Fast Sensitivity-based Economic Model Predictive Control for Degenerate Systems

Eka Suwartadi[a], Vyacheslav Kungurtsev[2b], Johannes Jäschke[a,*]

[a] *Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway*
[b] *Department of Computer Science, Czech Technical University in Prague, 1200 Praha 2, Czech Republic*

Abstract

We present a sensitivity-based nonlinear model predictive control (NMPC) algorithm and demonstrate it on a case study with an economic cost function. In contrast to existing sensitivity-based approaches that make strong assumptions on the underlying optimization problem (e.g. the linear independence constraint qualification implying unique multiplier), our method is designed to handle problems satisfying a weaker constraint qualification, namely the Mangasarian-Fromovitz constraint qualification (MFCQ). Our nonlinear programming (NLP) sensitivity update consists of three steps. The first step is a corrector step in which a system of linear equations is solved. Then a predictor step is computed by a quadratic program (QP). Finally, a linear program (LP) is solved to select the multipliers that give the correct sensitivity information. A path-following scheme containing these steps is embedded in the advanced-step NMPC (asNMPC) framework. We demonstrate our method on a large-scale case example consisting of a reactor and distillation process. We show that LICQ does not hold and the path-following method is able to accurately approximate the ideal

*Corresponding author
Email addresses: eka.suwartadi@ntnu.no (Eka Suwartadi), vyacheslav.kungurtsev@fel.cvut.cz (Vyacheslav Kungurtsev), jaschke@ntnu.no (Johannes Jäschke)

[1]E.S and J.J acknowledge the funding from Norwegian Research Council through IKT-PLUSS Young Researcher program
[2]VK was supported by the OP VVV project CZ.02.1.01/0.0/0.0/16 019/0000765 Research Center for Informatics

solutions generated by an NLP solver.

Keywords: numerical optimal control, NLP sensitivity, economic MPC, path-following.

1. Introduction

The recent progress in nonlinear model predictive control (NMPC) provides the possibility of designing controllers that directly optimize an economic performance index. This is known as *economic NMPC* [1]. Some of the latest advancements of economic NMPC are documented in a textbook [2] and a recent survey article [3]. Most of the contributions in economic NMPC consider the stability analysis. The work of [4] started the stability analysis of economic NMPC using the notion of rotated stage costs and the strong duality. Later, dissipativity [5] and *input-to-state stability* (ISS) [6] were used for this purpose. Other approaches can be found, e.g., in [7], [8], [9], [10], [11], [12], [13]. Distributed economic NMPC approaches were discussed in [14] and [15], and tube-based robust economic NMPC was proposed in [16].

Contributions to implementation and computational aspects for economic NMPC are fewer compared to those of stability analysis papers, although the problem of large computational delay is well documented. Among others, Idris and Engell [17] noted very long computation times in their realistic case studies on economic NMPC. When applying NMPC, the updated optimal solution has to be available within a very short time, because otherwise the controller acts on outdated information, which leads to performance loss, or even instability [18].

To minimize the time delay in NMPC, sensitivity-based approaches have proven to be very successful. Sensitivity-based approaches for fast NMPC exploit the fact that the optimization problem solved at any NMPC iteration is identical to the optimization problem at the previous iteration, except for the updated initial state. Therefore, the sensitivity of the optimal solution [19] can be used to obtain fast approximations of the solution.

Early work related to sensitivity-based methods includes the Newton-type controller by Li and Biegler [20], where a QP approximation of the optimal solution is solved at given sample times. Sensitivity was first used in the NMPC context by Diehl et al. [21] in the development of the real-time iteration method. Since then methods have been applied and further developed by [22], [23], and in the context of the *advanced Step NMPC* (asNMPC) by Zavala and Biegler [24]. Within the asNMPC framework, there have been further recent activities to handle problems active constraint changes, [25]. A review article on some of the recent developments in the field is given in [26]. A link between stability and the properties of the underlying nonlinear programming problem is made in [12].

The approaches for fast sensitivity NMPC outlined above, rely on strong regularity assumptions, such as the linear independence constraint qualification (LICQ), that imply unique multipliers of the NLP. However, in general this assumption may not be satisfied, as was observed by Vicente and Wright [27]. A more general and weaker assumption, that is more likely to be satisfied in an NMPC context, is the Mangasarian Fromowitz constraint qualification (MFCQ), which is consistent with non-unique multipliers of the NLP solution. In the context of direct optimal control methods, among others in direct collocation, the linear independent constraint qualification (LICQ) no longer holds if the number of active constraints exceed the number of degrees of freedom, a case that is not unlikely in economic NMPC applications. Sensitivity-based NMPC for such degenerate systems have been previously only been studied by Jäschke et al. [28].

The contribution of this paper is to present an advanced step NMPC approach that makes use of an improved version of path-following algorithm introduced in [28] and [25], that can handle NMPC problems for which the standard assumption of LICQ does not hold. In particular, we apply the path-following algorithm for parametric NLP with non-unique multipliers from Kungurtsev and Jäschke [29] to obtain fast approximate solutions of the NMPC problem. We present a large-scale case study, and show that the commonly assumed reg-

ularity conditions (LICQ) does not hold. For such formulations of the NMPC optimization problem, standard sensitivity approaches are not applicable, and would fail to work.

This paper is organized as follows. We begin by formulating the economic NMPC problem in Section 2 and present the proposed solution method in Section 3 with the path-following approach. We then demonstrate our proposed method in a case example in Section 4. Finally, we conclude with discussion and some remarks in Section 5.

2. Economic NMPC

2.1. Ideal Economic NMPC

The economic MPC controller computes the optimal control input by solving the following optimization problem

$$
\begin{aligned}
(\mathcal{P}_N): \min_{\mathbf{z}_l, \mathbf{v}_l} \quad & \Psi\left(\mathbf{z}_N\right) + \textstyle\sum_{l=0}^{N-1} \psi\left(\mathbf{z}_l, \mathbf{v}_l\right) && (1)\\
\text{s.t.} \quad & \mathbf{z}_{l+1} = f\left(\mathbf{z}_l, \mathbf{v}_l\right), && l = 0, \ldots, N-1\\
& \mathbf{z}_0 = \mathbf{x}_k, &&\\
& \left(\mathbf{z}_l, \mathbf{v}_l\right) \in \mathcal{Z}, && l = 0, \ldots, N-1\\
& \mathbf{z}_N \in \mathcal{X}_f, &&
\end{aligned}
$$

where $\mathbf{z}_l \in \mathbb{R}^{n_z}$ and $\mathbf{v}_l \in \mathbb{R}^{n_v}$ are the predicted state and control variables at sample time l, respectively. The objective function consists of the terminal cost $\Psi\left(\mathbf{z}_N\right) \in \mathcal{C}^2 : \mathbb{R}^{n_z} \to \mathbb{R}$ and the stage costs $\psi\left(\mathbf{z}_l, \mathbf{v}_l\right) \in \mathcal{C}^2 : \mathbb{R}^{n_z} \times \mathbb{R}^{n_v} \to \mathbb{R}$. The constraints include a discrete time dynamical system $f \in \mathcal{C}^2 : \mathbb{R}^{n_z} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_z}$, the equality constraint for the initial conditions $\mathbf{z}_0$, which are obtained from the measurement of the actual state $\mathbf{x}_k \in \mathbb{R}^{n_z}$ at the time instance k, and the final state variable $\mathbf{z}_N$ is contained within the set defined by the terminal constraint $\mathcal{X}_f$. The set $\mathcal{Z}$ denotes the path constraints on the predicted state and control. Since the set $\mathcal{Z}$ contains both the predicted state and control, for

ease of exposition, we split the set into $\mathcal{X} \subset \mathbb{R}^{n_z}$ and $\mathcal{U} \subset \mathbb{R}^{n_v}$ denoting the feasible state and control sets, respectively.

Having obtained the solution of the optimization problem $\mathcal{P}_N$, denoted by $\mathbf{v}^*_{\{0,\ldots,N-1\}}$, the first sequence of optimized predicted control input $\mathbf{u}_k := \mathbf{v}^*_o$ is applied to a plant that evolves according to

$$\mathbf{x}_{k+1} \quad = f\left(\mathbf{x}_k, \mathbf{u}_k\right) \tag{2}$$

where $\mathbf{x}_k$ is the actual state variable in the plant.

As the time instance k evolves, the optimization problem $\mathcal{P}_N$ is solved repeatedly in a rolling horizon fashion as follows:

1. Obtain measurement data $\mathbf{x}_k$,
2. Solve the optimization problem $\mathcal{P}_N$,
3. Inject the optimized predicted control input $\mathbf{u}_k$,
4. Set $k \leftarrow k + 1$, repeat from Step 1.

We refer to the procedure above as an *ideal NMPC* (*iNMPC*) controller, assuming negligible computation time for solving an online NLP problem.

2.2. The Advanced-step NMPC

The optimization problem (1) is solved at every MPC iteration. For large processes with realistic models, the MPC problems become quite large, and computing their solutions may require a non-negligible amount of time. This computation time causes a delay between obtaining the new measurement values, and implementing the updated input into the plant, and leads to performance loss or even loss of stability [18]. Sensitivity-based methods aim at reducing the computational delay. They are based on the insight that the MPC problems at every iteration are identical, except for the changing initial state $\mathbf{x}_k$. Hence, the initial state variable may be considered a parameter. Instead of solving a full NLP problem for updating the inputs, the asNMPC approach computes the sensitivity of the NLP solution with respect to the initial variable (parameter) $\mathbf{x}_k$. This can be used to obtain a first-order approximation of the solution of

the problem at a nearby parameter. Based on NLP sensitivity, the asNMPC procedure includes the following three steps [24].

1. (*Background step*) Solve the NLP problem $\mathcal{P}_N(\mathbf{z}_{k+1})$ background at time k while setting the initial state value to the predicted state at $k+1$.
2. (*Online step*) When the measurement $\mathbf{x}_{k+1}$ becomes available at time $k+1$, update the optimal solution obtained from the background step using the sensitivity of the optimal solution from step 1.
3. Implement the optimal control input, update $k \leftarrow k+1$, and repeat from Step 1.

If LICQ and strict complementarity hold together with a suitable second order condition, the optimal sensitivity update in Step 2 can be calculated using the implicit function theorem, by solving a system of linear equations that can be formulated using the Karush-Kuhn-Tucker (KKT) system of the NLP [24].

However, if strict complementary does not hold, the system is only directionally differentiable. In this case, a quadratic program must be solved to obtain the directional derivative [30, 28, 31]. Alternatively, it may be heuristically approximated by "clipping", as proposed by [26].

Moreover if the NLP satisfies MFCQ, a weaker constraint qualification that allows for non-unique NLP multipliers, additional challenges arise because setting up the QP requires evaluating the derivatives of the Lagrangian, and selecting the right multipliers values is necessary to obtain correct directional derivatives. If the predicted state $\mathbf{z}_{k+1}$ and the measured state $\mathbf{x}_{k+1}$ are not very close, the online step can be improved by applying a sequence of sensitivities step leading to a pathfollowing algorithm. This is described in the next section.

3. Predictor-Corrector Path-following Economic NMPC

3.1. Preliminaries

We explain the predictor-corrector path-following method in this section along with its application to an advanced-step economic NMPC controller.

First, we define some notation. The ith component of a vector $\mathbf{v}$ is denoted by $[\mathbf{v}]_i$ and if $\mathcal{K}$ is an index set then $[\mathbf{v}]_{\mathcal{K}}$ represents the vector with $|\mathcal{K}|$ components composed of the entries of $\mathbf{v}$ indexed by $\mathcal{K}$.

We consider Problem (1) as a parametric nonlinear optimization,

$$\begin{aligned} \min_{\boldsymbol{\chi}} \quad & F(\boldsymbol{\chi}, \mathbf{p}) \\ \text{subject to} \quad & c_i(\boldsymbol{\chi}, \mathbf{p}) = 0, \quad i \in \mathcal{E}, \\ & c_i(\boldsymbol{\chi}, \mathbf{p}) \leq 0, \quad i \in \mathcal{I}, \end{aligned} \tag{3}$$

where $F : \mathbb{R}^{n_\chi} \times \mathbb{R}^{n_p} \to \mathbb{R}$ is the objective function, $\boldsymbol{\chi} \in \mathbb{R}^{n_\chi}$ the primal variable (i.e. the states $\mathbf{z}$ and inputs $\mathbf{v}$ in Problem (1)) and $\mathbf{p} \in \mathbb{R}^{n_p}$ is the parameter (initial state $\mathbf{x}_k$). The equality and inequality constraint sets are $\mathcal{E} = \{1, \ldots, m\}$ and $\mathcal{I} = \{m+1, \ldots, n\}$, respectively. In sensitivity-based NMPC, the parameter change is from $\mathbf{p}_0$ (the predicted state $\mathbf{z}_k$) to a final value $\mathbf{p}_f$ (the measured (estimated) state $\mathbf{x}_k$).

The Lagrangian is defined as $L(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}) := F(\boldsymbol{\chi}, \mathbf{p}) + \boldsymbol{\lambda}^T c(\boldsymbol{\chi}, \mathbf{p})$, where $\boldsymbol{\lambda}$ is the dual variable (multiplier). In the following, we recall some basic properties regarding the sensitivity of nonlinear programs, see, e.g., [32] and [33]. The Karush-Kuhn-Tucker (KKT) conditions for the problem are

$$\begin{aligned} \nabla_{\boldsymbol{\chi}} L(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}) &= 0, \\ c_i(\boldsymbol{\chi}, \mathbf{p}) &= 0, \; i \in \mathcal{E}, \\ c_i(\boldsymbol{\chi}, \mathbf{p}) &\leq 0, \; i \in \mathcal{I}, \\ \boldsymbol{\lambda}^T c(\boldsymbol{\chi}, \mathbf{p}) &= 0, \\ \boldsymbol{\lambda}_i &\geq 0, \; i \in \mathcal{I}. \end{aligned} \tag{4}$$

We refer to a point $\boldsymbol{\chi}^*$ that satisfies (4) as a KKT point.

Definition 1. *The set of acceptable multipliers for problem* (3)*, given a KKT*

point $\boldsymbol{\chi}^*$*, is defined as*

$$\mathcal{M}\left(\boldsymbol{\chi}^*\right) \triangleq \left\{\boldsymbol{\lambda} \in \mathbb{R}^m \,:\, \nabla_{\boldsymbol{\chi}^*} L\left(\boldsymbol{\chi}^*, \boldsymbol{\lambda}, \mathbf{p}\right) = 0,\, \boldsymbol{\lambda}^T c\left(\boldsymbol{\chi}^*, p\right) = 0,\, and\, \boldsymbol{\lambda}_i \geq 0,\, i \in \mathcal{I}\right\}.$$

We denote active inequality constraints as a set $\mathcal{A}\left(\boldsymbol{\chi}, \mathbf{p}\right) = \{i \in \mathcal{I} \mid c_i\left(\boldsymbol{\chi}, \mathbf{p}\right) = 0\}$. For a KKT point $\boldsymbol{\chi}$ and multiplier $\boldsymbol{\lambda}$ that satisfy the optimality conditions in (4), the active inequality set $\mathcal{A}\left(\boldsymbol{\chi}, \mathbf{p}\right)$ has two disjoint subsets, the weakly active set $\mathcal{A}_0\left(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}\right) = \{i \in \mathcal{A}\left(\boldsymbol{\chi}, \mathbf{p}\right) \mid \boldsymbol{\lambda}_i = 0\}$ and the strongly active set $\mathcal{A}_+\left(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}\right) = \{i \in \mathcal{A}\left(\boldsymbol{\chi}, \mathbf{p}\right) \mid \boldsymbol{\lambda}_i > 0\}$.

The Hessian of the Lagrangian with respect to the primal variables is

$$H\left(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}\right) = \nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} F\left(\boldsymbol{\chi}, \mathbf{p}\right) + \sum_{i=1}^{n} \left(\nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}, \mathbf{p}\right)\right) \boldsymbol{\lambda}_i. \tag{5}$$

Definition 2. *The strong second-order sufficient conditions (SSOSC) holds at* $(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p})$ *if the primal dual pair* $(\boldsymbol{\chi}, \boldsymbol{\lambda})$ *satisfies the KKT first-order necessary conditions* (4) *at* $\mathbf{p}$ *and*

$$\mathbf{d}^T H\left(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}\right) \mathbf{d} > 0 \, for\, all\, \mathbf{d} \in \mathcal{C}\left(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}\right) \setminus \{0\},$$

where the set $\mathcal{C}\left(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}\right)$ *is defined as*

$$\mathcal{C}\left(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}\right) := \left\{\mathbf{d} \,:\, \nabla_{\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}, \mathbf{p}\right)^T \mathbf{d} = 0 \, for\, i \in \mathcal{A}_+\left(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}\right) \cup \mathcal{E}\right\}.$$

Definition 3. *The general strong second-order sufficient optimality conditions (GSSOSC) are satisfied at* $\boldsymbol{\chi}$ *if the SSOSC is satisfied for all* $\boldsymbol{\lambda} \in \mathcal{M}(\boldsymbol{\chi})$ *together with* $\boldsymbol{\chi}$ *that fulfill the first-order necessary conditions* (4)*.*

A constraint qualification is required to ensure that the KKT conditions (4) are necessary for optimality [34]. The standard constraint qualification that is most frequently used is the linear independence constraint qualification (LICQ), which requires that the gradients of the active constraints to be linearly independent:

Definition 4. *Given a point* $(\boldsymbol{\chi}, \mathbf{p})$*, the linear independence constraint qualification (LICQ) holds at* $(\boldsymbol{\chi}, \mathbf{p})$ *if the set of vectors* $\{\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, \mathbf{p}),\ i \in \mathcal{E} \cup \mathcal{A}(\boldsymbol{\chi}, \mathbf{p})\}$ *is linearly independent.*

If LICQ is satisfied at a point $\boldsymbol{\chi}^*$, the set $\mathcal{M}(\boldsymbol{\chi}^*)$ is a singleton, and the multiplier $\boldsymbol{\lambda}$ is unique.

However, in dynamic optimization with path constraints, LICQ may not be satisfied, see e.g., [28] and [27], and the multipliers that satisfy the KKT conditions may become non-unique. A constraint qualification that is more likely to hold in this case is the Mangasarian-Fromovitz constraint qualification. MFCQ requires that the equality constraints are linearly independent, and that there exist a strictly feasible direction into the interior of the feasible set:

Definition 5. *The Mangasarian-Fromovitz constraint qualification (MFCQ) holds at* $(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p})$ *for a feasible point* $\boldsymbol{\chi}$ *if*

1. $\{\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, \mathbf{p}),\ i \in \mathcal{E}\}$ *is linearly independent,*
2. *There exists a direction* $\mathbf{s} \neq 0$*, such that* $\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, \mathbf{p})^T \mathbf{s} = 0$ *for all* $i \in \mathcal{E}$ *and* $\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, \mathbf{p})^T \mathbf{s} < 0$ *for all* $i \in \mathcal{A}(\boldsymbol{\chi}, \mathbf{p})$*.*

The MFCQ implies that the set of multipliers $\mathcal{M}(\boldsymbol{\chi})$ is a bounded polytope [35].

In this work, we refer to an NLP that satisfies the MFCQ (but not the LICQ) as a degenerate NLP. To calculate the sensitivity of a degenerate NLP, we will further make use of the constant rank constraint qualification [36].

Definition 6. *The constant rank constraint qualification (CRCQ) holds at* $(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p})$ *if there exists a neighborhood* $\mathcal{N}$ *of* $\boldsymbol{\chi}$ *such that for all subsets* $\mathfrak{U} \subseteq \mathcal{E} \cup \mathcal{A}(\boldsymbol{\chi}, \mathbf{p})$*, the rank of* $\{\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, \mathbf{p}),\ i \in \mathfrak{U}\}$ *is equal to the rank of* $\{\nabla_{\boldsymbol{\chi}} c_i(\bar{\boldsymbol{\chi}}, \mathbf{p}),\ i \in \mathfrak{U}\}$ *for all* $\bar{\boldsymbol{\chi}} \in \mathcal{N}$*.*

Note that CRCQ is neither weaker nor stronger than MFCQ [36]. Under MFCQ and CRCQ, we present the following result of NLP sensitivity from [32].

Theorem 7. *Let* F *and* c *be twice continuously differentiable with respect to* $\mathbf{p}$ *and* $\boldsymbol{\chi}$ *near* $(\boldsymbol{\chi}^*, \mathbf{p}_0)$*, and let MFCQ and GSSOSC hold at* $(\boldsymbol{\chi}^*, \mathbf{p}_0)$*. Then the*

solution $\boldsymbol{\chi}^*(\mathbf{p})$ *as a function of* $\mathbf{p}$ *is Lipschitz continuous in a neighborhood of* $(\boldsymbol{\chi}^*, \mathbf{p}_0)$*, and the solution function* $\boldsymbol{\chi}^*(\mathbf{p})$ *is directionally differentiable. Moreover, for each* $\mathbf{p}$ *in a neighborhood of* $\mathbf{p}_0$*, and direction* $\mathbf{s} \in \mathbb{R}^{n_p}$ *there exists a set of multipliers* $\boldsymbol{\lambda} \in \mathcal{M}$ *such that the directional derivative uniquely solves the following quadratic program*

$$\begin{aligned} \min_{\Delta_p \boldsymbol{\chi}} \quad & \tfrac{1}{2} \Delta_p \boldsymbol{\chi}^T \nabla_{\boldsymbol{\chi}\boldsymbol{\chi}} L(\boldsymbol{\chi}^*, \boldsymbol{\lambda}, \mathbf{p}_0)\ \Delta_p \boldsymbol{\chi} + \mathbf{s}^T \nabla_{\boldsymbol{\chi}\mathbf{p}} L(\boldsymbol{\chi}^*, \boldsymbol{\lambda}, \mathbf{p}_0)\, \Delta_p \boldsymbol{\chi} \\ \text{subject to} \quad & \nabla_{\mathbf{p}} c_i(\boldsymbol{\chi}^*, \mathbf{p}_0)\, \mathbf{s} + \nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}^*, \mathbf{p}_0)\, \Delta_p \boldsymbol{\chi} = 0,\ i \in \mathcal{E} \\ & \nabla_{\mathbf{p}} c_i(\boldsymbol{\chi}^*, \mathbf{p}_0)\, \mathbf{s} + \nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}^*, \mathbf{p}_0)\, \Delta_p \boldsymbol{\chi} = 0,\ i \in \mathcal{A}_+ \\ & \nabla_{\mathbf{p}} c_i(\boldsymbol{\chi}^*, \mathbf{p}_0)\, \mathbf{s} + \nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}^*, \mathbf{p}_0)\, \Delta_p \boldsymbol{\chi} \le 0,\ i \in \mathcal{A}_0 \end{aligned} \tag{6}$$

If in addition CRCQ holds, then the multiplier value $\boldsymbol{\lambda}$ *at which the quadratic program* (6) *must be evaluated, can be found as a solution of the linear program:*

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & \boldsymbol{\lambda}^T \nabla_{\mathbf{p}} c(\boldsymbol{\chi}^*, \mathbf{p}_0)\, \mathbf{s} \\ \text{subject to} \quad & \boldsymbol{\lambda} \in \mathcal{M}(\boldsymbol{\chi}^*, \mathbf{p}_0). \end{aligned} \tag{7}$$

Proof. *See Ralph and Dempe [32].*

We define the optimality residual as

$$\eta(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}) = \left\| \begin{pmatrix} \nabla_{\boldsymbol{\chi}} F(\boldsymbol{\chi}, \mathbf{p}) + \nabla_{\boldsymbol{\chi}} c(\boldsymbol{\chi}, \mathbf{p})\, \boldsymbol{\lambda} \\ c(\boldsymbol{\chi}, \mathbf{p})_{\mathcal{E}} \\ [\min(c(\boldsymbol{\chi}, \mathbf{p}), \boldsymbol{\lambda})]_{\mathcal{I}} \end{pmatrix} \right\|_{\infty}. \tag{8}$$

In a neighborhood of the solution, this optimality residual can be used to estimate the strongly active inequality constraints as

$$\mathcal{A}_+(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}) = \{ i \in \mathcal{I} : \boldsymbol{\lambda}_i > \eta(\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}) \}, \tag{9}$$

and an estimate of the strongly active set A_+ is obtained by including the indices of equality constraints, such that $A_+ = \mathcal{A}_+ \cup \mathcal{E}$.

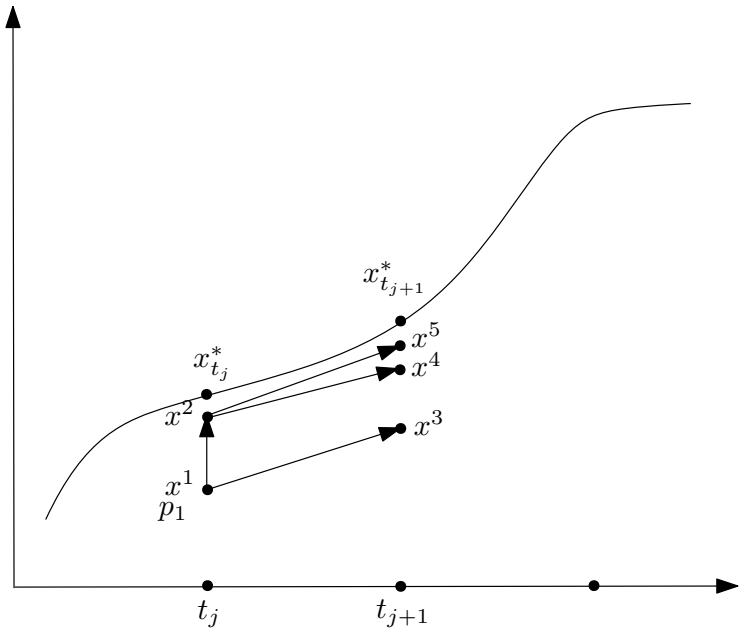

Figure 1: Illustration of the corrector and predictor steps in the path-following method.

3.2. Predictor-Corrector Path-Following

The path-following method for tracing the optimal solution along a parameter change is described in detail in [29] and consists of three main steps: a corrector step, a predictor step, and a multiplier jump step. These three steps are run repeatedly to follow the path of optimal solutions, starting from an initial parameter value $\mathbf{p}_0$ until the final parameter $\mathbf{p}_f$.

We introduce a new parameter $t \in [0, 1]$, to describe the progress along the path in $\mathbf{p}$, such that $\mathbf{p}(t) = (1-t)\,\mathbf{p}_0 + t\mathbf{p}_f$. The path-following algorithm approximates the solution of the NLP, starting at $t_0 = 0$, where $\mathbf{p} = \mathbf{p}_0$, at timesteps t_j, such that $t_0 = 0 < t_1 < \ldots < t_j < \ldots <= 1$. We denote the primal and dual variables during at the path-following iterations as $\boldsymbol{\chi}_j$ and $\boldsymbol{\lambda}_j$ respectively, where j represents the index of the iteration along the path.

The corrector and predictor steps are illustrated in Figure 1, where the goal is to track optimal solutions at times t_j and t_{j+1} denoted with $x^*_{t_j}$ and $x^*_{t_{j+1}}$, respectively. Consider a step in the path-following method starting at point

x^1 that is an approximation of the solution of the NLP at $\mathbf{p}_1 = \mathbf{p}_{t_1}$. If one applies a corrector step only it moves x^1 to the point x^2. A pure predictor step starting from x^1 would lead to the point x^3. If one applies a standard predictor-corrector method [31], it steps from x^1 to the point x^4. The improved path-following method employed in this work starts from the point x_1 and, using the improved problem data (derivatives) from x^2, results in a step to x^5. The steps of the path-following algorithm are described below in detail.

3.2.1. Corrector Step

This step takes an approximate solution of the primal variables and the strongly active dual variables and refines them for a given value of $\mathbf{p}$. This is done by solving a system of a linear equations

$$\begin{pmatrix} H\left(\boldsymbol{\chi}_j, \boldsymbol{\lambda}_j, t\right) & \nabla_{\boldsymbol{\chi}} c_{A_+,j}\left(\boldsymbol{\chi}_j, t\right) \\ \nabla_{\boldsymbol{\chi}} c_{A_+,j}\left(\boldsymbol{\chi}_j, t\right)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta_c \boldsymbol{\chi} \\ \Delta_+ \boldsymbol{\lambda} \end{pmatrix} = - \begin{pmatrix} \nabla_{\boldsymbol{\chi}} F\left(\boldsymbol{\chi}_j, t\right) + \nabla_{\boldsymbol{\chi}} c\left(\boldsymbol{\chi}_j, t\right) \boldsymbol{\lambda}_j \\ \nabla_{\boldsymbol{\chi}} c_{A_+,j}\left(\boldsymbol{\chi}_j, t\right) \end{pmatrix}. \tag{10}$$

Since the LICQ does not hold, the Jacobian $\nabla_{\boldsymbol{\chi}} c_{A_+,j}\left(\boldsymbol{\chi}_j, t\right)$ is not full rank, unless the dual variables $\boldsymbol{\lambda}_j$ used to define A_+ are chosen from vertex of the polytope of optimal multipliers [32]. This can be done by obtaining $\boldsymbol{\lambda}_j$ in the multiplier jump step (see Section 3.2.3) by solving the linear program using a simplex method.

The improved approximate dual variables for the strongly active constraint are obtained from the solution (10), i.e., $[\Delta_c \boldsymbol{\lambda}]_{A_+,j} = \Delta_+ \boldsymbol{\lambda}$ and the remaining multipliers are set to zero, $[\Delta_c \boldsymbol{\lambda}]_{\{1,\ldots,n\} \setminus A_+,j} = 0$. In Figure 1, this corrector step moves point x^1 to the point x^2.

3.2.2. Predictor Step

Based on (6), we compute a predictor by solving the following QP

$$\begin{aligned} \min_{\Delta_p \boldsymbol{\chi}} \quad & \left(\nabla_{\boldsymbol{\chi}} F\left(\boldsymbol{\chi}_j, t + \Delta t\right) - \nabla_{\boldsymbol{\chi}} F\left(\boldsymbol{\chi}_j, t\right)\right)^T \Delta_p \boldsymbol{\chi} + \tfrac{1}{2} \Delta_p \boldsymbol{\chi}^T H\left(\boldsymbol{\chi}_j, \boldsymbol{\lambda}_j, t + \Delta t\right) \Delta_p \boldsymbol{\chi} & (11) \\ \text{subject to} \quad & \nabla_t c_i\left(\boldsymbol{\chi}_j, t\right) \Delta t + \left(\nabla_{\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j, t + \Delta t\right) + \nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j, t + \Delta t\right) \Delta_c \boldsymbol{\chi}\right)^T \Delta_p \boldsymbol{\chi} & = 0,\ i \in A_{+,j} \\ & \nabla_t c_i\left(\boldsymbol{\chi}_j, t\right) \Delta t + \left(\nabla_{\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j, t + \Delta t\right) + \nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j, t + \Delta t\right) \Delta_c \boldsymbol{\chi}\right)^T \Delta_p \boldsymbol{\chi} & \leq 0,\ i \in A_j \setminus A_{+,j}. \end{aligned}$$

Note that we add a second order gradient correction of the constraints, to obtain improved accuracy. Here, we need to evaluate the derivatives of the objective function, the Lagrangian, and the constraints at $t + \Delta t$. Moreover since the parameter, in this case is the initial state variable, enters linearly to the problem, the objective function derivative remains the same at $t+\Delta t$. Thus, the QP (11) simplifies to

$$\begin{aligned}
\min_{\Delta_p \boldsymbol{\chi}} \quad & \tfrac{1}{2} \Delta_p \boldsymbol{\chi}^T H\left(\boldsymbol{\chi}_j, \boldsymbol{\lambda}_j, t + \Delta t\right) \Delta_p \boldsymbol{\chi} && (12)\\
\text{subject to} \quad & \nabla_t c_i\left(\boldsymbol{\chi}_j, t\right) \Delta t + \left(\nabla_{\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j, t + \Delta t\right) + \nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j, t + \Delta t\right) \Delta_c \boldsymbol{\chi}\right)^T \Delta_p \boldsymbol{\chi} && = 0,\ i \in A_{+,j}\\
& \nabla_t c_i\left(\boldsymbol{\chi}_j, t\right) \Delta t + \left(\nabla_{\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j, t + \Delta t\right) + \nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j, t + \Delta t\right) \Delta_c \boldsymbol{\chi}\right)^T \Delta_p \boldsymbol{\chi} && \leq 0,\ i \in A_j \backslash A_{+,j}
\end{aligned}$$

We obtain the primal and dual solution in this step $(\Delta_p \boldsymbol{\chi}, \Delta_p \boldsymbol{\lambda})$. Combining with the solution from the corrector step, we get $(\Delta \boldsymbol{\chi}, \Delta \boldsymbol{\lambda}) = (\Delta_c \boldsymbol{\chi} + \Delta_p \boldsymbol{\chi}, \Delta_c \boldsymbol{\lambda} + \Delta_p \boldsymbol{\lambda})$. We update the primal and dual variables solutions, i.e., $\boldsymbol{\chi}_{j+1} = \boldsymbol{\chi}_j + \Delta \boldsymbol{\chi}$, $\boldsymbol{\lambda}_{j+1} = \boldsymbol{\lambda}_j + \Delta \boldsymbol{\lambda}$, and consequently the active set $\mathcal{A}_{j+1}$. The predictor QP will lead the point x^2 to x^5 described in Figure 1.

3.2.3. Multiplier Jump Step

To allow discontinuity in the multipliers along the path, we compute the dual variable solutions by solving the following LP,

$$\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & \boldsymbol{\lambda}^T \nabla_t c\left(\boldsymbol{\chi}_j + \Delta \boldsymbol{\chi}, t + \Delta t\right) \Delta t && (13)\\
\text{subject to} \quad & -|\Omega| \leq \vartheta\left(\boldsymbol{\lambda}\right) \leq |\Omega|\\
& \boldsymbol{\lambda}_{\mathcal{I}} \geq 0\\
& \boldsymbol{\lambda}_{i \notin A_{j+1}} = 0.
\end{aligned}$$

where

$$\Omega = \nabla_{\boldsymbol{\chi}} L(\boldsymbol{\chi}_j + \Delta \boldsymbol{\chi}, \boldsymbol{\lambda}_j + \Delta \boldsymbol{\lambda}, t + \Delta t),$$

$$\vartheta\left(\boldsymbol{\lambda}\right) = \nabla_{\boldsymbol{\chi}} F(\boldsymbol{\chi}_j + \Delta \boldsymbol{\chi}, t + \Delta t) + \sum_{i \in A_{j+1}} \nabla_{\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_j + \Delta \boldsymbol{\chi}, t + \Delta t\right) \boldsymbol{\lambda}_i$$

The solution ($\boldsymbol{\lambda}_{LP}$) updates the dual variable solutions $\boldsymbol{\lambda}_{j+1} = \boldsymbol{\lambda}_{LP}$ and also the strongly active set $A_{+,j+1} = \left\{i : [\boldsymbol{\lambda}_{j+1}]_i > 0\right\} \cup \mathcal{E}$. The three steps are summarized in Algorithm 1.

3.2.4. Adaptation of stepsize Δt

The stepsize Δt may vary from one step to another, depending on the optimality residual η_j defined in (8). If $\eta_{j+1} < \eta_j$, then Δt is increased to $\Delta t = min\left(1 - t_j, \frac{t_j}{\alpha}\right)$. In our experiment, the parameter α is set to 0.6. In case $\eta_{j+1} > \eta_{max}$, the stepsize Δt is decreased by multiplication with $\alpha \in (0, 1)$.

Remark 8. *In [29], an NLP solver is invoked as contingency in case a predictor-corrector step fails. In the context of asNMPC, it is not required because an NLP solver in already called in the background step of asNMPC.*

Remark 9. *The parameters in the algorithm include the maximum optimality residual η_{max} and γ for determining the size of Δt.*

Remark 10. *The algorithm can be implemented in a distributed fashion where the initial Δt is initiated with different values (multi-start Δt).*

Further details on the algorithm, including a proof the convergence, can be found in [29].

3.3. Path-following NMPC (pf-NMPC)

We use the predictor-corrector path-following method in the online step of the asNMPC controller. The resulting pf-NMPC controller in Algorithm 2 and the predictor-corrector path-following method is invoked in the function `MFCQ_PC_PF`. Note that the final parameter value $\mathbf{p}_f$ in the algorithm is obtained from measurement data.

Algorithm 1 Predictor-corrector path-following method

Input: t, $\boldsymbol{\chi}$, $\boldsymbol{\lambda}$ close to solution $(\boldsymbol{\chi}^*(t), \boldsymbol{\lambda}^*(t))$ such that $\{\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, t)\}_{\{i \in \mathcal{I}:\, \boldsymbol{\lambda}_i > 0\} \cup \mathcal{E}}$ is linearly independent, Δt, $\eta_{max} < 1$.
Output: $\boldsymbol{\chi}$ and $\boldsymbol{\lambda}$ at $\mathbf{p}_f$

1: **function** MFCQ_PC_PF($\boldsymbol{\chi}, \boldsymbol{\lambda}, \mathbf{p}_0, \mathbf{p}_f, \Delta t$)
2: Define parameter γ satisfying $0 < \gamma < 1$.
3: Define A_+.
4: Set $j \leftarrow 0$.
5: Set $t_j = 0$.
6: **while** $t_j < 1$ **do**
7: Solve (**CorrectStep**) for $(\Delta_c \boldsymbol{\chi}, \Delta_+ \boldsymbol{\lambda})$.
8: Solve (**QPPredict**) for $(\Delta_p \boldsymbol{\chi}, \Delta_p \boldsymbol{\lambda})$.
9: Set $(\Delta\boldsymbol{\chi}, \Delta\boldsymbol{\lambda}) = (\Delta_p \boldsymbol{\chi}, \Delta_p \boldsymbol{\lambda}) + (\Delta_c \boldsymbol{\chi}, \Delta_c \boldsymbol{\lambda})$.
10: Compute $\eta_{j+\Delta} := \eta\left(\boldsymbol{\chi}_j + \Delta\boldsymbol{\chi},\, \boldsymbol{\lambda}_j + \Delta\boldsymbol{\lambda},\, t_j + \Delta t\right)$.
11: **if** $\eta_{j+\Delta} < \eta_{max}$ **then**
12: $\boldsymbol{\chi}_{j+1} \leftarrow \boldsymbol{\chi}_j + \Delta\boldsymbol{\chi}$
13: $\boldsymbol{\lambda}_{j+1} \leftarrow \boldsymbol{\lambda}_j + \Delta\boldsymbol{\lambda}$
14: $t_{j+1} \leftarrow t_j + \Delta t$
15: $\mathbf{p}(t_j) = (1 - t_j)\,\mathbf{p}_0 + t_j \mathbf{p}_f$
16: **if** $\eta_{j+\Delta} < \eta_j^{1+\gamma}$ **then** ▷ very good step
17: Increase Δt.
18: **end if**
19: Update $\mathcal{A}_+$.
20: Solve (**JumpLP**) to redefine $\boldsymbol{\lambda}_{j+1}$.
21: Let $A_+ = \left\{i : [\boldsymbol{\lambda}_{j+1}]_i > 0\right\} \cup \mathcal{E}$.
22: **else**
23: Decrease Δt
24: Go to line 12
25: **end if**
26: $j \leftarrow j + 1$.
27: **end while**
28: Return $\boldsymbol{\chi}$
29: **end function**

Algorithm 2 Economic pf-NMPC algorithm

Input: initial state $\mathbf{x}_0$ and stepsize Δt.
Output: The actual state $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, ...

for $k = 0, 1, 2, \ldots$ **do**
  $[\boldsymbol{\chi}^*, \boldsymbol{\lambda}^*] \leftarrow$ solution of the NLP $\mathcal{P}_N(\mathbf{z}_{k+1})$ for $k+1$.
  if a measurement of $\mathbf{x}_{k+1}$ is available **then**
    Set $\mathbf{p}_0 = \mathbf{x}_0$
    Set $\mathbf{p}_f = \mathbf{x}_{k+1}$
    $\Delta\boldsymbol{\chi}^* \leftarrow$ MFCQ_PC_PF$(\boldsymbol{\chi}^*, \boldsymbol{\lambda}^*, \mathbf{p}_0, \mathbf{p}_f, \Delta t)$
    Set solution $\boldsymbol{\chi}^* = \boldsymbol{\chi}^*(\mathbf{p}_0) + \Delta\boldsymbol{\chi}^*$
    Inject the first input move of $\boldsymbol{\chi}^*$ into the plant
  end if
  Update initial state $\mathbf{x}_0 \leftarrow \mathbf{x}_{k+1}$
  Set $k+1 \leftarrow k$
end for

4. A Numerical Case Example

In this section we demonstrate the pf-NMPC controller performance and compare its results against the iNMPC controller. All simulations are done in `MATLAB` using `CasADi` [37] version 3.4.5, which interfaces `IPOPT` [38] as the NLP solver used in this work. We use the MINOS QP [39] solver from TOMLAB and CPLEX [40] as the LP solver.

4.1. Process Description

We consider a reactor and separator processes illustrated in Figure 2 as the plant, which was also used in [31]. The original distillation column model is from [41]. The continuous-stirred tank reactor (CSTR) is fed with a stream F_0 consisting 100% of component A. A first order reaction $A \rightarrow B$ takes place in the reactor where B is the desired product and the product flow rate F is fed to the column. In the distillation column, the raw material is separated from the product and recycled into the reactor. The desired product B leaves the distillation column as the bottom product, which is required to have a certain purity.

The process model has 84 dynamic state variables of which 82 are from the distillation (concentration and holdup for each stage) and two from the CSTR

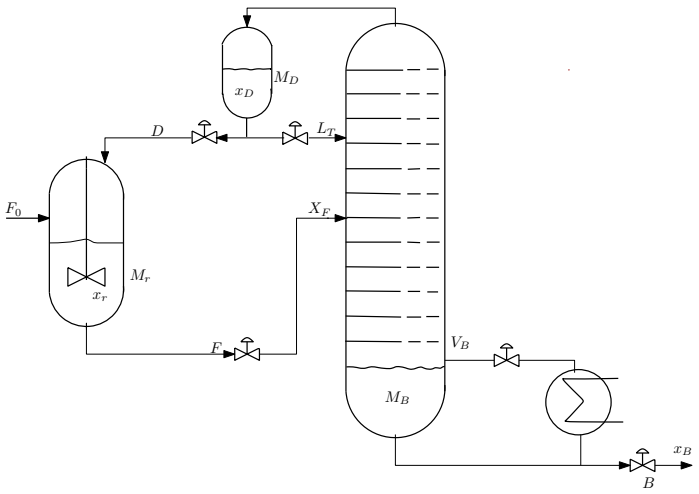

Figure 2: Diagram of CSTR and Distillation Column with control inputs reflux flow (L_T), boilup flow (V_B), feeding rate to the distillation (F), distillate (top) and bottom product flow rates (D and B)

(one concentration and one holdup). We discretize the system by applying direct collocation on finite elements using Lagrange collocation with three collocation points per finite element. The inputs are discretized such that the input is kept constant within a finite element. In contrast to previous work [31], we now impose bound constraints for all collocation points. If the bound constraints become active, this potentially results in more active constraints than degrees of freedom, and thus the LICQ is violated. In this case the standard methods that require LICQ for doing sensitivity-based MPC will fail.

However, as there generally exists a strictly feasible point in the interior of the bounds and the discretized ordinary differential equation equality constraints are expected to have linearly independent gradients, MFCQ can be expected to hold, and the approach outlined in this work can still be applied.

The stage cost of the economic objective function to optimize under operation is to minimize operating cost, i.e.,

$$J = p_F F_0 + p_V V_B - p_B B, \tag{14}$$

where p_F is the feed cost, p_V is the cost of the steam used in the reboiler of the

distillation column, and p_B is the price obtained for selling the product. The price setting is $p_F = 1\,\$/kmol$, $p_V = 0.02\,\$/kmol$, $p_B = 2\,\$/kmol$.

To ensure stability, we use the regularized objective function with a state steady objective function value as follows

$$J_m = p_F F_0 + p_V V_B - p_B B + \|\mathbf{x} - \mathbf{x}_s\|^2_{\mathcal{Q}_1} . \tag{15}$$

The operational constraints enforced at all collocation points are the concentration of the bottom product ($x_B \leq 0.1$) and the holdup in the CSTR ($0.3 \leq M_{CSTR} \leq 0.75$) kmol. The control inputs are reflux flow (L_T), boilup flow (V_B), feeding rate to the distillation (F), distillate (top) and bottom product flow rates (D and B). These control inputs have bound constraints as follows

$$\begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \leq \begin{bmatrix} L_T \\ V_B \\ F \\ D \\ B \end{bmatrix} \leq \begin{bmatrix} 10 \\ 4.008 \\ 10 \\ 1.0 \\ 1.0 \end{bmatrix} [kmol/min] .$$

We consider the case of optimizing a production rate change starting operation with an initial feed rate $F_0 = 0.30$ (kmol/min), which is increased to $F_0 = 0.31$ (kmol/min).

We run the economic NMPC controller every one minute with a prediction horizon of 45 minutes yielding 15429 variables and 15204 nonlinear equality constraints as well as bound constraints for each variable. In the case that there are more active constraints than the number of variables in the system, LICQ is violated. That is, when more than 225 (i.e. 15429 (variables) - 15204 (equality constraints)) bound constraints are active, the LICQ fails and standard methods for calculating the sensitivity update cannot be used.

In our path-following algorithm, we set the initial Δt to 1.0 and $\eta_{max} = 0.01$. That means we first try a large step, and then reduce the stepsize if necessary to maintain accuracy. The 'real' plant model is simulated by the 'ode15s' solver

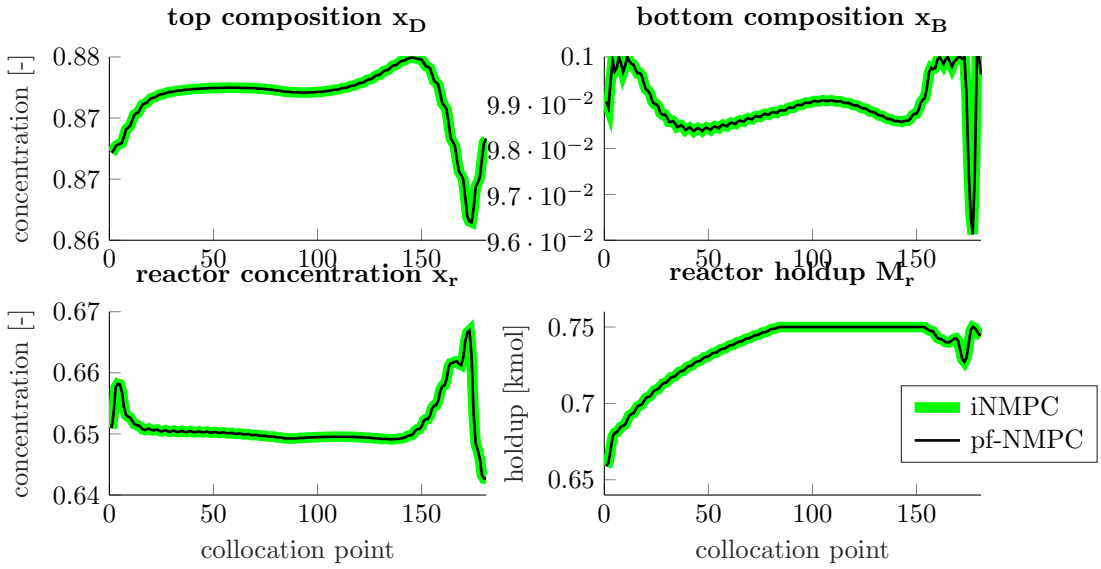

Figure 3: Comparison of selected open loop state trajectories of the iNMPC and pf-NMPC from the second iteration. The top figures represent the top and bottom compositions at the distillation, respectively. The bottom figures show the concentration and holdup at the CSTR, respectively.

in `MATLAB`. We add noise to the holdup measurement data in the distillation, where the noise is taken to have a normal distribution with zero mean and a variance of one percent of the steady state values.

4.2. Comparison of Open-loop Optimization Results

We compare open loop optimization solutions (predicted states) at the second iterations of the iNMPC and pf-NMPC approaches with the presence of measurement noise. That is, we compare how the path-following solution matches the solution of the full NLP. The difference between the path-following solution and NLP solution is shown in Figure 4 and selected state trajectories are given in Figure 3. We observe that the pf-NMPC closely tracks the solutions from the iNMPC controller.

We show the Lagrange multipliers for the active bound constraints of the bottom concentration x_B and reactor holdup in Figure 5. For the reactor holdup (Figure 5 top), it is seen that the Lagrange multipliers are zero over large periods

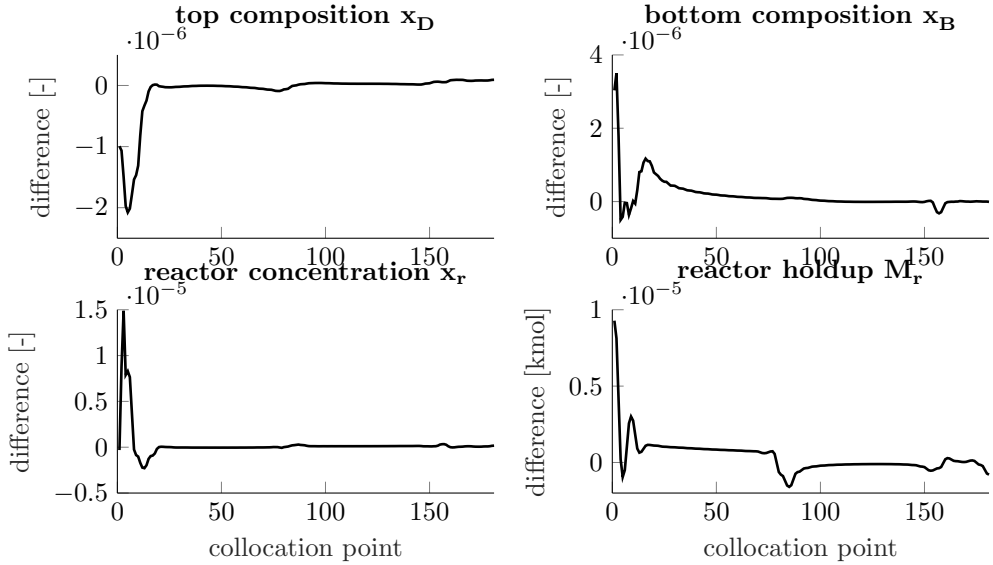

Figure 4: The difference in selected open loop state-trajectories of the iNMPC and pf-NMPC from the second iteration. The top figures represent the top and bottom compositions at the distillation, respectively. The bottom figures show the concentration and holdup at the CSTR, respectively.

of the prediction horizons where the constraint is inactive. Between collocation point 85 and 153, and the very end of the prediction horizon, the path constraint becomes active, this is also reflected in the positive multiplier value obtained from the interior point optimizer (IPOPT) used (Figure 5 green line). As LICQ does not hold here, and the multiplier is non-unique, IPOPT returns a multiplier value in the analytic center of the multiplier set, that is not suitable for calculating sensitivity updates. In our path-following algorithm, the multiplier jump step selects one vertex of the feasible multiplier set, resulting in the multiplier being zero for these collocation points, at which the constraint is not strongly active. This is nicely shown in Figure 5 top.

For the bottom concentration, shown in the bottom of Figure 5, the Lagrange multipliers are zero for most of the time, because the constraint is inactive most of the time, except at the end of the horizon (see Figure 3 top right). Since we have added noise to the state measurements, the predicted state for the NMPC in the background step will be different from the measured state obtained from

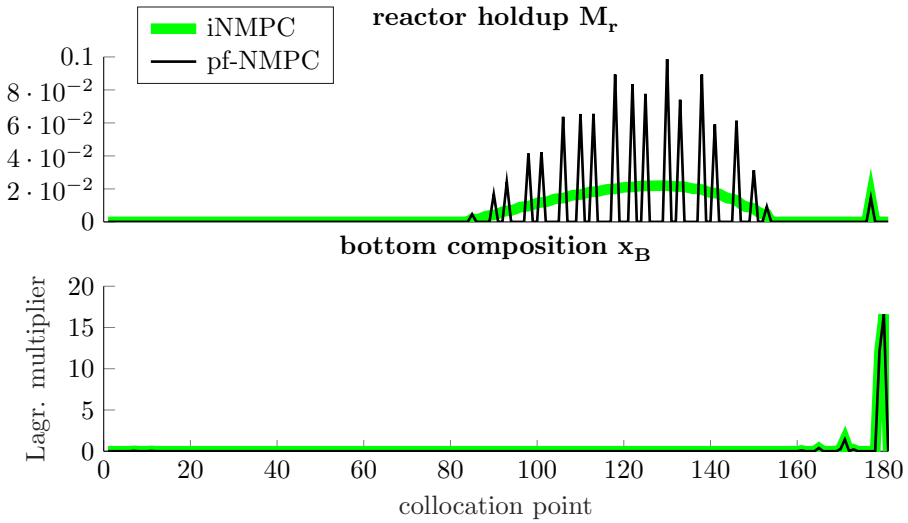

Figure 5: The Lagrange multipliers for the active upper bound constraints at bottom concentration and reactor holdup.

the plant. This difference may cause active set changes. We also plot active-set changes for the bottom concentration and reactor holdup from 26th MPC iteration in Figure 6 and 7, respectively, as the times at which the background (predicted solution without noise) and the online solution (with a measured noisy state) are different. We see that we have active set changes between the background and the online solutions.

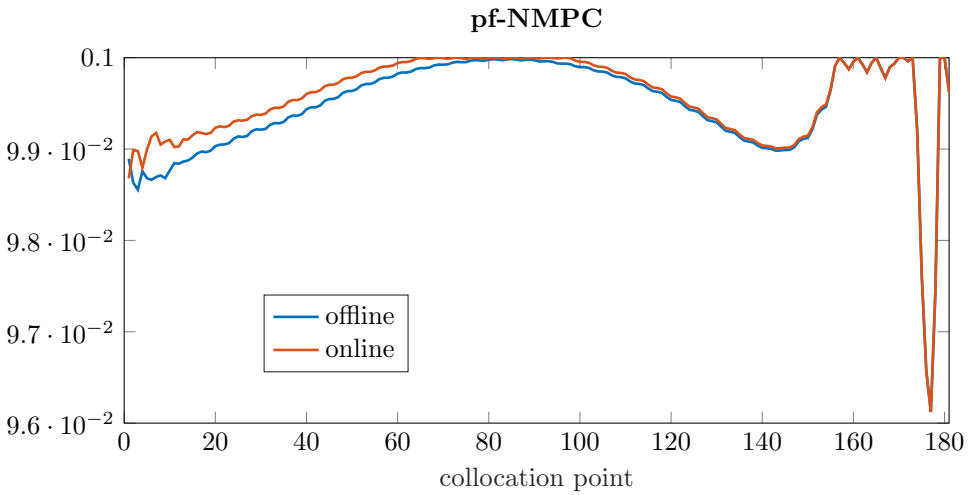

Figure 6: Bottom concentration during the background (offline) and online steps of pf-NMPC at iteration 26.

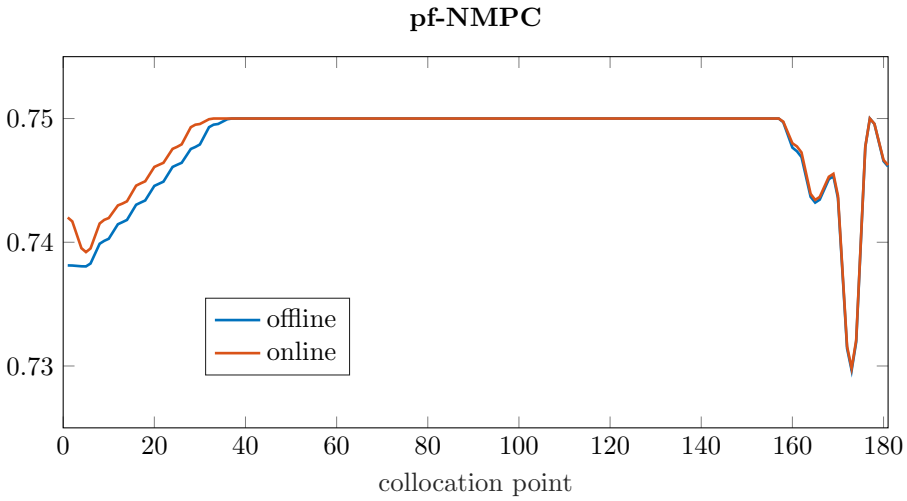

Figure 7: Open loop solution of reactor holdup from pf-NMPC at iteration 26.

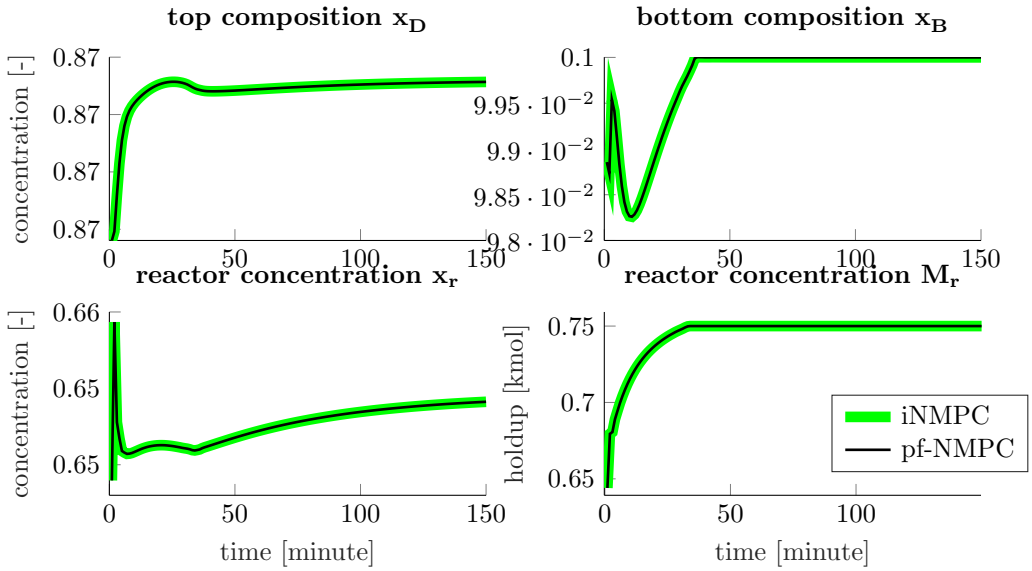

Figure 8: Comparison of close loop state variables between iNMPC and pf-NMPC controllers. The top figures represent the top and bottom compositions at the distillation, respectively. The bottom figures show the concentration and holdup at the CSTR, respectively.

We ran simulations for both NMPC controllers without measurement noise. We plot the four selected state variables in Figure 8, optimized control inputs Figure 9, and number of active bound constraints in Figures 10. As the closed-loop system evolves toward a steady-state, the number of active bound constraints increases to around 280. This makes sense, as the reactor holdup and the bottom concentration constraints are active at steady-state. These constraints are imposed at every collocation point and for this formulation the standard sensitivity approach would fail. The solutions confirm that the pf-NMPC controller gives very good approximation to the results of an iNMPC controller.

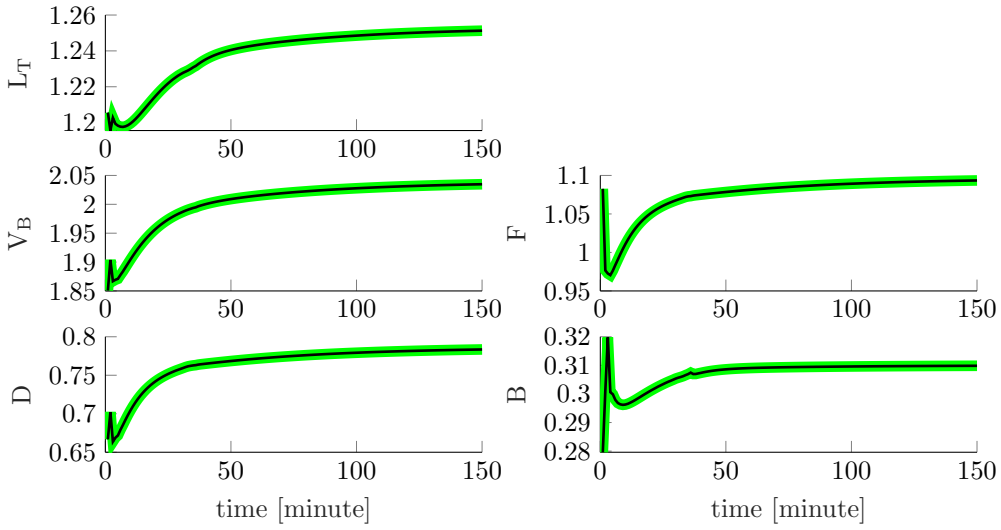

Figure 9: Comparison of optimized control inputs between iNMPC and pf-NMPC controllers.

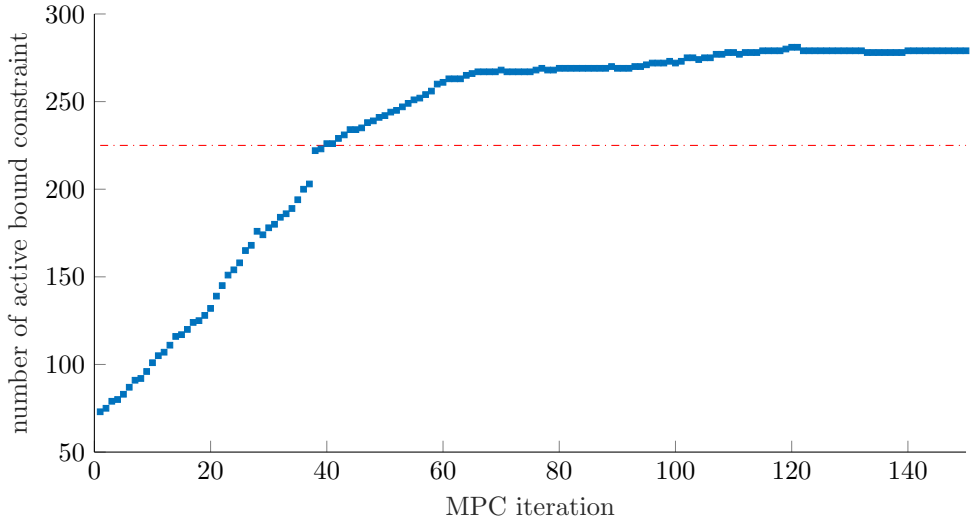

Figure 10: Number of active bound constraints in the course of MPC iterations for more than 225 active bound constraints, LICQ fails and standard sensitivity approach cannot be applied.

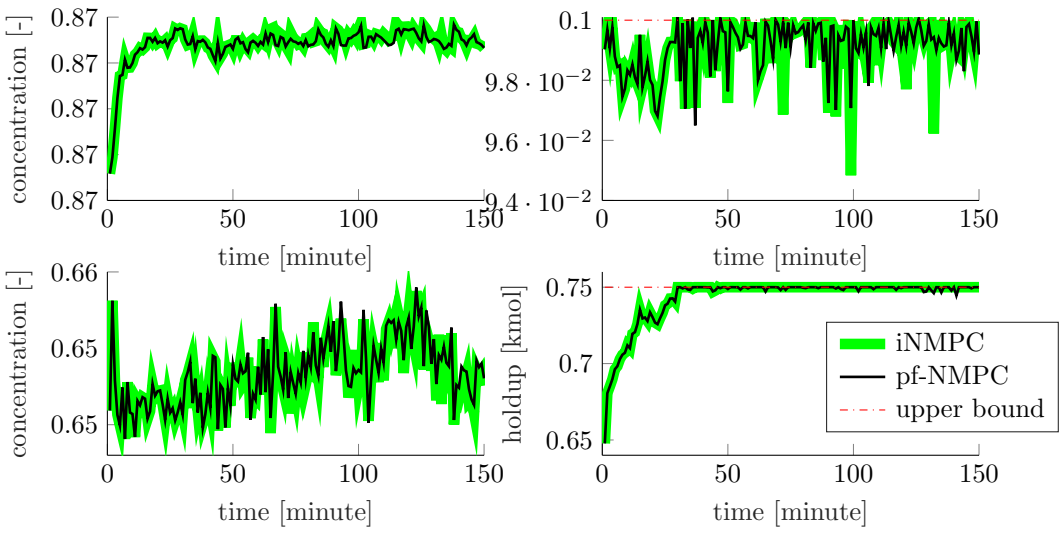

Figure 11: Comparison of close loop state variables between iNMPC and pf-NMPC controllers. The top figures represent the top and bottom compositions at the distillation, respectively. The bottom figures show the concentration and holdup at the CSTR, respectively.

4.4. Closed-loop Results – With Measurement Noise

Next, we run simulations with measurement noise on all the holdups in the system. The noise is taken to have a normal distribution with zero mean and a variance of one percent of the optimal steady state values. The closed-loop simulation results for a selection of close-loop state variables are shown in Figure 11, and the corresponding input variables are given in Figure 12. The active bound constraints are the upper bound constraint of the composition at the bottom of the distillation, which must be below 0.1, and the upper bound of the holdup at the CSTR constraining below 0.75. Due to the measurement noise, the number of active bound constraints can vary from one MPC iteration to another. This variation is depicted in Figure 13.

As seen in Figure 13, and similarly observed in Figure 10, the number of active bound constraints is increasing from the beginning until around the 31st minute. The reason is because the closed-loop system approaches a new steady-point with the feed rate $F_0 = 0.31$ (kmol/min).

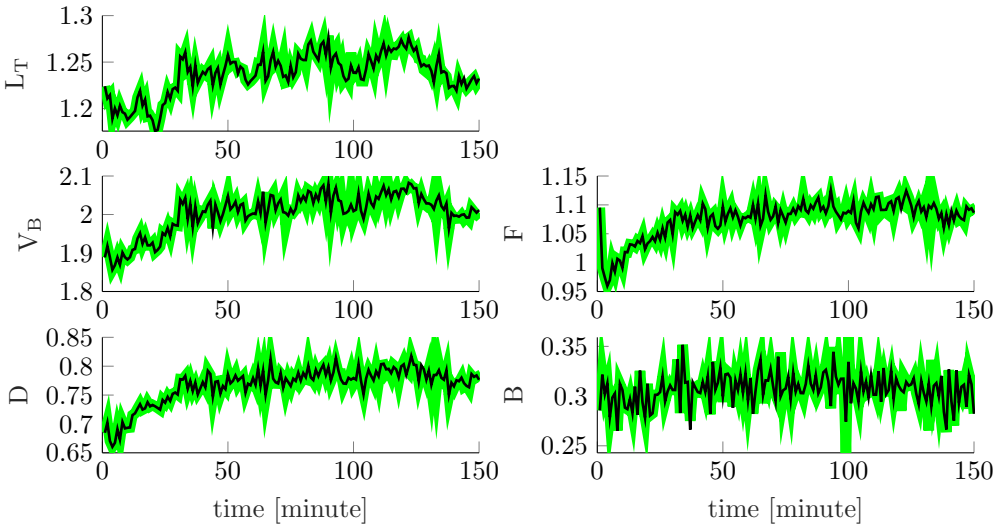

Figure 12: Comparison of optimized control inputs between iNMPC and pf-NMPC controllers.

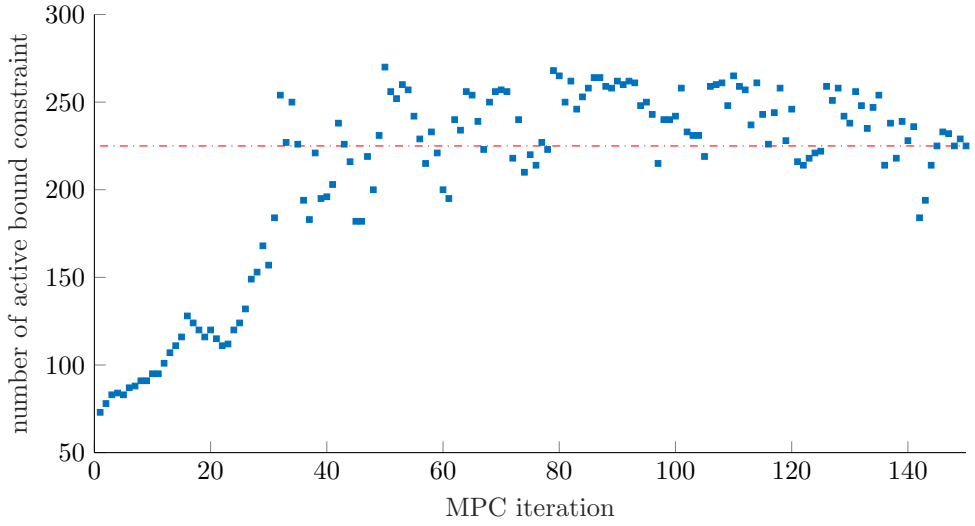

Figure 13: Number of active bound constraints in the course of MPC iterations at more than 225 active bound constraints LICQ fails.

5. Conclusion

We have proposed the use of a predictor-corrector path-following method, consisting of the three steps (corrector, predictor, and multiplier jump step), for solving the online open-loop optimal control problem in an economic NMPC. We have shown that the pf-NMPC works as expected in the case example, and accurately tracks the solutions of an iNMPC controller in the presence of non-unique multipliers. Future work may include to develop a distributed version for the path-following algorithm such as to initialize Δt with a set of different values (multi-start). Furthermore, an efficient implementation of computing the Hessian of the constraints in the predictor step will be investigated. We intend to have the Hessian readily available from computing the second derivative of the Lagrangian.

References

[1] J. B. Rawlings, D. Angeli, C. N. Bates, Fundamentals of economic model predictive control, in: 51st IEEE Conference on Decision and Control (CDC), IEEE, 2012, pp. 3851–3861.

[2] M. Ellis, J. Liu, P. D. Christofides, Economic Model Predictive Control: Theory, Formulations and Chemical Process Applications, Springer, 2016.

[3] T. Faulwasser, L. Grüne, M. A. Müller, et al., Economic nonlinear model predictive control, Foundations and Trends® in Systems and Control 5 (1) (2018) 1–98.

[4] M. Diehl, R. Amrit, J. B. Rawlings, A lyapunov function for economic optimizing model predictive control, IEEE Transactions on Automatic Control 56 (3) (2011) 703–707.

[5] D. Angeli, R. Amrit, J. B. Rawlings, On average performance and stability of economic model predictive control, IEEE transactions on automatic control 57 (7) (2012) 1615–1626.

[6] A. Alessandretti, A. P. Aguiar, C. N. Jones, An input-to-state-stability approach to economic optimization in model predictive control, IEEE Transactions on Automatic Control.

[7] M. Heidarinejad, J. Liu, P. D. Christofides, Economic model predictive control of nonlinear process systems using lyapunov techniques, AIChE Journal 58 (3) (2012) 855–870.

[8] M. Zanon, S. Gros, M. Diehl, Indefinite linear mpc and approximated economic mpc for nonlinear systems, Journal of Process Control 24 (8) (2014) 1273–1281.

[9] M. Zanon, S. Gros, M. Diehl, A tracking mpc formulation that is locally equivalent to economic mpc, Journal of Process Control 45 (2016) 30–42.

[10] O. I. Olanrewaju, J. M. Maciejowski, Implications of dissipativity on stability of economic model predictive control—the indefinite linear quadratic case, Systems & Control Letters 100 (2017) 43–50.

[11] V. M. Zavala, A multiobjective optimization perspective on the stability of economic mpc, in: In the Proceeding of 9th International Symposium of Advanced Control of Chemical Procesess, IFAC, June 7-10, 2015, Whistler, British Columbia, Canada, 2015.

[12] D. W. Griffith, V. M. Zavala, L. T. Biegler, Robustly stable economic nmpc for non-dissipative stage costs, Journal of Process Control 57 (2017) 116–126.

[13] L. Grüne, Economic receding horizon control without terminal constraints, Automatica 49 (3) (2013) 725–734.

[14] X. Chen, M. Heidarinejad, J. Liu, P. D. Christofides, Distributed economic mpc: Application to a nonlinear chemical process network, Journal of Process Control 22 (4) (2012) 689–699.

[15] M. A. Müller, F. Allgöwer, Distributed economic mpc: a framework for cooperative control problems, in: In Proceedings of the 19th IFAC World Congress, Cape Town, South Africa. August 24-29, 2014, 2014.

[16] F. A. Bayer, M. A. Müller, F. Allgöwer, Tube-based robust economic model predictive control, Journal of Process Control 24 (8) (2014) 1237–1246.

[17] E. A. Idris, S. Engell, Economics-based nmpc strategies for the operation and control of a continuous catalytic distillation process, Journal of Process Control 22 (10) (2012) 1832–1843.

[18] R. Findeisen, F. Allgöwer, Computational delay in nonlinear model predictive control, in: 7th International Symposium on Advanced Control of Chemical Processes (ADCHEM 2003), Hong-Kong, 11-14 January 2004, 2004.

[19] A. V. Fiacco, Introduction to sensitivity and stability analysis in nonlinear programming, Elsevier, 1983.

[20] W. C. Li, L. T. Biegler, Process control strategies for constrained nonlinear systems, Industrial & engineering chemistry research 27 (8) (1988) 1421–1433.

[21] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, F. Allgöwer, Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations, Journal of Process Control 12 (4) (2002) 577–585.

[22] C. Büskens, H. Maurer, Sensitivity analysis and real-time optimization of parametric nonlinear programming problems, in: Online Optimization of Large Scale Systems, Springer, 2001, pp. 3–16.

[23] J. V. Kadam, W. Marquardt, Sensitivity-based solution updates in closed-loop dynamic optimization, in: 7th IFAC Symposium on Dynamics and Control of Process Systems 2004 (DYCOPS -7), Cambridge, USA, 5-7 July, 2004, 2004.

[24] V. M. Zavala, L. T. Biegler, The advanced-step nmpc controller: Optimality, stability and robustness, Automatica 45 (1) (2009) 86–93.

[25] E. Suwartadi, J. Jäschke, Fast sensitivity-based nonlinear economic model predictive control with degenerate nlp, in: 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018: Shenyang, China, 25–27 July 2018, 2018.

[26] L. Biegler, X. Yang, G. Fischer, Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization, Journal of Process Control 30 (2015) 104–116.

[27] L. N. Vicente, S. J. Wright, Local convergence of a primal-dual method for degenerate nonlinear programming, Computational Optimization and Applications 22 (3) (2002) 311–328.

[28] J. Jäschke, X. Yang, L. T. Biegler, Fast economic model predictive control based on nlp-sensitivities, Journal of Process Control 24 (8) (2014) 1260–1272.

[29] V. Kungurtsev, J. Jäschke, A predictor-corrector path-following algorithm for dual-degenerate parametric optimization problems, SIAM Journal on Optimization 27 (1) (2017) 538–564.

[30] M. Diehl, Real-time optimization for large scale nonlinear processes, Ph.D. thesis, University of Heidelberg (2001).

[31] E. Suwartadi, V. Kungurtsev, J. Jäschke, Sensitivity-based economic nmpc with a path-following approach, Processes 5 (1) (2017) 8.

[32] D. Ralph, S. Dempe, Directional derivatives of the solution of a parametric nonlinear program, Mathematical programming 70 (1-3) (1995) 159–172.

[33] J. Kyparisis, Sensitivity analysis for nonlinear programs and variational inequalities with nonunique multipliers, Mathematics of operations research 15 (2) (1990) 286–298.

[34] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, Nonlinear programming: theory and algorithms, John Wiley & Sons, 2013.

[35] J. Gauvin, A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming, Mathematical Programming 12 (1) (1977) 136–138.

[36] R. Janin, Directional derivative of the marginal function in nonlinear programming, Springer, 1984, pp. 110–126.

[37] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, M. Diehl, CasADi – A software framework for nonlinear optimization and optimal control, Mathematical Programming Computation.

[38] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical programming 106 (1) (2006) 25–57.

[39] B. A. Murtagh, M. A. Saunders, A projected lagrangian algorithm and its implementation for sparse nonlinear constraints, Algorithms for Constrained Minimization of Smooth Nonlinear Functions (1982) 84–117.

[40] IBM ILOG CPLEX Optimizers 12.7.1.

[41] S. Skogestad, I. Postlethwaite, Multivariable feedback control: analysis and design, Vol. 2, Wiley New York, 2007.

Declaration of interests

 The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: