MDPI

*Article*

# Sensitivity-based Economic NMPC with a Pathfollowing Approach

**Eka Suwartadi [1], Vyacheslav Kungurtsev [2] and Johannes Jäschke [1,*]**

[1] Affiliation 1; Department of Chemical Engineering, Norwegian University of Science and Technology. Email: suwartad@ntnu.no, jaschke@ntnu.no.
[2] Affiliation 2; Department of Computer Science, Czech Technical University in Prague. V. Kungurtsev was supported by the Czech Science Foundation project 17-26999S. Email: vyacheslav.kungurtsev@fel.cvut.cz.
* Correspondence: jaschke@ntnu.no

Academic Editor: Dominique Bonvin
Version February 3, 2017 submitted to Processes; Typeset by LaTeX using class file mdpi.cls

**Abstract:** We present a sensitivity-based predictor-corrector pathfollowing algorithm for fast nonlinear model predictive control (NMPC), and demonstrate it on a large case study with an economic cost function. The pathfollowing method is applied within the Advanced-Step NMPC framework to obtain fast and accurate approximate solutions of the NMPC problem. In our approach, we solve a sequence of quadratic programs to trace the optimal NMPC solution along a parameter change. A distinguishing feature of the path-following algorithm in this paper is that the strongly active inequality constraints are included as equality constraints in the quadratic programs, while the weakly active constraints are left as inequalities. This leads to close tracking of the optimal solution. The approach is applied to an economic NMPC case study consisting of a process with a reactor, a distillation column and a recycle. We compare the pathfollowing NMPC solution with an ideal NMPC solution, which is obtained by solving the full nonlinear programming problem. Our simulations show that the proposed algorithm effectively traces the exact solution.

**Keywords:** Fast economic NMPC; NLP sensitivity; Pathfollowing algorithm; Nonlinear programming; Dynamic optimization

## 1. Introduction

The idea of economic model predictive control (MPC) is to integrate the economic optimization layer and the control layer in the process control hierarchy into a single dynamic optimization layer. While classic model predictive control approaches typically employ a quadratic objective to minimize the error between the setpoints and selected measurements, economic MPC adjusts the inputs to minimize the economic cost of operation directly. This makes it possible to optimize the cost during transient operation of the plant. In recent years, this has become increasingly desirable, as stronger competition, volatile energy prices and rapidly changing product specifications require agile plant operations, where also transients are optimized to maximize profit.

The first industrial implementations of economic MPC are reported in [1,2] for oil refinery applications. The development of theory and stability analysis for economic MPC arose almost a decade afterwards see e.g., [3,4]. Recent progress on economic MPC is reviewed and surveyed in [5,6]. Most of the current research activities focus on the stability analysis of economic MPC, and do not discuss its performance (an exception is [7]).

Because nonlinear process models are often used for economic optimization, a potential drawback of economic MPC is that it requires to solve a large-scale nonlinear optimization problem (NLP) associated with the nonlinear model predictive control (NMPC) problem at every sample time.

The solution of this NLP may take a significant amount of time [8], and this can lead to performance degradation and even to instability of the closed-loop system [9].

To reduce the detrimental effect of computational delay in NMPC, several sensitivity-based methods were proposed [14,15,19]. All these fast sensitivity approaches exploit the fact that the NMPC optimization problems are identical at each sample time, except for one varying parameter: the initial state. Instead of solving the full nonlinear optimization problem when new measurements of the state become available, these approaches use the sensitivity of the NLP solution at a previously computed iteration to obtain fast approximate solutions to the new NMPC problem. These approximate solutions can be computed and implemented in the plant with minimal delay. A recent overview of the developments in fast sensitivity-based nonlinear MPC is given in [10], and a comparison of different approaches to obtain sensitivity updates for NMPC is compiled in the paper by Wolf and Marquardt [21] .

Diehl et al. [11] proposed the concept of real-time iterations, in which the full NLP is not solved at all during the MPC iterations. Instead, at each NMPC sampling time, a single QP related to the sequential quadratic programming (SQP) iteration for solving the full NLP is solved. The real-time iteration scheme contains two phases: 1) Preparation phase and 2) Feedback phase. In the preparation phase, the model derivatives are evaluated using a predicted state measurement, and a QP is formulated based on data of this predicted state. In the feedback phase, once the new initial state is available, the QP is updated to include the new initial state, and solved for the control input that is injected into the plant. The real-time iteration scheme has been applied to economic NMPC in the context of wind turbine control [26,27]. Similar to the real-time iteration scheme are the approaches by Ohtsuka [17], and the early paper by Li and Biegler [25] where one single Newton-like iteration is performed per sampling time.

A different approach, the Advanced-Step NMPC (asNMPC), was proposed by Zavala and Biegler [14]. The asNMPC approach involves solving the full NLP at every sample time. However, the full NLP solution is computed in advance for a predicted initial state. Once the new state measurement is available, the NLP solution is corrected using a fast sensitivity update to match the measured or estimated initial state. A simple sensitivity update scheme is implemented in the software package sIPOPT [16]. However, active set changes are handled rather heuristically, see [28] for an overview. Kadam and Marquardt [44] proposed a similar approach, where nominal NLP solutions are updated by solving QPs in a neighboring-extremal scheme, see also [19,20].

The framework of asNMPC was also applied by Jäschke and Biegler [22] who use a multiple-step predictor pathfollowing algorithm to correct the NLP predictions. Their approach included measures to handle active set changes rigorously, and their pathfollowing Advanced-Step NMPC algorithm is also the first one to handle non-unique Lagrange multipliers.

The contribution of this paper is to apply an improved path-following method for correcting the NLP solution within the Advanced-Step NMPC framework. In particular, we replace the *predictor* pathfollowing method from [22] by a *predictor-corrector* method, and demonstrate numerically that the method works efficiently on a large-scale case study. We present how the asNMPC with the predictor-corrector pathfollowing algorithm performs in presence of measurement noise, and compare it with a pure predictor pathfollowing asNMPC approach and an ideal NMPC approach, where the NLP is assumed to be solved instantly. We also give a brief discussion about how our method differs from previously published approaches.

The structure of this paper is the following. We start by introducing the ideal NMPC and Advanced-Step NMPC frameworks in Section 2, and give a description of our pathfollowing algorithm together with some relevant background material and a brief discussion in Section 3. The proposed algorithm is applied to a process with a reactor, distillation and recycle in Section 4, where we consider the cases with and without measurement noise, and discuss the results. The paper is closed with our conclusions in Section 5.

**Algorithm 1:** General NMPC Algorithm

1. set $k \leftarrow 0$
2. **while** *MPC is running* **do**
3. 1. Measure or estimate $x_k$.
4. 2. Assign the initial state: set $\mathbf{z}_0 = x_k$ .
5. 3. Solve the optimization problem $\mathcal{P}_{nmpc}$ to find $\mathbf{v}_0^*$.
6. 4. Assign the plant input $\mathbf{u}_k = \mathbf{v}_0^*$.
7. 5. Inject $\mathbf{u}_k$ to the plant (1).
8. 6. set $k \leftarrow k + 1$

## 2. NMPC Problem Formulations

### 2.1. The NMPC Problem

We consider a nonlinear discrete-time dynamic system

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

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ denotes the state variable, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the control input, and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is a continuous model function which calculates the next state $\mathbf{x}_{k+1}$ from the previous state $\mathbf{x}_k$ and control input $\mathbf{u}_k$ where $k \in \mathbb{N}$. This system is optimized by a nonlinear model predictive controller which solves the problem

$$\begin{aligned} (\mathcal{P}_{nmpc}): \quad & \min_{\mathbf{z}_l, \mathbf{v}_l} && \Psi(\mathbf{z}_N) + \sum_{l=0}^{N-1} \psi(\mathbf{z}_l, \mathbf{v}_l) \\ \text{s.t.} \quad & \mathbf{z}_{l+1} = f(\mathbf{z}_l, \mathbf{v}_l) && l = 0, \ldots, N-1, \\ & \mathbf{z}_0 = \mathbf{x}_k, \\ & (\mathbf{z}_l, \mathbf{v}_l) \in \mathcal{Z}, && l = 0, \ldots, N-1, \\ & \mathbf{z}_N \in \mathcal{X}_f, \end{aligned} \tag{2}$$

at each sample time. Here $\mathbf{z}_l \in \mathbb{R}^{n_x}$ is the predicted state variable, $\mathbf{v}_l \in \mathbb{R}^{n_u}$ is the predicted control input, and $\mathbf{z}_N \in \mathcal{X}_f$ is the final predicted state variable restricted to the terminal region $\mathcal{X}_f \in \mathbb{R}^{n_x}$. The stage cost is denoted by $\psi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and the terminal cost by $\Psi : \mathcal{X}_f \to \mathbb{R}$. Further, $\mathcal{Z}$ denotes the path constraints, i.e., $\mathcal{Z} = \{(\mathbf{z}, \mathbf{v}) \mid q(\mathbf{z}, \mathbf{v}) \leq 0\}$, where $q : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_q}$.

The solution of the optimization problem $\mathcal{P}_{nmpc}$ is denoted $\{\mathbf{z}_0^*, \ldots, \mathbf{z}_N^*, \mathbf{v}_0^*, \ldots, \mathbf{v}_{N-1}^*\}$. At sample time $k$ an estimate or measurement of the state $\mathbf{x}_k$ is obtained, and problem $\mathcal{P}_{nmpc}$ is solved. Then the first part of the optimal control sequence is assigned as plant input, such that $\mathbf{u}_k = \mathbf{v}_0^*$. This first part of the solution to $\mathcal{P}_{nmpc}$ defines an implicit feedback law $\mathbf{u}_k = \kappa(\mathbf{x}_k)$ and the system will evolve according to $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \kappa(\mathbf{x}_k))$. At the next sample time $k + 1$, when the measurement of the new state $\mathbf{x}_{k+1}$ is obtained, the procedure is repeated. The NMPC algorithm is summarized in Algorithm 1.

### 2.2. Ideal NMPC and Advanced-Step NMPC framework

For achieving optimal economic performance and good stability properties, problem $\mathcal{P}_{nmpc}$ needs to be solved instantly, so that the optimal input can be injected without time delay as soon as the values of the new states are available. We refer to this hypothetical case without computational delay as *ideal NMPC*.

In practice, there will always be some time delay between obtaining the updated values of the states and injecting the updated inputs into the plant. The main reason for this delay is the time it

requires to solve the optimization problem $\mathcal{P}_{nmpc}$. As the process models become more advanced, solving the optimization problems requires more time, and the computational delay cannot be neglected any more. This has led to the development of fast sensitivity-based NMPC approaches. One such approach that will be a adopted in this paper is the *Advanced-Step NMPC* (asNMPC) approach approach [14]. It is based on the following steps:

1. Solve the NMPC problem at time $k$ with a predicted state value of time $k+1$
2. When the measurement $\mathbf{x}_{k+1}$ becomes available at time $k+1$, compute an approximation of the NLP solution using fast sensitivity methods.
3. Update $k \leftarrow k+1$ and repeat from Step 1.

Zavala and Biegler proposed a fast one-step sensitivity update that is based on solving a linear system of equations [14]. Under some assumptions this corresponds to a first-order Taylor approximation of the optimal solution. In particular, this approach requires strict complementarity of the NLP solution, which ensures no changes in the active set. A more general approach involves allowing for changes in the active set and making several sensitivity updates. This was proposed in [22], and will be developed further in this paper.

## 3. Sensitivity-based Pathfollowing NMPC

In this section we present some fundamental sensitivity results from the literature, and then use them in a pathfollowing scheme for obtaining fast approximate solutions to the NLP.

### *3.1. Sensitivity Properties of NLP*

The dynamic optimization problem (2) can be cast as a general parametric NLP problem

$$\begin{aligned} (\mathcal{P}_{NLP}): \min_{\chi} \quad & F(\chi, \mathbf{p}) \\ \text{s.t.} \quad & c(\chi, \mathbf{p}) = 0 \\ & g(\chi, \mathbf{p}) \leq 0, \end{aligned} \tag{3}$$

where $\chi \in \mathbb{R}^{n_\chi}$ are the decision variables (which generally include the state variables and the control input $n_\chi = n_x + n_u$) and $\mathbf{p} \in \mathbb{R}^{n_p}$ is the parameter, which is typically the initial state variable $\mathbf{x}_k$. In addition, $F : \mathbb{R}^{n_\chi} \times \mathbb{R}^{n_p} \to \mathbb{R}$ is the scalar objective function, $c : \mathbb{R}^{n_\chi} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_c}$ denotes the equality constraints, and finally $g : \mathbb{R}^{n_\chi} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_g}$ denotes the inequality constraints. The instances of Problem (3) that are solved at each sample time differ only in the parameter $\mathbf{p}$.

The Lagrangian function of this problem is defined as

$$\mathcal{L}(\chi, \mathbf{p}, \lambda, \mu) = F(\chi, \mathbf{p}) + \lambda^T c(\chi, \mathbf{p}) + \mu^T g(\chi, \mathbf{p}), \tag{4}$$

and the KKT (Karush-Kuhn Tucker) conditions are

$$\begin{aligned} c(\mathbf{x}, \mathbf{p}) = 0, \quad & g(\mathbf{x}, \mathbf{p}) \leq 0 \qquad (primal\ feasibility) \\ \mu \geq 0, \quad & (dual\ feasibility) \\ \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{p}, \lambda, \mu) = 0, \quad & (stationary\ condition) \\ \mu^T g(\mathbf{x}, \mathbf{p}) = 0, \quad & (complementary\ slackness). \end{aligned} \tag{5}$$

In order for the KKT condition to be a necessary condition of optimality, we require a constraint qualification (CQ) to hold. In this paper we will assume that the LICQ holds:

**Definition 1** (LICQ)**.** Given a vector $\mathbf{p}$ and a point $\chi$, the *linear independence constraint qualification* (LICQ) holds at $\chi$ if the set of vectors $\left\{ \{\nabla_\chi c_i(\chi, \mathbf{p})\}_{i \in \{1,\dots,n_c\}} \cup \{\nabla_\chi g_i(\chi, \mathbf{p})\}_{i: g_i(\chi, \mathbf{p}) = 0} \right\}$ are linearly independent.

The LICQ implies that the multipliers $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ satisfying the KKT conditions are unique. If additionally a suitable second order condition holds, then the KKT conditions guarantee a unique local minimum. A suitable second order condition states that the Hessian matrix has to be positive definite in a set of appropriate directions, defined in the following property:

**Definition 2** (SSOSC)**.** The *strong second order sufficient condition* (SSOSC) holds at $\boldsymbol{\chi}$ with multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ if $\mathbf{d}^T \nabla^2_{\boldsymbol{\chi}} \mathcal{L}(\boldsymbol{\chi}, \mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{\mu})\ \mathbf{d} > 0$ for all $\mathbf{d} \neq 0$ such that $\nabla_{\boldsymbol{\chi}} c(\boldsymbol{\chi}, \mathbf{p})^T \mathbf{d} = 0$ and $\nabla_{\boldsymbol{\chi}} g_i(\boldsymbol{\chi}, \mathbf{p})^T \mathbf{d} = 0$ for $i$ such that $g_i(\boldsymbol{\chi}, \mathbf{p}) = 0$ and $\mu_i > 0$.

For a given $\mathbf{p}$, denote the solution to (3) by $\boldsymbol{\chi}^*(\mathbf{p}), \boldsymbol{\lambda}^*(\mathbf{p}), \boldsymbol{\mu}^*(\mathbf{p})$, and if no confusion is possible we omit the argument and write simply $\boldsymbol{\chi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$. We are interested in knowing how the solution changes with a perturbation in the parameter $\mathbf{p}$. Before we state a first sensitivity result, we define another important concept:

**Definition 3** (SC)**.** Given a vector $\mathbf{p}$ and a solution $\boldsymbol{\chi}^*$ with vectors of multipliers $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$, *strict complimentary* (SC) holds if $\mu_i^* - g_i(\boldsymbol{\chi}^*, \mathbf{p}) > 0$ for each $i = 1, \ldots, n_g$.

Now we are ready to state the result below given by Fiacco [24].

**Theorem 4** (Implicit function theorem applied to optimality conditions)**.** *Let $\boldsymbol{\chi}^*(\mathbf{p})$ be a KKT point that satisfies (5), and assume that LICQ, SSOSC, and SC hold at $\boldsymbol{\chi}^*$. Further let the function $F$, $c$, $g$ be at least $k+1$ times differentiable in $\boldsymbol{\chi}$ and $k$ times differentiable in $\mathbf{p}$. Then*

- *$\boldsymbol{\chi}^*$ is an isolated minimizer, and the associated multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are unique.*
- *for $\mathbf{p}$ in a neighborhood of $\mathbf{p}_0$ the set of active constraints remains unchanged.*
- *for $\mathbf{p}$ in a neighborhood of $\mathbf{p}_0$ there exists a $k$ times differentiable function $\sigma(\mathbf{p}) = \begin{bmatrix} \boldsymbol{\chi}^*(\mathbf{p})^T & \boldsymbol{\lambda}^*(\mathbf{p})^T & \boldsymbol{\mu}^*(\mathbf{p})^T \end{bmatrix}$, that corresponds to a locally unique minimum for (3).*

**Proof.** See Fiacco [24]. □

Using this result, the sensitivity of the optimal solution $\boldsymbol{\chi}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ in a small neighborhood of $\mathbf{p}_0$ can be computed by solving a system of linear equations that arises from applying the implicit function theorem to the KKT conditions of (3):

$$\begin{bmatrix} \nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} \mathcal{L}(\boldsymbol{\chi}*, \mathbf{p}_0, \boldsymbol{\lambda}*, \boldsymbol{\mu}*) & \nabla_{\boldsymbol{\chi}} c(\boldsymbol{\chi}*, \mathbf{p}_0) & \nabla_{\boldsymbol{\chi}} g_A(\boldsymbol{\chi}*, \mathbf{p}_0) \\ \nabla_{\boldsymbol{\chi}} c(\boldsymbol{\chi}*, \mathbf{p}_0)^T & 0 & 0 \\ \nabla_{\boldsymbol{\chi}} g_A(\boldsymbol{\chi}*, \mathbf{p}_0)^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{p}} \boldsymbol{\chi} \\ \nabla_{\mathbf{p}} \boldsymbol{\lambda} \\ \nabla_{\mathbf{p}} \boldsymbol{\mu} \end{bmatrix} = - \begin{bmatrix} \nabla^2_{\mathbf{p}\boldsymbol{\chi}} \mathcal{L}(\boldsymbol{\chi}*, \mathbf{p}_0, \boldsymbol{\lambda}*, \boldsymbol{\mu}*) \\ \nabla_{\mathbf{p}} c(\boldsymbol{\chi}*, \mathbf{p}_0) \\ \nabla_{\mathbf{p}} g_A(\boldsymbol{\chi}*, \mathbf{p}_0) \end{bmatrix}. \tag{6}$$

Here the constraint gradients with subscript, $g_A$ indicate that we only include the vectors and components of the Jacobian corresponding to the active inequality constraints at $\boldsymbol{\chi}$, i.e., $i \in A$ if $g_i(\boldsymbol{\chi}, \mathbf{p}_0) = 0$. Denoting the solution of the equation above as $\begin{bmatrix} \nabla_{\mathbf{p}} \boldsymbol{\chi} & \nabla_{\mathbf{p}} \boldsymbol{\lambda} & \nabla_{\mathbf{p}} \boldsymbol{\mu} \end{bmatrix}^T$, for small $\Delta\mathbf{p}$ we obtain a good estimate

$$\boldsymbol{\chi}(\mathbf{p}_0 + \triangle\mathbf{p}) = \boldsymbol{\chi}^* + \nabla_{\mathbf{p}} \boldsymbol{\chi}\, \triangle\mathbf{p}, \tag{7}$$
$$\boldsymbol{\lambda}(\mathbf{p}_0 + \triangle\mathbf{p}) = \boldsymbol{\lambda}^* + \nabla_{\mathbf{p}} \boldsymbol{\lambda}\, \triangle\mathbf{p}, \tag{8}$$
$$\boldsymbol{\mu}(\mathbf{p}_0 + \triangle\mathbf{p}) = \boldsymbol{\mu}^* + \nabla_{\mathbf{p}} \boldsymbol{\mu}\, \triangle\mathbf{p}, \tag{9}$$

of the solution to NLP problem (3) at the parameter value $\mathbf{p}_0 + \Delta\mathbf{p}$. This approach was applied by Zavala and Biegler [14].

If $\Delta\mathbf{p}$ becomes large, the approximate solution may no longer be accurate enough, because the SC assumption implies that the active set cannot change. While that is usually true for small perturbations, large changes in $\Delta\mathbf{p}$ may very well induce active set changes.

It can be seen that the sensitivity system corresponds to the stationarity conditions for a particular QP. This is not coincidental. It can be shown that for $\Delta\mathbf{p}$ small enough, the set $\{i : \mu(\bar{\mathbf{p}})_i > 0\}$ is constant for $\bar{\mathbf{p}} = \mathbf{p}_0 + \Delta\mathbf{p}$. Thus we can form a QP wherein we are potentially moving off of weakly active constraints while staying on the strongly active ones. The primal-dual solution of this QP is in fact the directional derivative of the primal-dual solution path $\chi^*(\mathbf{p}), \lambda^*(\mathbf{p}), \mu^*(\mathbf{p})$.

**Theorem 5.** *Let $F, c, g$ be twice continuously differentiable in $\mathbf{p}$ and $\chi$ near $(\chi^*, \mathbf{p}_0)$, and let the LICQ and SSOSC hold at $(\chi^*, \mathbf{p}_0)$. Then the solution $(\chi^*(\mathbf{p}), \lambda^*(\mathbf{p}), \mu^*(\mathbf{p}))$ is Lipschitz continuous in a neighborhood of $(\chi^*, \lambda^*, \mu^*, \mathbf{p}_0)$, and the solution function $(\chi^*(\mathbf{p}), \lambda^*(\mathbf{p}), \mu^*(\mathbf{p}))$ is directionally differentiable.*

*Moreover, the directional derivative uniquely solves the following quadratic problem*

$$
\begin{aligned}
\min_{\triangle\chi} \quad & \tfrac{1}{2}\triangle\chi^T \nabla^2_{\chi\chi}\mathcal{L}(\chi^*, \mathbf{p}_0, \lambda^*, \mu^*)\triangle\chi + \triangle\chi^T \nabla^2_{\mathbf{p}\chi}\mathcal{L}(\chi^*, \mathbf{p}_0, \lambda^*, \mu^*)\triangle\mathbf{p} && (10)\\
\text{s.t.} \quad & \nabla_\chi c_i(\chi^*, \mathbf{p}_0)^T \triangle\chi + \nabla_{\mathbf{p}} c_i(\chi^*, \mathbf{p}_0)^T \triangle\mathbf{p} = 0 && i = 1, \dots n_c\\
& \nabla_\chi g_j(\chi^*, \mathbf{p}_0)^T \triangle\chi + \nabla_{\mathbf{p}} g_j(\chi^*, \mathbf{p}_0)^T \triangle\mathbf{p} = 0 && j \in K_+\\
& \nabla_\chi g_j(\chi^*, \mathbf{p}_0)^T \triangle\chi + \nabla_{\mathbf{p}} g_j(\chi^*, \mathbf{p}_0)^T \triangle\mathbf{p} \leq 0 && j \in K_0,
\end{aligned}
$$

*where $K_+ = \{j \in \mathbb{Z} : \mu_j > 0\}$ is the strongly active set and $K_0 = \{j \in \mathbb{Z} : \mu_j = 0\}$ is the union of the weakly active and inactive set.*

**Proof.** See [42, Section 5.1-5.2] and [43, Proposition 3.4.1]. □

The theorem above gives solution of the perturbed NLP (3) by solving a QP problem. Note that regardless of the inertia of the Lagrangian Hessian, if the SSOSC holds, it is positive definite on the null-space of the equality constraints, and thus the QP defined is convex with an easily obtainable finite global minimizer. In [30] it is noted that as the solution to this QP is the directional derivative of the primal-dual solution of the NLP, it is a *predictor* step, a tangential first-order estimate of the change in the solution subject to a change in the parameter. We refer to the QP (10) as a *pure-predictor*. Note that obtaining the sensitivity via (10) instead of (6) has the advantage that changes in the active set can be accounted for correctly and strict complementarity (SC) is not required. On the other hand, when SC does hold (6) and (10) are equivalent.

### *3.2. Pathfollowing based on Sensitivity Properties*

Equation (6) and the QP (10) describe the change in the optimal solutions for small perturbations. They cannot be guaranteed to reproduce the optimal solution accurately for larger perturbations, because of curvature in the solution path and active set changes that happen further away from the linearization point. One approach to handle such cases is to divide the overall perturbation into several smaller intervals, and to iteratively use the sensitivity to track the path of optimal solutions.

The general idea of a pathfollowing method is to reach the solution of the problem at a final parameter value $\mathbf{p}_f$ by tracing a sequence of solutions $(\chi_k, \lambda_k, \mu_k)$ for a series of parameter values $\mathbf{p}(t_k) = (1 - t_k)\mathbf{p}_0 + t_k \mathbf{p}_f$ with $0 = t_0 < t_1 < \dots < t_k < \dots < t_N = 1$. The new direction is found by evaluating the sensitivity at the current point. This is similar to an Euler integration for ordinary differential equations.

However, just as in the case of integrating differential equations with an Euler method, a pathfollowing algorithm that is only based on the sensitivity calculated by the pure predictor QP may fail to track the solution accurately enough, and may lead to poor solutions. To address this problem, a common approach is to include elements that are similar to a Newton step, which force

the pathfollowing algorithm towards the true solution. It has been found that such a corrector element can be easily included into a QP that is very similar to the predictor QP (10). Consider approximating (3) by a QP, linearizing with respect to both $\chi$ and $\mathbf{p}$, but again enforcing equality of the strongly active constraints, as we expect them to remain strongly active at a perturbed NLP:

$$\min_{\triangle\chi,\triangle\mathbf{p}} \frac{1}{2}\triangle\chi^T\nabla^2_{\chi\chi}\mathcal{L}\left(\chi^*,\mathbf{p}_0,\lambda^*,\mu^*\right)\triangle\chi + \triangle\chi^T\nabla^2_{\mathbf{p}\chi}\mathcal{L}\left(\chi^*,\mathbf{p}_0,\lambda^*,\mu^*\right)\triangle\mathbf{p} + \nabla_\chi F^T\triangle\chi + \nabla_\mathbf{p} F\triangle\mathbf{p} + \frac{1}{2}\triangle\mathbf{p}^T\nabla^2_{\mathbf{pp}}\mathcal{L}\left(\chi^*,\mathbf{p}_0,\lambda^*,\mu^*\right)\triangle\mathbf{p} \tag{11}$$

$$\begin{aligned}
\text{s.t.} & c_i\left(\chi^*,\mathbf{p}_0\right) + \nabla_\chi c_i\left(\chi^*,\mathbf{p}_0\right)^T\triangle\chi + \nabla_\mathbf{p} c_i\left(\chi^*,\mathbf{p}_0\right)^T\triangle\mathbf{p} = 0 \qquad & i = 0,\dots n_c\\
& g_j\left(\chi^*,\mathbf{p}_0\right) + \nabla_\chi g_j\left(\chi^*,\mathbf{p}_0\right)^T\triangle\chi + \nabla_\mathbf{p} g_j\left(\chi^*,\mathbf{p}_0\right)^T\triangle\mathbf{p} = 0 \qquad & j \in K_+\\
& g_j\left(\chi^*,\mathbf{p}_0\right) + \nabla_\chi g_j\left(\chi^*,\mathbf{p}_0\right)^T\triangle\chi + \nabla_\mathbf{p} g_j\left(\chi^*,\mathbf{p}_0\right)^T\triangle\mathbf{p} \leq 0 \qquad & j \in K_0.
\end{aligned}$$

In our NMPC problem $\mathcal{P}_{nmpc}$, the parameter $\mathbf{p}$ corresponds to the current "initial" state, $\mathbf{x}_k$. Moreover, the cost function is independent of $\mathbf{p}$, and we have that $\nabla_\mathbf{p} F = 0$. Since the parameter enters the constraints linearly, we have that $\nabla_\mathbf{p} c$ and $\nabla_\mathbf{p} g$ are constants. With these facts, the above QP simplifies to

$$\begin{aligned}
\min_{\triangle\chi} \quad & \tfrac{1}{2}\triangle\chi^T\nabla^2_{\chi\chi}\mathcal{L}\left(\chi^*,\mathbf{p}_0+\triangle\mathbf{p},\lambda^*,\mu^*\right)\triangle\chi + \nabla_\chi F^T\triangle\chi \qquad & (12)\\
\text{s.t.} \quad & c_i\left(\chi^*,\mathbf{p}_0+\triangle\mathbf{p}\right) + \nabla_\chi c_i\left(\chi^*,\mathbf{p}_0+\triangle\mathbf{p}\right)^T\triangle\chi = 0 \qquad & i = 1,\dots n_c\\
& g_j\left(\chi^*,\mathbf{p}_0+\triangle\mathbf{p}\right) + \nabla_\chi g_j\left(\chi^*,\mathbf{p}_0+\triangle\mathbf{p}\right)^T\triangle\chi = 0 \qquad & j \in K_+\\
& g_j\left(\chi^*,\mathbf{p}_0+\triangle\mathbf{p}\right) + \nabla_\chi g_j\left(\chi^*,\mathbf{p}_0+\triangle\mathbf{p}\right)^T\triangle\chi \leq 0 \qquad & j \in K_0.
\end{aligned}$$

We denote the QP formulation (12) as the *predictor-corrector*. We note that this QP is similar to the QP proposed in the real-time iteration scheme [11]. However, it is not quite the same, as we enforce the strongly active constraints as equality constraints in the QP. As explained in [30], this particular QP tries to estimate how the NLP solution changes as the parameter does in the predictor component and refines the estimate, in more closely satisfying the KKT conditions at the new parameter, as a corrector.

The predictor-corrector QP (12) is well suited for use in a pathfollowing algorithm, where the optimal solution path is tracked from $\mathbf{p_0}$ to a final value $\mathbf{p}_f$ along a sequence of parameter points $\mathbf{p}(t_k) = (1-t_k)\,\mathbf{p}_0 + t_k\,\mathbf{p}_f$ with $0 = t_0 < t_1 < \ldots < t_k < \ldots < t_N = 1$. At each point $\mathbf{p}(t_k)$, the QP is solved, and the primal-dual solutions updated as

$$\chi(t_{k+1}) = \chi(t_k) + \Delta\chi \tag{13}$$
$$\lambda(t_{k+1}) = \Delta\lambda \tag{14}$$
$$\mu(t_{k+1}) = \Delta\mu, \tag{15}$$

where $\Delta\chi$ is obtained from the primal solution of QP (12), and where $\Delta\lambda$ and $\Delta\mu$ correspond to the Lagrange multipliers of QP (12).

Changes in the active set along the path are detected by the QP as follows: If a constraint becomes inactive at some point along the path, the corresponding multiplier $\mu_j$ will first become weakly active, i.e. it will be added to the set $K_0$. Since it is not included as an equality constraint, the next QP solution can move away from the constraint. Similarly, if a new constraint $g_j$ becomes active along the path, it will make the corresponding linearized inequality constraint in the QP active, and be tracked further along the path.

The resulting pathfollowing algorithm is summarized with its main steps in Algorithm 2, and we are now in the position to apply it in the Advanced-Step NMPC setting described in Section 2.2. In particular, the pathfollowing algorithm is used to find a fast approximation of the optimal

**Algorithm 2:** Pathfollowing Algorithm

**Input:** initial variables from NLP $\chi^*(\mathbf{p}_0), \lambda^*(\mathbf{p}_0), \mu^*(\mathbf{p}_0)$
fix stepsize $\triangle t$ and set $N = \frac{1}{\Delta t}$
set initial parameter value $\mathbf{p}_0$,
set final parameter value $\mathbf{p}_f$,
set $t = 0$,
set constant $0 < \alpha_1 < 1$.
**Output:** primal variable $\chi$ and dual variables $\lambda, \mu$ along the path

```
for k ← 1 to N do
    Compute step Δp = p_k − p_{k−1}
    Solve QP problem ;                                   /* to obtain Δχ, Δλ, Δμ   */
    if QP is feasible then
        /* perform update                                                          */
        χ ← χ + Δχ;                                      /* update primal variables */
        Update dual variables appropriately; using equations (8) and (9) for the
        pure-predictor method or (14) and (15) for the predictor-corrector method
        t ← t + Δt ;                                     /* update stepsize */
        k ← k + 1
    else
        /* QP is infeasible, reduce QP stepsize                                    */
        △t ← α_1 △t
        t ← t − α_1 △t
```

NLP solution corresponding to the new available state measurement, which is done by following the optimal solution path from the predicted state to the measured state.

### *3.3. Discussion of the path-following asNMPC approach*

In this section, we discuss some characteristics of the path-following asNMPC approach presented in this paper. We also present a small example to demonstrate the effect of including the strongly active constraints as equality constraints in the QP.

A reader who is familiar with the real-time iteration scheme [11] will have realized that the QPs (12) that are solved in our path-following algorithm are similar to the ones proposed and solved in the real-time iteration scheme. However, there are some fundamental differences between the standard real-time iteration scheme as described in [11] and the asNMPC with a path-following approach.

This work is set in the Advanced-Step NMPC framework, i.e. at every time-step, the full NLP is solved for a predicted state. When the new measurement becomes available, the precomputed NLP solution is updated by tracking the optimal solution curve from the predicted initial state to the new measured or estimated state. Any numerical homotopy algorithm can be used to update the NLP solution, and we have presented a suitable one in this paper. Note that the solution of the last QP along the path corresponds to the updated NLP solution, and only the inputs computed in this last QP will be injected into the plant.

The situation is quite different in the RTI scheme described in [11]. Here the NLP is not solved at all during the MPC sampling times. Instead, at each sampling time, a single QP is solved, and the computed input is applied to the plant. This will require very fast sampling times, and if the QP fails to track the true solution due to very large disturbances, similar measures as in the Advanced-Step NMPC procedure (i.e. solving the full NLP) must be performed to get the controller "on track" again. Note that the inputs computed from *every* QP are applied to the plant, and not as in our path-following asNMPC only the input computed in the last QP along the homotopy.

Finally, in the QPs of the previously published real-time iteration schemes [11] all inequality constraints are linearized and included as QP *inequality constraints*. Our approach in this paper,

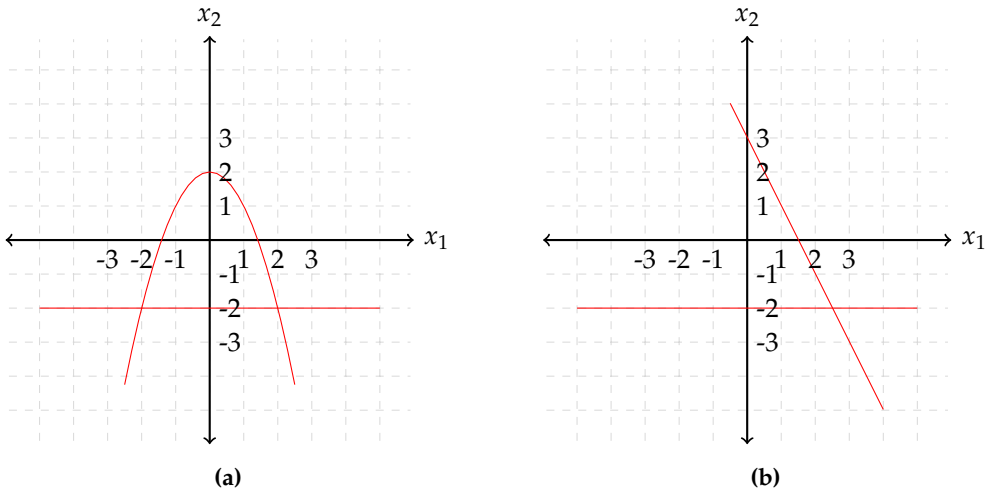


**Figure 1.** Constraints of NLP (16) in Example 1, and their linearization at $\hat{x} = (0.1, -2)$.

however, distinguishes between strongly and weakly active inequality constraints. Strongly active inequalities are included as linearized *equality constraints* in the QP, while weakly active constraints are linearized and added as *inequality constraints* to the QP. This ensures that the true solution path is tracked more accurately also when the full Hessian of the optimization problem becomes non-convex. We illustrate this in the small example below.

**Example 1.** Consider the following parametric "NLP"

$$\begin{aligned} \min_{x} \quad & x_1^2 - x_2^2 \\ \text{s.t.} \quad & x_2 \geq -2 + t, \\ & -x_2 - x_1^2 + 2 \geq 0, \end{aligned} \tag{16}$$

for which we have plotted the constraints at $t = 0$ in Figure 1a. The feasible region is between the parabola and the horizontal line. Taking $t$ from 0 to 1 has the effect of moving the lower constraint up from $x_2 = -2$ to $x_2 = -1$. The objective gradient is $\nabla f(x) = (2x_1, -2x_2)$ and the Hessian of the objective is always indefinite $H = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$. The constraint gradients are $\nabla c(x) = \begin{pmatrix} 0 & 1 \\ -2x_1 & -1 \end{pmatrix}$. For $t \in [0, \quad 1]$ the primal solution is given by $x^*(t) = (0, t - 2)$ and the second constraint is inactive. The dual solution is $\lambda^*(t) = (-2x_2, 0)$. At $t = 0$ we have $x^* = (0, -2)$ and the optimal multiplier at $t = 0$ is $\lambda^* = (4, 0)$. The linearized constraints for this point are shown in Figure 1b.

We consider starting from an approximate solution at the point $\hat{x}(t = 0) = (0.1, -2)$, with dual variables $\hat{\lambda}(t = 0) = (4, 0)$, such that the first constraint is strongly active. Now consider taking the homotopy to $t = 1$.

The pure predictor QP has the form, recalling that we enforce linearized feasibility of the inactive upper constraint, and that we enforce the strongly active lower constraint as equality:

$$\begin{aligned} \min_{\Delta x} \quad & \Delta x_1^2 - \Delta x_2^2 \\ \text{s.t.} \quad & \Delta x_2 = 1, \\ & 5 - 2\Delta x_1 - \Delta x_2 \geq 0 \end{aligned} \tag{17}$$

This QP is convex with a unique solution $\Delta x = (0, 1)$ resulting in the subsequent point $\hat{x}(t = 1) = (0.1, -1)$.

The predictor corrector QP has the form

$$\begin{aligned} \min_{\Delta x} \quad & \Delta x_1^2 - \Delta x_2^2 + 0.2\Delta x_1 + 4\Delta x_2 \\ \text{s.t.} \quad & \Delta x_2 = 1, \\ & 5 - 2\Delta x_1 - \Delta x_2 \geq 0. \end{aligned} \tag{18}$$

Again the problem is convex with a unique primal solution $\Delta x = (-0.1, 1)$. This step has the effect of moving the iterate to the true optimal solution $\hat{x}(t = 1) = (0, -1) = x^*(t = 1)$.

Now consider a QP, which is the predictor-corrector QP, but without enforcing the strongly active constraints as equalities. This QP has been popular in the literature, and has also been applied in the real-time iterations scheme:

$$\begin{aligned} \min_{\Delta x} \quad & \Delta x_1^2 - \Delta x_2^2 + 0.2\Delta x_1 + 4\Delta x_2 \\ \text{s.t.} \quad & \Delta x_2 \geq 1, \\ & 5 - 2\Delta x_1 - \Delta x_2 \geq 0 \end{aligned} \tag{19}$$

This QP is nonconvex and unbounded; we can decrease the objective arbitrarily by setting $\Delta x = (2.5 - 0.5r, r)$ and letting a scalar $r \geq 1$ go to infinity. Although, there is a local minimizer at $\Delta x = (-0.1, 1)$, a QP solver that behaves "optimally" should find the unbounded "solution". This last approach cannot be expected to work reliably if the full Hessian of the optimization problem may become non-convex, which easily can be the case when optimizing economic objective functions. We note, however, that if the Hessian $\nabla_{xx}\mathcal{L}$ is positive definite, QP (19) will give the same solution as QP (18).

## 4. Numerical Case Study

### *4.1. Process Description*

We demonstrate the path-following NMPC (pf-NMPC) on an isothermal reactor and separator process depicted in Figure 2. The continuously stirred tank reactor (CSTR) is fed with a stream $F_0$ containing pure component $\mathcal{A}$, and a recycle $R$ from the distillation column. A first-order reaction $\mathcal{A} \rightarrow \mathcal{B}$ takes place in the CSTR where $\mathcal{B}$ is the desired product and the product with flow rate $F$ is fed to the column. In the distillation column, the unreacted raw material is separated from the product, and recycled into the reactor. The desired product $\mathcal{B}$ leaves the distillation column as bottom product, which is required to have a certain purity. Reaction kinetic parameters for the reactor are described in Table 1. The distillation column model is taken from [37]. Table 2 summarizes parameters used in the distillation. In total, the model has 84 state variables of which 82 are from the distillation (concentration and holdup for each stage) and 2 from the CSTR (one concentration and one holdup).

**Table 1.** Reaction kinetics parameters

| **Reaction** | **Reaction rate constant** $[min^{-1}]$ | **Activation energy [in** $J/mol$**]** |
|---|---|---|
| $\mathcal{A} \rightarrow \mathcal{B}$ | $1 \times 10^8$ | $6 \times 10^4$ |

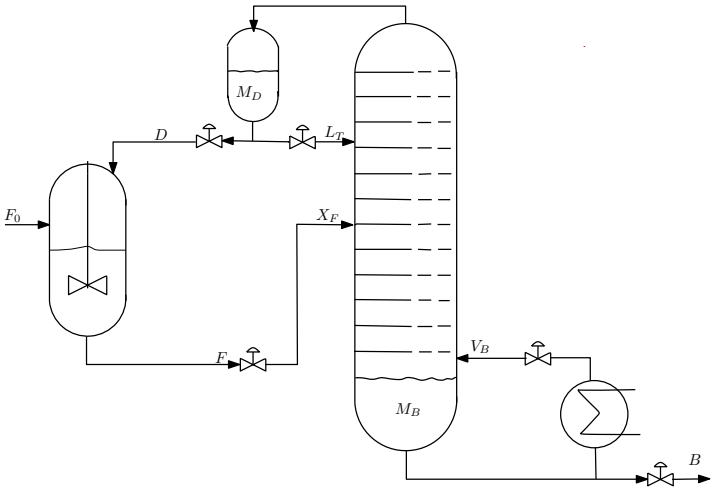


**Figure 2.** Diagram of CSTR and Distillation Column

**Table 2.** Distillation column A parameters

| Parameter | Value |
|---|---|
| $\alpha_{AB}$ | 1.5 |
| number of stages | 41 |
| feed stage location | 21 |

The stage cost of the economic objective function to optimize under operation is

$$J = p_F F_0 + p_V V_B - p_B B, \tag{20}$$

where $p_F$ is the feed cost, $p_V$ is steam cost, and $p_B$ is the product price. The price setting is $p_F = 1\,\$/kmol$, $p_V = 0.02\,\$/kmol$, $p_B = 2\,\$/kmol$. The operational constraints are the concentration of the bottom product ($x_B \leq 0.1$), as well as the liquid holdup at the bottom and top of the distillation column and in the CSTR ($0.3 \leq M_{\{B,D,CSTR\}} \leq 0.7$) 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

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

First, we run a steady-state optimization with the following feed rate $F_0 = 0.3$ [kmol/min].

This gives us the optimal values for control inputs and state variables. The optimal steady state input values are $\mathbf{u}_s = \begin{bmatrix} 1.18 & 1.92 & 1.03 & 0.74 & 0.29 \end{bmatrix}^T$. The optimal state and control inputs are used to construct regularization term added to the objective function (20). Now, the regularized stage becomes

$$J_m = p_F F_0 + p_V V_B - p_B B - p_D D + (\mathbf{z} - \mathbf{x}_s)^T \mathbf{Q}_1 (\mathbf{z} - \mathbf{x}_s) + (\mathbf{v} - \mathbf{u}_s)^T \mathbf{Q}_2 (\mathbf{v} - \mathbf{u}_s). \tag{21}$$

The weights $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are selected to make the rotated stage cost of the steady state problem strongly convex, for details see [22]. This is done to obtain an economic NMPC controller that is stable.

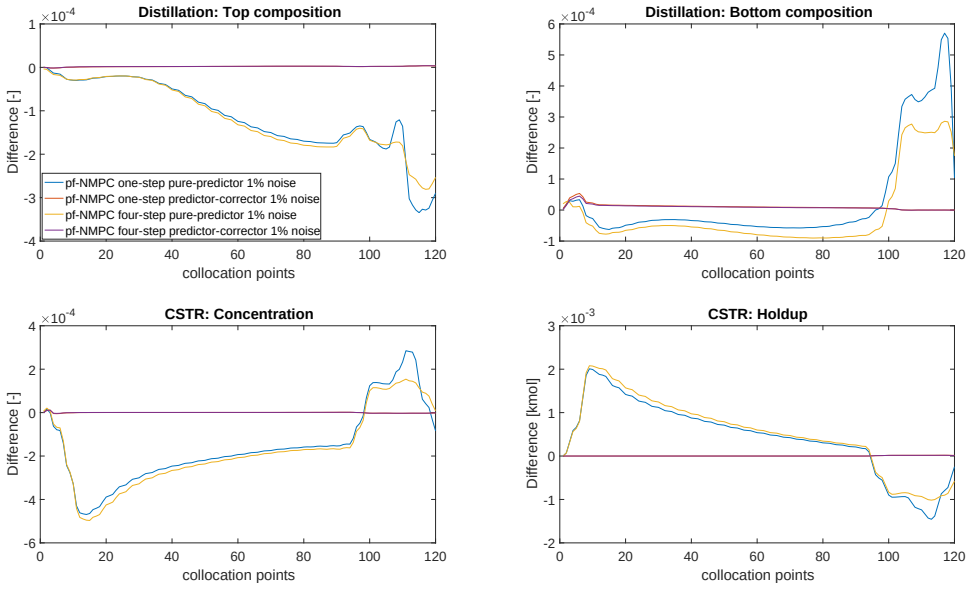


**Figure 3.** The difference in predicted states variables between iNMPC and pf-NMPC from second iteration.

Secondly, we set up the NLP for calculating the predicted state variables $\mathbf{z}$ and predicted control inputs $\mathbf{v}$. We employ a direct collocation approach on finite elements using Lagrange collocation to discretize the dynamics, where we use three collocation points in each finite element. By using the direct collocation approach, the state variables and control inputs become optimization variables.

The economic NMPC case study is initialized with the steady state values for a production rate $F_0 = 0.29$ kmol/min, such that the Economic NMPC controller is effectively controlling a throughput change from $F_0 = 0.29$ kmol/min to $F_0 = 0.3$ kmol/min. We simulate 150 MPC iterations, with a sample time of 1 minute. The prediction horizon of the NMPC controller is set to 30 minutes. This setting results in an NLP with 10314 optimization variables. We use CasADi [38] (version 3.1.0-rc1) with IPOPT [23] as NLP solver. For the QPs, we use MINOS QP [39] from TOMLAB.

### *4.2. Comparison of Open-loop Optimization Results*

In this section we compare the solutions obtained from the pathfollowing algorithm with the "true" solution of the optimization problem $\mathcal{P}_{nmpc}$ obtained by solving the full NLP. To do this, we consider the second MPC iteration, where the pathfollowing asNMPC is used for the first time to correct the one-sample ahead-prediction (in the first MPC iteration, to start up the asNMPC procedure, the full NLP is solved twice). We focus on the interesting case where the predicted state is corrupted by noise, such that the pathfollowing algorithm is required to update the solution. In Figure 3 we have plotted the difference between a selection of predicted states, obtained by applying the pathfollowing NMPC approaches, and the ideal NMPC approach. We observe that the one-step pure-predictor tracks the ideal NMPC solution worst, and the four-step pathfollowing with predictor-corrector tracks best. This happens because the predictor-corrector pathfollowing QP has an additional linear term in the objective function and constraint for the purpose of moving closer to the solution of the NLP (the "corrector" component), as well as tracing the first order estimate of the change in the solution (the "predictor"). The four-step pathfollowing performs better because a smaller step size gives finer approximation of the parametric NLP solution.

**Table 3.** Approximation error using pathfollowing algorithms

| Average approximation error between ideal NMPC and Pathfollowing asNMPC | |
|---|---|
| PF with predictor QP, 1 step | 4.516 |
| PF with predictor QP, 4 step | 4.517 |
| PF with predictor-corrector QP, 1 step | $1.333 \times 10^{-2}$ |
| PF with predictor-corrector QP, 4 step | $1.282 \times 10^{-2}$ |

This is also reflected in the average approximation errors given in 3. The average approximation error has been calculated by averaging the error 1-norm $\left|\left|\chi_{pathfollowing} - \chi_{ideal\,NMPC}\right|\right|_1$ over all MPC iterations.

We observe that in this case study, the accuracy of a single predictor-corrector step is almost as good as performing four predictor-corrector steps along the path. That is, a single predictor-corrector QP update may be sufficient for this application. However, in general, in presence of larger noise magnitudes and longer sampling intervals, which cause poorer predictions, a single-step update may no longer lead to good approximations. We note that the large error in the predictor-path-following method the solution accuracy several orders of magnitude.

On the other hand, given that the optimization vector $\chi$ has dimension 10164 for our case study, the average 1-norm approximation error of ca. 4.5 does result in very small errors on the individual variables.

### *4.3. Closed-loop Results – No Measurement Noise*

In this section we compare the results for closed loop process operation. We consider first the case without measurement noise, and we compare the results for ideal NMPC with the results obtained by the pathfollowing algorithm with the pure-predictor QP (10) and the predictor-corrector QP (12). Figure 4 shows the trajectories of the top and bottom composition in the distillation column, and the reactor concentration and holdup. Note that around 120 minutes the bottom composition constraint in the distillation column becomes active, while the CSTR holdup is kept at its upper bound all the time (any reduction in the holdup will result in an economic and product loss). In this case (without noise), the prediction and the true solution only differ due to numerical noise. There is no need to update the prediction, and all approaches give exactly the same closed-loop behavior. This is also reflected in the accumulated stage cost, which is shown in Table 4.

**Table 4.** Comparison of economic NMPC controllers. Accumulated stage cost is in $.

| Economic NMPC controller | | Acc. stage cost |
|---|---|---|
| iNMPC | | -296.42 |
| pure-predictor QP: | | |
| | pf-NMPC one step | -296.42 |
| | pf-NMPC four steps | -296.42 |
| predictor-corrector QP: | | |
| | pf-NMPC one step | -296.42 |
| | pf-NMPC four steps | -296.42 |

The closed-loop control inputs are given in Figure 5. Note here that the feed rate into the distillation column is adjusted such that the reactor holdup is at its constraint all the time.

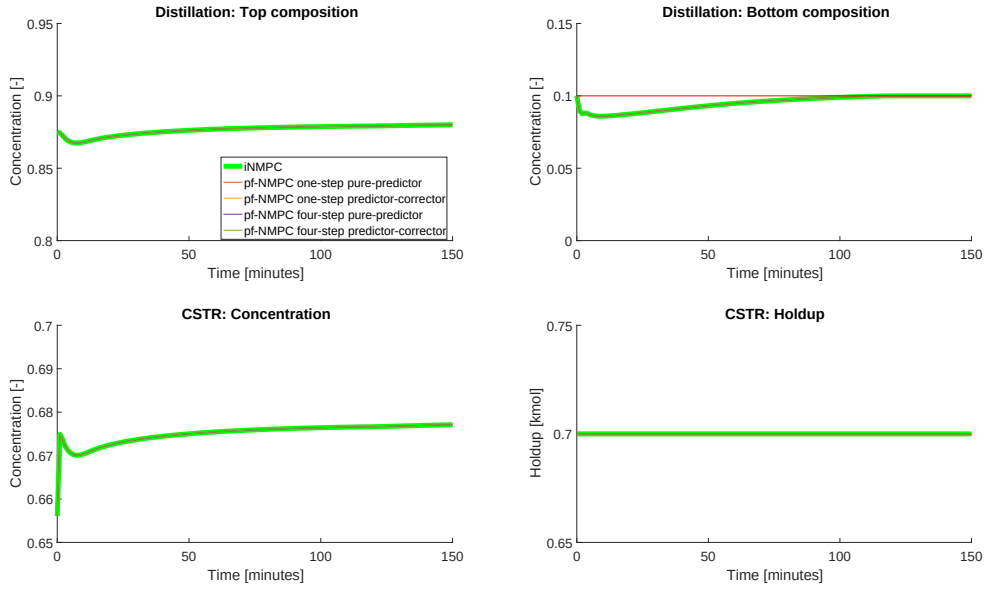


**Figure 4.** Recycle composition, bottom composition, reactor concentration, and reactor holdup.

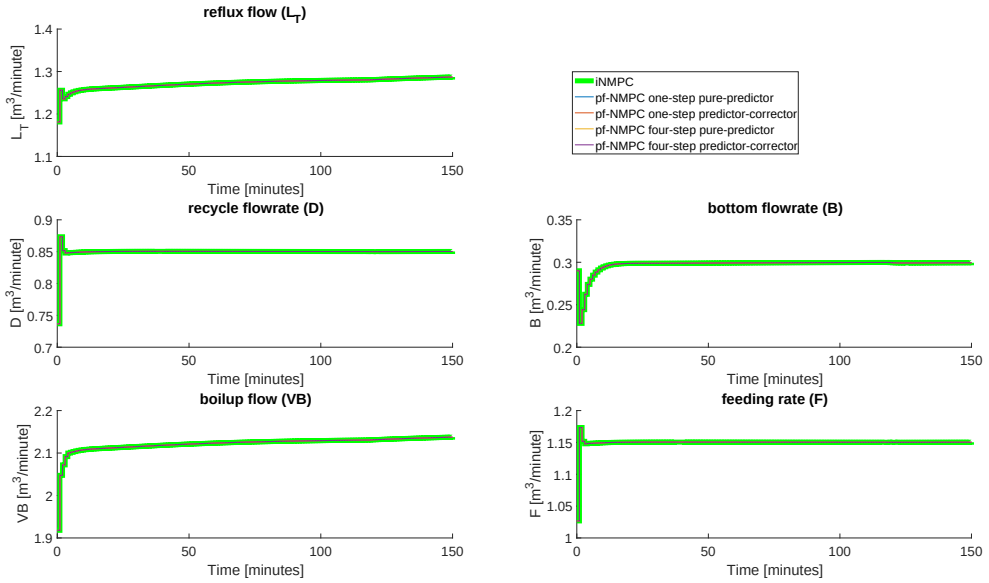


**Figure 5.** Optimized control inputs.

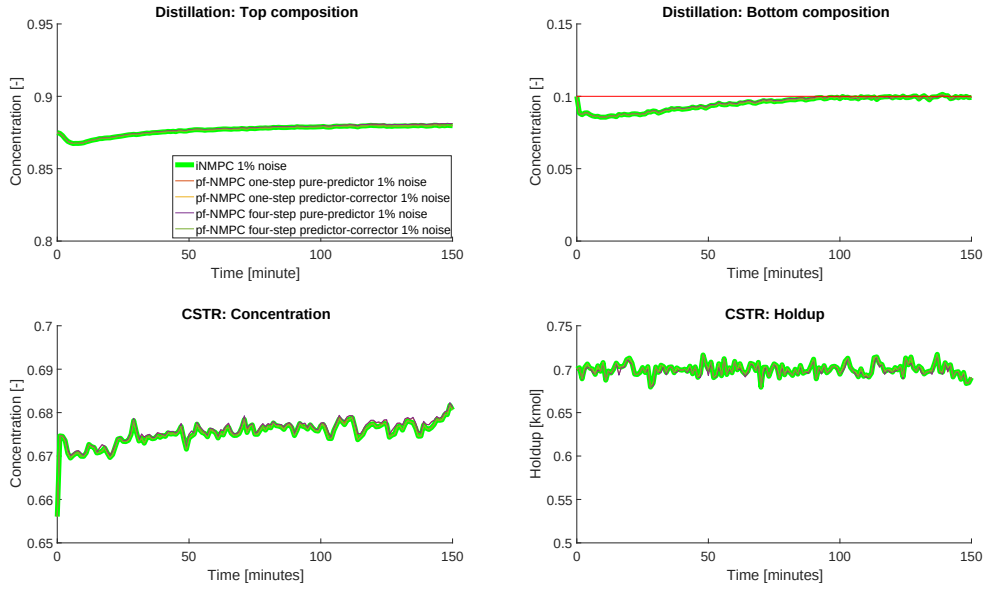


**Figure 6.** Recycle composition, bottom composition, reactor concentration, and reactor holdup.

**Table 5.** Comparison of economic NMPC controllers. Accumulated stage cost is in $.

| economic NMPC controller | | Acc. stage cost |
|---|---|---|
| iNMPC | | -296.82 |
| pure-predictor QP: | | |
| | pf-NMPC one step | -297.54 |
| | pf-NMPC four steps | -297.62 |
| predictor-corrector QP: | | |
| | pf-NMPC one step | -296.82 |
| | pf-NMPC four steps | -296.82 |

### *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 steady state values. This will result in corrupted predictions that have to be corrected for by the pathfollowing algorithms. Again, we perform simulations with one and four steps of pure-predictor and predictor-corrector QPs.

Figure 6 shows the top and bottom compositions of the distillation column, together with the concentration and holdup in the CSTR. The states are obtained under closed-loop operation with the ideal and pathfollowing NMPC algorithms. Due to noise it is not possible to avoid violation of the active constraints in the holdup of the CSTR and the bottom composition in the distillation column. This is the case for both the ideal NMPC and the pathfollowing approaches.

The input variables shown in Figure 7 are also reflecting the measurement noise, and again we see that the fast sensitivity NMPC approaches are very close to the ideal NMPC inputs.

Finally, we compare the accumulated economic stage cost in Table 5. Here we observe that our proposed predictor-corrector pathfollowing algorithm performs identically to the ideal NMPC. This is as expected, since the predictor-corrector pathfollowing algorithm is trying to reproduce the true NLP solution. Interestingly, in this case, the larger error in the pure predictor pathfollowing NMPC

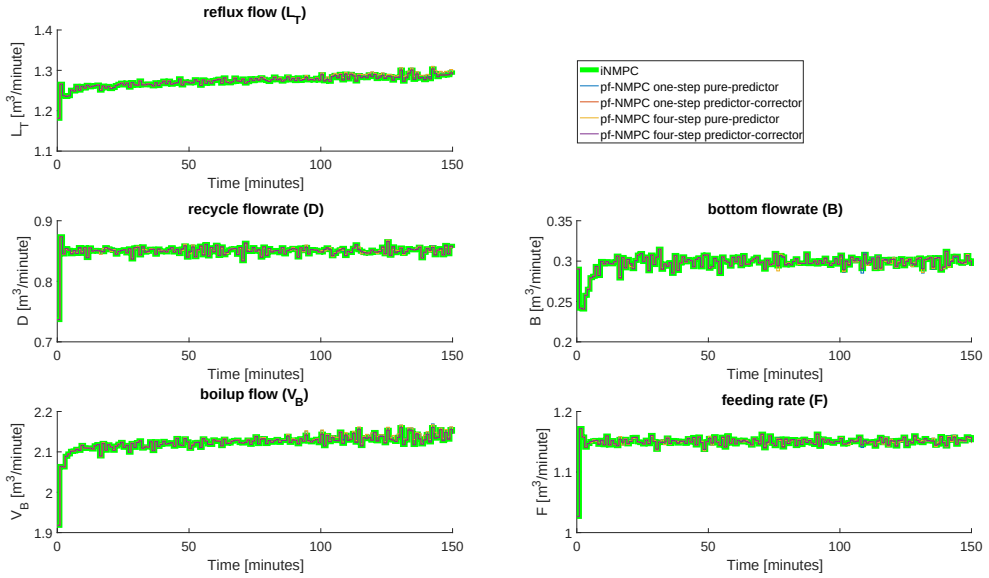


**Figure 7.** Optimized control inputs.

leads to a better economic performance of the closed loop system. This behavior is due to the fact that the random measurement noise can have positive and negative effect on the operation, which is not taken into account by the ideal NMPC (and also the predictor-corrector NMPC). In this case, the inaccuracy of the pure predictor pathfollowing NMPC led to better economic performance in closed loop. But it could also have been the opposite.

## 5. Discussion and Conclusion

We applied the pathfollowing ideas developed in Jäschke et al. [22] and Kungurtsev and Diehl [30] to a large-scale process containing a reactor, a distillation column and a recycle stream. Compared with single-step updates based on solving a linear system of equations as proposed by [14] our pathfollowing approach requires somewhat more computational effort. However, the advantage of the pathfollowing approach is that active set changes are handled rigorously. Moreover, solving a sequence of a few QPs can be expected to be much faster than solving the full NLP, especially since they can be initialized very well, such that the computational delay between obtaining the new state and injecting the updated input into the plant is still sufficiently small. In our computations, we have considered a fixed step-size for the pathfollowing, such that the number of QPs to be solved is known in advance.

The case without noise does not require the pathfollowing algorithm to correct the solution, because the prediction and the true measurement are identical, except for numerical noise. However, when measurement noise is added to the holdups, the situation becomes different. In this case, the prediction and the measurements differ, such that an update is required. All four approaches track the ideal NMPC solution to some degree, however, in terms of accuracy the predictor-corrector performs consistently better. Given that the pure sensitivity QP and the predictor-corrector QP are very similar in structure, it is recommended to use the latter in the pathfollowing algorithm, especially for highly nonlinear processes and cases with significant measurement noise.

We have presented basic algorithms for pathfollowing, and they seem to work well for the cases we have studied, such that the pathfollowing algorithms do not diverge from the true solution. In principle, however, the path-following algorithms may get lost, and more sophisticated implementations need to include checks and safeguards. We note, however, that the application of

the pathfollowing algorithm in the Advanced-Step NMPC framework has the desirable property that the solution of the full NLP acts as a corrector, such that if the pathfollowing algorithm diverges from the true solution, this will be most likely for only one sample time, until the next full NLP is solved.

The pathfollowing algorithm in this paper (and the corresponding QPs) still relies on the assumption of linearly independent constraint gradients. If there are path-constraints present in the discretized NLP, care must be taken to formulate them in such a way that LICQ is not violated. In future work we will consider extending the pathfollowing NMPC approaches to handle more general situations with linearly dependent inequality constraints.

## References


1. Zanin, A. C., Tvrzská de Gouvêa, M., and Odloak, D., Industrial implementation of a real-time optimization strategy for maximizing production of LPG in a FCC unit. Comput. Chem. Eng. 24, 525-531, 2000.
2. Zanin, A. C., Tvrzská de Gouvêa, M., and Odloak, D., Integrating real-time optimization into the model predictive controller of the FCC system. Control Eng. Practice 10, 819-831, 2002.
3. Rawlings, J. B., and Amrit, R, Optimizing process economic performance using model predictive control. A chapter in *Nonlinear Model Predictive Control* book. Volume 384 of the series Lecture Notes in Control and Information Sciences pp 119-138, Springer, 2009.
4. Rawlings, J. B, Angeli, D, and Bates, C. N., Fundamentals of economic model predictive control. In proceeding of *The 51st IEEE Conference on Conference on Decision and Control (CDC)*, Maui, Hawaii, December 10-13, 2012.
5. Ellis, M., Durand, H., and Christofides, P. D., A tutorial review of economic model predictive methods. Journal of Process Control 24, 1156-1178, 2014.
6. Tran, T., Ling, K-V., and Maciejowski, J. M., Economic model predictive control - a review. In proceeding of *The 31st ISARC*, Sydney, Australia, July 9-11, 2014.
7. Angeli, D., Amrit, R., and Rawlings, J. B, On average performance and stability of economic model predictive control. IEEE Transactions on Automatic Control Vol 57(7), 1615-1626, 2012.
8. Idris, E. A. N., and Engell, S., Economics-based NMPC strategies for the operation and control of a continuous catalytic distillation process. Journal of Process Control, Volume 22, Issue 10, Pages 1832–1843, 2012.
9. Findeisen, R., and Allgöwer, F., Computational delay in nonlinear model predictive control. In proceeding of *The International Symposium on Advanced Control of Chemical Proceses, ADCHEM'03*, Hongkong, 2004.
10. Biegler, L. T., Yang, X., and Fischer, G. A. G., Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization. Journal of Process Control, Volume 30, Pages 104–116, 2015.
11. Diehl, M., Bock, H. G, Schlöder, J. P, Findeisen, R, Nagy, Z, and Allgöwer, F, Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. Journal of Process Control, volume 12 (4), page 577-585, 2002.
12. Bock, H. G., Diehl, M, Kostina, E, and Schlöder, J. P, Constrained Optimal Feedback Control of Systems Governed by Large Differential Algebraic, In book of *Real-Time PDE-constrained optimization*, SIAM publisher, 2007.
13. Diehl, M, Findeisen, R, and Allgöwer, F, A stabilizing real-time implementation of nonlinear model predictive control, In book of *Real-Time PDE-constrained optimization*, SIAM publisher, 2007.
14. Zavala, V. M, and Biegler, L. T, The advanced-step NMPC controller: optimality, stability, and robustness, Automatica, Vol. 45 (1), Pages 86–93, 2009.
15. Diehl, M, Bock, H. G., and Schlöder, J. P., A real-time iteration scheme for nonlinear optimization in optimal feedback control, SIAM Journal on control and optimization, Volume 43, Issue 5, pp 1714-1736, 2005.
16. Pirnay, H, López-Negrete, R, and Biegler, L. T, Optimal sensitivity based on IPOPT, Mathematical Programming Computation, Volume 4, Issue 4, pp 307-331, 2002.
17. Ohtsuka, T, A continuation/GMRES method for fast computation of nonlinear receding horizon control, Automatica, Volume 40, Issue 4, Pages 563–574, 2004.
18. Büskens, C, and Maurer, H., SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control, Journal of Computational and Applied Mathematics, Volume 120, Pages 85-108, 2000.

19. Würth, L, Hannemann, R, and Marquardt, W, Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization, Journal of Process Control, Volume 19, Issue 8, Pages 1277–1288, 2009.
20. Würth, L, Hannemann, R, and Marquardt, W, A two-layer architecture for economically optimal process control and operation, Journal of Process Control, Volume 21, Issue 3, Pages 311–321, 2011.
21. Wolf, I. J, and Marquadt, W, Fast NMPC schemes for regulatory and economic NMPC - a review, Journal of Process Control, Volume 44, page 162-183, 2016.
22. Jäschke, J, Yang, X, and Biegler, L. T, Fast economic model predictive control based on NLP-sensitivities, Journal of Process Control, Volume 24, page 1260-1272, 2014.
23. Wächter, A, and Biegler, L. T, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming, Volume 106, Issue 1, page 25-57, March 2006.
24. Fiacco, A. V, Introduction to sensitivity and stability analysis in nonlinear programming, Academic press, New York, 1983.
25. Li, W.C., Biegler, L.T. Multistep, Newton-type control strategies for constrained nonlinear processes. Chem. Eng. Res. Des. 67, 562–577 (1989)
26. Gros, S and Quirynen, R. and Diehl, M. An improved real-time economic NMPC scheme for Wind Turbine control using spline-interpolated aerodynamic coefficients. 53rd IEEE Conference on Decision and Control, Los Angeles, CA, 2014, pp. 935-940. doi: 10.1109/CDC.2014.7039501
27. Gros, S and Vukov, M and Diehl, M. "A real-time MHE and NMPC scheme for wind turbine control," 52nd IEEE Conference on Decision and Control, Firenze, 2013, pp. 1007-1012. doi: 10.1109/CDC.2013.6760014
28. Yang, X, and Biegler, L. T, Advanced-multi-step nonlinear model predictive control, Journal of Process Control, Volume 23, Pages 1116-1128, 2013.
29. Kojima, M, Strongly stable stationary solutions in nonlinear programming, in: S.M. Robinson (Ed.), Analysis and Computation of Fixed Points, Academic Press, New York, 1980.
30. Kungurtsev, V, and Diehl, M, Sequential quadratic programming methods for parametric nonlinear optimization, Computational Optimization and Applications, Volume 59, Issue 3, pp 475–509, December 2014.
31. Rawlings, J. B, Mayne, D. Q, Model predictive control: theory and design, Nob Hill Publishing, Madison, Wisconsin, 2009.
32. Diehl, M., Amrit, R., and Rawlings, J. B, A lyapunov function for economic optimizing model predictive control. IEEE Transaction of Automatic Control, Vol. 56, No. 3, March 2011. pp. 703-707.
33. Amrit, R., Rawlings, J. B, and Angeli, D, Economic optimization using model predictive control with a terminal cost. Annual Reviews in Control 35, 2011. pp. 178-186.
34. Magni, L, and Scattolini, R, Robustness and robust design of MPC for nonlinear systems. In Assessment and future directions of NMPC, Berlin, Springer-Verlag. pp. 239-254.
35. Jiang, Z. P, and Wang, Y, Input-to-state stability for discrete-time nonlinear systems. Automatica, 37(6). pp. 857-869.
36. Müller, M. A, Angeli, D, and Allgöwer, F, On necessity and robustness of dissipativity in economic model predictive control, IEEE Transaction of Automatic Control, Vol. 60, No. 6, June 2015. pp. 1671-1676.
37. Skogestad, S, and Postlethwaite, I, Multivariate feedback control: analysis and design. Wiley-Interscience, 2005. ISBN: 047011688.
38. Andersson, J, A general purpose software framework for dynamic optimization. PhD thesis, Arenberg Doctoral School, KU Leuven, October 2013.
39. Murtagh, B.A., and Saunders, M. A., A projected lagrangian algorithm and its implementation for sparse nonlinear constraints. Mathematical Programming Study Vol. 16, 1982, pp. 84-117.
40. Bonnans, J. F., and Shapiro, A. Perturbation analysis of optimization problems. Springer Science & Business Media, 2013.
41. Dontchev, A. L., Krastanov, M. I, Rockafellar, R. T, and Veliov, V. M. "An Euler–Newton Continuation Method for Tracking Solution Trajectories of Parametric Variational Inequalities." SIAM Journal on Control and Optimization 51.3 (2013): 1823-1840.
42. Bonnans, J. F., and Shapiro, A. "Optimization problems with perturbations: A guided tour." SIAM review 40.2 (1998): 228-264.

503 43. Levy, A. B. "Solution sensitivity from general principles." SIAM journal on control and optimization 40.1
504 (2001): 1-38.
505 44. Kadam, J. and Marquardt, W. Sensitivity-based solution updates in closed-loop dynamic optimization, in:
506 Proceedings of the DYCOPS 7 Conference, Cambridge, USA, 2004.