\boldsymbol{x}(0) = \boldsymbol{xo}$
$\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), \boldsymbol{u}(k), \sigma_1(k))$
$0 = f(\boldsymbol{xs}(k), \boldsymbol{us}(k), \sigma_1(k))$
$0 \le u_1 \le 1$
$0 \le u_2 \le 10$
Constraints 12.e-12.i

As explained in Section 2 $\alpha$ is a tuning parameter that determines the rate of convergence of the Lyapunov function that increases as $\alpha$ gets smaller. For this problem $\alpha$ was chosen to be either $\alpha_1 = 0.9$ or $\alpha_2 = 0.8$.

The closed loop responses of the system for the parameter adaptation are shown in Figs. 4 and 5. Fig. 4 shows the closed loop state trajectories for a particular initial condition where an initial mismatch exists between the model and the plant in terms of the value of the parameter $\sigma_1$. The figure describes the corresponding trajectories for both $\alpha_1 = 0.9$ or $\alpha_2 = 0.8$. It is evident from Fig. 4c that the cost, i.e. $x_3$, converges much faster to the target for the smaller value of $\alpha$, i.e. for $\alpha_2 = 0.8$. Also, this figure shows that regardless of the value of $\alpha$, the cost $x_3$ converges to the same value.

In Fig. 5, the closed loop input trajectories, the evolution of the adapting parameter and the evolution of the Lyapunov function are shown for both $\alpha_1 = 0.9$ or $\alpha_2 = 0.8$.

From Fig. 4, it is evident that the optimizer initially chooses the upper bound for the parameter since at the initial stage it results in more profit. The initial value for $\sigma_1$ assumed for the model was

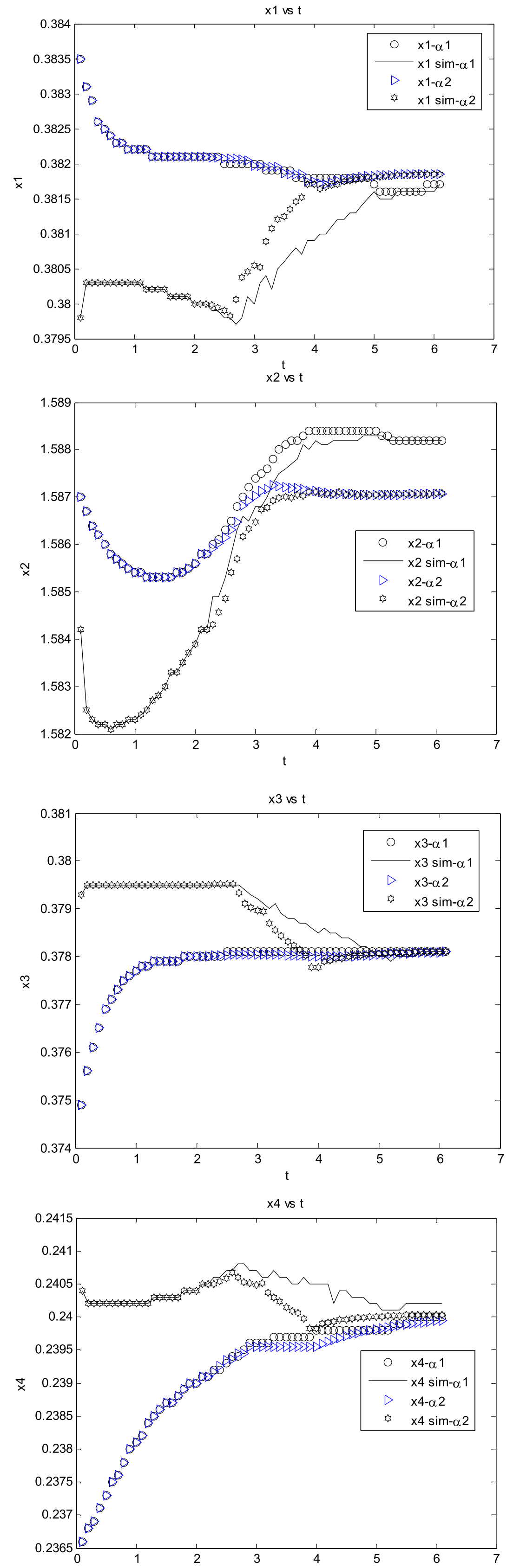



**Fig. 4.** Evolution of the actual states and the simulated states as a function of time for case study 2 where the parameter $\sigma_1$ is in error with respect to the actual value under control by the proposed robust EMPC algorithm (parameter updating is realized by including the parameter value as an additional decision variable for optimization as described in Eqs. (12) and (33)).

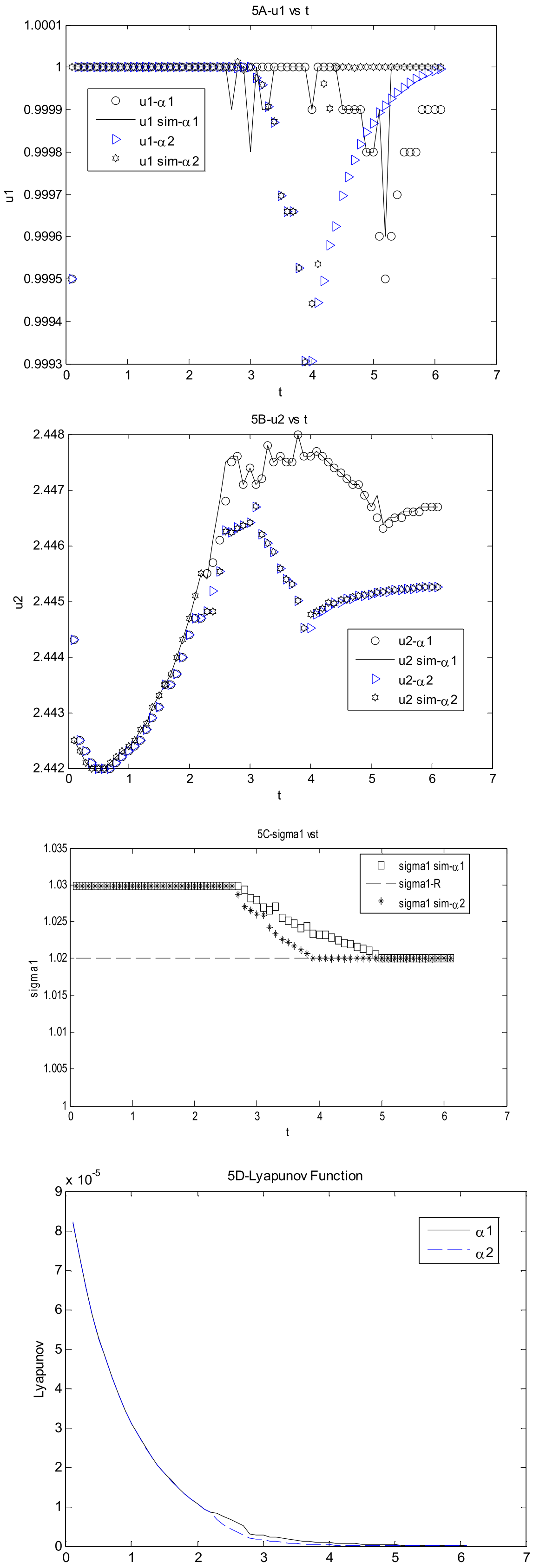



**Fig. 5.** (A and B) evolution of manipulated variables with time using the proposed algorithm for case study 2, (C) evolution of the updating parameter $\sigma_1$ with time (sigma1-sim) as compared to the actual value (sigma1-R), (D) evolution of the Lyapunov function with time using the proposed algorithm for case study 2.

1 and the true value in the plant was 1.02. This relatively small difference is enough to change the new best steady state and as a consequence the optimal operating point. The optimizer computes the upper bound of $\sigma_1$ until approximately t = 3 at which time the set-point constraint (12.f) forces the optimizer to choose a smaller value of $\sigma_1$ in order to satisfy this constraint. From t = 3 and on, the set-point constraint forces the optimizer to continuously decrease the value of $\sigma_1$ until the true value is reached. It is also important to point out that there are some time intervals where the value for $\sigma_1$ was temporarily increased with respect to the previous iteration but nevertheless the optimization problem still remained feasible. Also, Fig. 5C clearly shows that the smaller value of α, i.e. for $\alpha_2 = 0.8$, results in faster convergence of the updating parameter to the true value.

This case study illustrates three key points of the proposed algorithm: i- the updating parameter converges to the true value, ii- the system is robustly stable, i.e. maintains stability despite the initial parameter error, iii- the cost and the parameter converge faster to their respective actual values for smaller values of α.

The monotonic decrease of the Lyapunov function calculated with the actual states' values again corroborates robust stability and that the real system is correctly approximated by the polytopic model representation used for the purpose of enforcing robust stability (constraint 12.h).

### 3.3. Case study 3 (Williams-Otto reactor)

The proposed approach is applied to the Williams-Otto reactor presented in (Amrit, 2011). Since this system is of higher dimensions than the system used in the previous two case studies, this example shows the ability of the algorithm to cope with the added numerical complexity as compared to the previous case studies.

The mass balances are derived from standard modeling assumptions and it is the following:

$$W\frac{dx_A}{dt} = F_A - (F_A + F_B)x_A - r_1 \tag{34}$$

$$W\frac{dx_B}{dt} = F_B - (F_A + F_B)x_B - r_1 - r_2 \tag{35}$$

$$W\frac{dx_C}{dt} = -(F_A + F_B)x_C + 2r_1 - 2r_2 - r_3 \tag{36}$$

$$W\frac{dx_E}{dt} = -(F_A + F_B)x_E + 2r_2 \tag{37}$$

$$W\frac{dx_G}{dt} = -(F_A + F_B)x_G + 1.5r_3 \tag{38}$$

$$W\frac{dx_P}{dt} = -(F_A + F_B)x_P + r_2 - 0.5r_3 \tag{39}$$

Where $x_A, x_B, x_C, x_E, x_G, x_P$ represent the mass fraction of the components in the reactions which are defined as: $r_1 = k_1x_Ax_BW$, $r_2 = k_2x_Bx_CW$ and $r_3 = k_3x_Cx_PW$. The reaction constants follow the standard Arrhenius equation as a function of the reactor temperature. Thus the current case is not only of larger dimensions than the previous problem used in case studies 1 and 2 but also the nonlinear behaviour is here more pronounced since it is of exponential type. $F_A$, $F_B$ and $W$ represent the flow rates of A, B and mass hold up respectively.

In this problem, the manipulated inputs are the flow rate of B ($F_B$) and the temperature of the reactor ($T_R$) where the latter is explicitly used in the exponential term of the Arrhenius equation. $F_A$ and $W$ are constant values. The numerical values for $F_A$, $W$ and $k_i$ were taken from (Amrit, 2011).

The stage cost was also taken from (Amrit, 2011), and it is the profit of the process calculated from the difference between the revenues of the products E and P and the costs of raw materials A and B.

$$l_P = 5554.1\,(F_A + F_B)\,x_P + 125.91\,(F_A + F_B)\,x_E - 370.3F_A - 555.42F_B \tag{40}$$

The inputs (manipulated variables) are bounded as follows:

$$2 \le F_B \le 10 \tag{41}$$

$$349 \le T_R \le 367 \tag{42}$$

The dynamic optimization problem can be stated as:

$$\min_{\boldsymbol{u}s, \boldsymbol{K}, \boldsymbol{P}} \sum_{k=1}^{N} -\left(l_P' + Ql_Ps\right) \tag{43}$$

s.t.

$$\boldsymbol{x}(0) = \boldsymbol{x}o$$

$$\boldsymbol{x}(k+1) = f\left(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\theta}(k)\right)$$

$$0 = f\left(\boldsymbol{x}s(k), \boldsymbol{u}s(k), \boldsymbol{\theta}(k)\right)$$

$$2 \le F_B \le 10$$

$$349 \le T_R \le 367$$

Constraints 12.e–12.i

Where the horizon was 15 and $Q$ was selected to be equal to 1000.

Fig. 6 shows the closed loop state trajectories for an initial condition (suboptimal steady state). The proposed EMPC computes the best steady state and is able to drive the states to this optimal steady state.

Fig. 7 shows the closed loop inputs trajectories and the Lyapunov function as functions of time.

It is clear from Fig. 7 that the plant is stable and the Lyapunov function is monotonically decreasing corroborating that the polytopic uncertainty description correctly describes the actual nonlinear system and robust stability is ensured. Furthermore, the closed loop inputs in Fig. 7 show that there are not significant deviations between the best steady state values and the dynamic values of the manipulated variables. This is because Q was selected such as $Ql_Ps \gg l_P{}'$. On the other hand, in contrast to case studies 1 and 2, in the current case study the manipulated variables at the best economic steady state are not at their bounds.

The algorithm converges to the best economic steady state as follows:

$$\boldsymbol{x}s = [0.0870, 0.3896, 0.0152, 0.2909, 0.1074, 0.1096]^T$$

$$\boldsymbol{u}s = [4.7231, 362.6852]^T$$

This is identical to the best economic steady state computed by the optimization described in (23). Due to the highly nonlinear behaviour of this process and to the larger number of states, the computational time of the algorithm was obviously larger as compared to case studies 1 and 2.

## 4. Conclusions

An Economic Model Predictive Control online algorithm was proposed that is robust to model error. The algorithm uses the set point as an additional decision variable for economic optimization and it also updates the model parameters to account for parametric errors. The proposed EMPC algorithm was successfully applied to three nonlinear reaction engineering applications and it was compared to a previously reported technique.

It was shown that the proposed EMPC algorithm may result in improvement in profit as compared to previously reported formulations that use terminal constraints and/or modified costs to satisfy dissipativity. Also, the algorithm provides robust stability

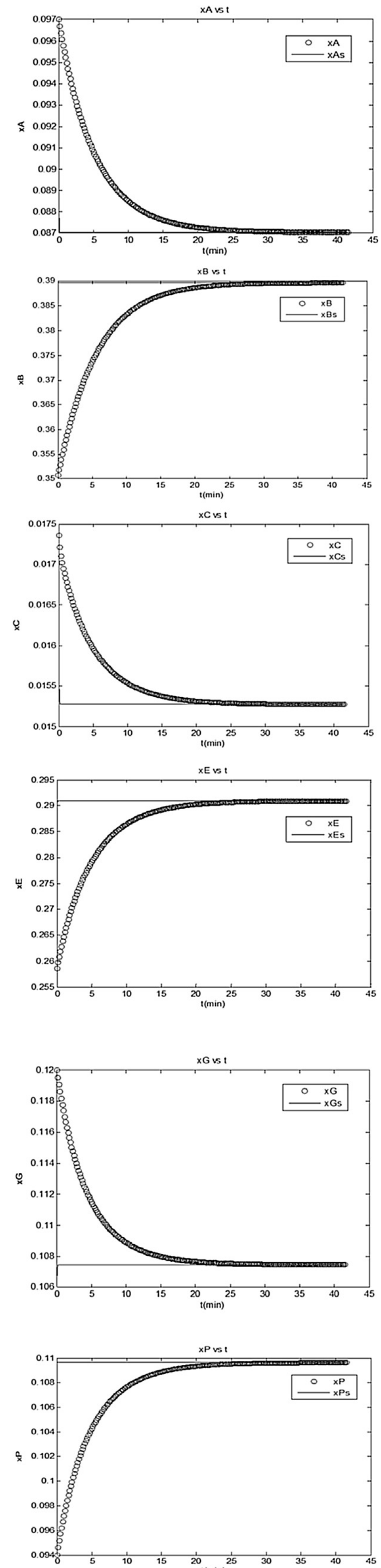



**Fig. 6.** Evolution of the states for case study 3 (Legend xI- refers to the state I under closed loop control, Legend xIs refers to the best steady state value of state I as calculated by (18)).

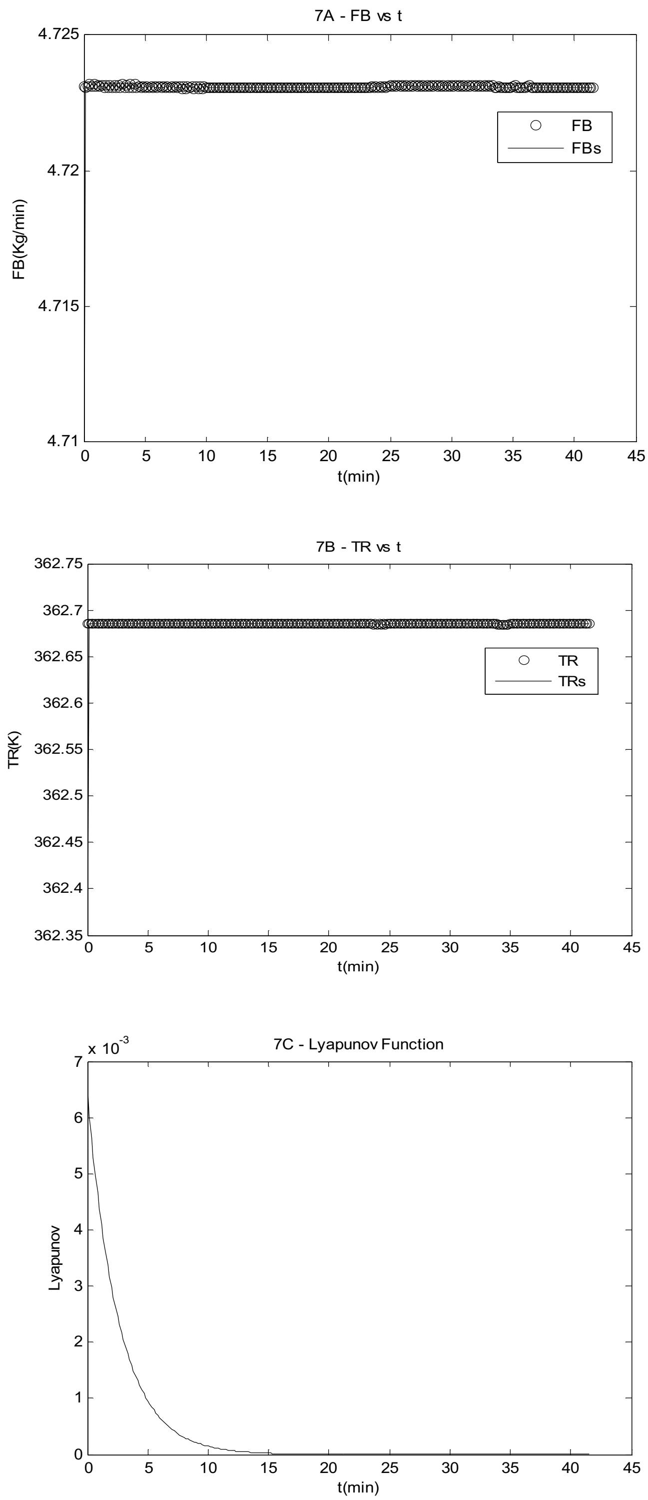



**Fig. 7.** (A and B) input values for case study 3, (C) evolution Lyapunov function with time for case study 3.

in the presence of parametric error and it can successfully update the erroneous parameters to their actual value by including these parameters as additional decision variables in the optimization. It was found that the penalization of the steady state value within the cost function using a weight (Q) helped significantly to speed up convergence when the prediction horizon is chosen relatively short to avoid expensive computations. An additional tuning parameter ($\alpha$) was introduced in one of the constraints to speed up the convergence of the parameter to its true value in case of initial parameter error. Smaller values of $\alpha$ were shown to increase the speed of convergence. Also, it was shown that regardless of the choice of Q and $\alpha$ the cost ultimately converged to the best economic optimum. On the other hand, the algorithm has high computational requirements since it enforces the robust stability conditions on-line. These limitations may be reduced in the future by the use of more effective software and hardware or by the use of a distributed computation strategy.

## Acknowledgments

The authors acknowledge financial support from CONACYT and SEP.

## Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.compchemeng.2016.08.010.

## References

Al-Gherwi, W., Budman, H., Elkamel, A., 2011. A robust distributed model predictive control algorithm. J. Process Control 21, 1127–1137.
Amrit, R., 2011. Optimizing Process Economics in Model Predictive Control, PhD Thesis. University of Wisconsin-Madison.
Amrit, R., Rawlings, J.B., Angeli, D., 2011. Economic optimization using model predictive control with a terminal cost. Annu. Rev. Control 35, 178–186.
Angeli, D., Amrit, R., Rawlings, J.B., 2009. Receding horizon cost optimization for overly constrained nonlinear plants. 48st IEEE Conference on Decision and Control.
Angeli, D., Amrit, R., Rawlings, J.B., 2012. On average performance and stability of economic model predictive control. IEEE Trans. Autom. Control 57, 1615–1626.
Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V., 1994. Linear Matrix Inequalities in Systems and Control Theory. SIAM, Philadelphia, PA.
Budman, H., Silveston, P.L., 2008. Control of periodically operated reactors. Chem. Eng. Sci. 63, 4942–4954.
Budman, H., Kzyonsek, M., Silveston, P., 1996. Control of a nonadiabatic packed bed reactor under periodic flow reversal. Can. J. Chem. Eng. 74, 751–759.
Diehl, M., Amrit, R., Rawlings, J.B., 2011. Lyapunov function for economic optimizing model predictive control. IEEE Trans. Autom. Control 56, 703–707.
Ellis, M., Christofidies, P.D., 2013. Economic model predictive control with time varying objective function for nonlinear process systems. AIChE J. 60, 507–519.
Ellis, M., Christofidies, P.D., 2014. Integrating dynamic economic optimization and model predictive control for optimal operation of nonlinear process systems. Control Eng. Pract. 22, 242–251.
Ellis, M., Durand, H., Christofidies, P.D., 2014. A tutorial review of economic model predictive control methods. J. Process Control 24, 1156–1178.
Findeisen, W., Bailey, F.N., Bryds, M., Malinowski, K., Tatjewski, P., Wozniak, A., 1980. Control and Coordination in Hierarchical Systems. John Wiley & Sons, New York.
Grune, L., Pannek, J., 2011. Nonlinear Model Predictive Control: Theory and Algorithms. Springer-Verlag, London.
Heidarinejad, M., Liu, J., Christofides, P.D., 2012. Economic model predictive control of nonlinear process systems using Lyapunov techniques. AIChE J. 58, 855–870.
Huang, R., Harinath, E., Biegler, L.T., 2011. Lyapunov stability of economically oriented NMPC for cyclic processes. J. Process Control 21, 501–509.
Huang, R., Biegler, L.T., Harinath, E., 2012. Robust stability of economically oriented infinite horizon NMPC that include cyclic processes. J. Process Control 22, 51–59.
Kothare, M.V., Balakrishnan, V., Morari, M., 1996. Robust constrained model predictive control using linear matrix inequalities. Automatica 32, 1361–1379.
Limon, D., Pereira, M., Munoz de la Pena, D., Alamo, T., Grosso, J.M., 2014. Single-layer economic model predictive control for periodic operation. J. Process Control 24, 1207–1224.
Luyben, W.L., Tyreus, B., Luyben, M.L., 1990. Plantwide Process Control. McGraw Hill, New York.
Morari, M., Lee, J.H., 1999. Model predictive control: past, present and future. Comput. Chem. Eng. 23, 667–682.
Naysmith, M.R., Douglas, P.L., 1995. Review of real time optimization in the chemical process industries. Asia-Pac. J. Chem. Eng. 3, 67–87.
Rawlings, J.B., 2000. Tutorial overview of model predictive control. IEEE Control Syst. Mag. 20, 38–52.
Rawlings, J.B., Angeli, D., Bates, C.N., 2012. Fundamentals of economic model predictive control. 51st IEEE Conference on Decision and Control.
Watcher, A., Biegler, L.T., 2006. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programing. Math. Program. 106, 25–57.