A Predictor-Corrector Path-Following Method for Fast Dual-Degenerate Economic Model Predictive Control

Eka Suwartadi * **Johannes Jäschke** *

* *Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway (e-mail: eka.suwartadi@ntnu.no and jaschke@ntnu.no).*

Abstract: We present a sensitivity-based nonlinear model predictive control (NMPC) in which dual-degeneracy may arise in dynamic optimization problem. We consider an optimization algorithm under a weak constraint qualification, i.e., Mangasarian-Fromovitz constraint qualification (MFCQ). We solve an economic model predictive control problem by applying a path-following method within the advance-step NMPC (asNMPC) framework. The path-following method comprises of corrector and predictor steps, which are manifested as a system of linear equations as well as a quadratic programming problem, and a multiplier jump step in the form of a linear programming problem. In order to assess our proposed method, we implement an economic NMPC controller for an isothermal reactor process.

Keywords: Optimal control, economic model predictive control, parametric optimization.

1. INTRODUCTION

Optimizing economic performance is a natural choice of objective in process control industries. A common practice is to split the control system into two layers, i.e., real-time optimization (RTO) and a lower level model predictive control (MPC) layer. Steady-state optimization using an economic objective function in the RTO layer yields optimal setpoint values, which are passed to the lower MPC layer. A dynamics optimization is then performed in the lower MPC layer in which a tracking cost function, that is the difference the state values and their setpoints, is minimized. Because the steady-state optimization is performed on a slow time scale, the two-layer control system may not be able to optimally handle fast changes that happen in the range of minutes or seconds. To mitigate this, it was proposed to combine economics and control into a single layer. This gives rise to the economic MPC, which is reviewed in Faulwasser et al. (2018).

In this work we present a computational method for obtaining a fast solution to the economic model predictive control (EMPC) problem. As the process models become more complex, the online optimization problem cannot be solved sufficiently fast, which may result in instability of the closed loop system. To reduce the delay between obtaining a new measurement and implementing the input to the plant, *real-time iteration* (RTI) scheme (Diehl et al. (2002)) and the *advanced-step NMPC* (Zavala and Biegler (2009)) were proposed. These methods are based on the concept of parametric nonlinear programming (NLP) (Guddat et al. (1990)) (also known as NLP-sensitivity), where the initial state values are considered as a parameter in the optimization problem.

Most current methods for sensitivity based NMPC assume strong regularity conditions typically linear independent constraint qualification (LICQ), which result in unique multipliers. An exception is the paper by Jäschke et al. (2014), who developed a path-following NMPC procedure that can handle non unique multipliers, a case that occurs when Mangasarian-Fromovitz constraint qualification (MFCQ) holds. The contribution of this work is to present an improvement version of the path-following NMPC presented in Jäschke et al. (2014). In particular, we employ a dual-degenerate parametric NLP algorithm proposed in Kungurtsev and Jäschke (2017) to obtain approximate solutions to the dynamic optimization problem within the framework of advanced-step NMPC (asNMPC). For each step along the path, the method performs the following steps. Firstly, a system of linear equations is solved as a Newton corrector step. This step refines the current primal and dual variables. Secondly, a quadratic programming (QP) problem is solved to find a directional derivative, that is used as a predictor step. Finally, a linear program (LP) is solved to allow for jumps in the dual variables.

This paper is organized as follows. We begin by formulating NMPC problem in Section 2 and present the ideal solution strategy. Then the predictor-corrector path-following method for dual-degenerate case is explained in Section 3 as well as the use of the method in path-following NMPC (pf-NMPC) approach. We demonstrate our proposed method in a case example in Section 4. Finally, we conclude with discussion and remarks in Section 5.

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

2. ECONOMIC NMPC

2.1 Dynamic Optimization

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

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

where $\mathbf{z}_l \in \mathbb{R}^{n_z}$ and $\mathbf{v}_l \in \mathbb{R}^{n_v}$ are internal variables known as predicted state and control at sample time l, respectively. The objective function consists of the terminal cost $\Psi(\mathbf{z}_N) \in \mathcal{C}^1 : \mathbb{R}^{n_z} \to \mathbb{R}$ and the stage costs $\psi(\mathbf{z}_l,\mathbf{v}_l) \in \mathcal{C}^1 : \mathbb{R}^{n_z} \times \mathbb{R}^{n_v} \to \mathbb{R}$. The constraints include a discrete time dynamical system $f \in \mathcal{C}^1 : \mathbb{R}^{n_z} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_z}$, the equality constraint for initial condition $\mathbf{z}_0$, 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 of terminal constraint $\mathcal{X}_f$. The set $\mathcal{Z}$ denotes the path constraint limiting the predicted state and control. Since the set $\mathcal{Z}$ contains both feasible state and control, for ease of exposition, we split the set onto $\mathcal{X} \subset \mathbb{R}^{n_z}$ and $\mathcal{U} \subset \mathbb{R}^{n_v}$ denoting the feasible state and control sets, respectively.

Having obtain solution of the optimization problem $\mathcal{P}_N$, the first move of the optimized predicted control input $\mathbf{v}_l^* := \kappa(\mathbf{x}_k)$ is applied the plant which then evolves such that

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \kappa(\mathbf{x}_k)), \qquad (2)$$

where $\mathbf{x}_k$ is the actual state variable given from measurement of the plant. When the measurement is corrupted by noise, the closed loop dynamics become

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \kappa(\mathbf{x}_k)) + \mathbf{w}_k, \qquad (3)$$

here $\mathbf{w}_k \in \mathbb{R}^{n_z}$ represents the measurement noise.

As the time instant k evolves, the optimization problem $\mathcal{P}_N$ is solved in receding 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 $\mathbf{v}_l^*$,
(4) Set $k \leftarrow k + 1$.

We refer to the procedure above as an *ideal NMPC* (*iNMPC*) controller.

2.2 The Advanced-step NMPC

The optimization problem (1) remains almost the same from one MPC iteration to another, the only difference is the actual state variable values, which are obtained from measurement data $\mathbf{x}_k$. Hence, the initial state variable values are considered as a parameter. In order to reduce the computational time for solving the optimization problem $\mathcal{P}_N$ (1), instead of solving a full NLP problem, the asNMPC (Zavala and Biegler (2009)) computes the sensitivity of NLP solution with respect to the initial state variable values $\mathbf{z}_0$.

The change in the measurement data is considered as a perturbation to the optimal NLP solution. If LICQ and strict complementarity hold together with a suitable second order condition, the sensitivity can be calculated by solving a system of linear equations that can be formulated using the Karush-Kuhn-Tucker (KKT) system of the NLP and the solution is updated using the NLP sensitivity. While waiting for new measurement data, the asNMPC scheme performs an offline step where the full NLP problem is solved. In summary, the asNMPC procedure includes the following three steps (Zavala and Biegler (2009)).

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

The asNMPC faces a challenge when an active-set change occurs. Some heuristics (Biegler et al. (2015)), such as "clipping in the first interval", have been suggested to overcome the challenge.

2.3 Enforcing convergence for economic NMPC

Since the stage cost in the optimization problem $\mathcal{P}_N$ (1) can be any arbitrary economic measure, it may be difficult to ensure that the closed loop system is stable. This can be done by solving regularizing the NMPC problem such that the process approaches its steady-state optimal point, which can be found by the following steady-state optimization problem

$$\begin{aligned}\min_{(\mathbf{x},\mathbf{u})\in\mathcal{Z}} \quad \psi(\mathbf{x},\mathbf{u}) \qquad (4)\\ \text{s.t.}\ \mathbf{x} - f(\mathbf{x},\mathbf{u}) = 0.\end{aligned}$$

We modify the stage cost by incorporating a regularization term to the economic stage cost so that it becomes

$$\psi_m(\mathbf{z},\mathbf{v}) := \psi(\mathbf{z},\mathbf{v}) + \alpha_{state}(\| \mathbf{z} - \mathbf{x}_s \|) + \alpha_{input}(\| \mathbf{v} - \mathbf{u}_s \|). \qquad (5)$$

Here $\mathbf{x}_s$ and $\mathbf{u}_s$ are the steady-state optimal values obtained from (4). The value of the weights α_{state} and α_{input} may be chosen by Gershgorin bound criteria, see Jäschke et al. (2014), aiming to make the stage cost convex. It should be noted that convexity implies dissipativity which guarantees the resulted closed-loop system is asymptotically stable (Diehl et al. (2011)).

3. PREDICTOR-CORRECTOR PATH-FOLLOWING ECONOMIC NMPC

We explain a predictor-corrector path-following method in this section along with its use in an economic NMPC controller. We begin by giving required definitions for building up the path-following algorithm.

3.1 Preliminaries

We consider the optimization problem (1) as a general parametric nonlinear optimization problem of the form

$$\begin{aligned} \min_{\boldsymbol{\chi}} \quad & F(\boldsymbol{\chi}, \mathbf{p}) \\ \text{subject to}\; & c_i(\boldsymbol{\chi}, \mathbf{p}) = 0, i \in \mathcal{E}, \\ & c_i(\boldsymbol{\chi}, \mathbf{p}) \leq 0, i \in \mathcal{I}, \end{aligned} \tag{6}$$

where $F : \mathbb{R}^{n_\chi} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}$ is the objective function, $\boldsymbol{\chi} \in \mathbb{R}^{n_\chi}$ is the primal variable and $\mathbf{p} \in \mathbb{R}^{n_p}$ is the parameter. The equality and inequality constraint sets are denoted by $\mathcal{E} = \{1, \ldots, m\}$ and $\mathcal{I} = \{m+1, \ldots, n\}$, respectively.

The Lagrangian is defined as

$$L(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p}) := F(\boldsymbol{\chi}, \mathbf{p}) + \mathbf{y}^T c(\boldsymbol{\chi}, \mathbf{p}), \tag{7}$$

where $\mathbf{y}$ is the dual variable. The Karush-Kuhn-Tucker (KKT) conditions for the problem are

$$\begin{aligned} \nabla_{\boldsymbol{\chi}} L(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p}) &= 0, \\ c_i(\boldsymbol{\chi}, \mathbf{p}) &= 0, i \in \mathcal{E}, \\ c_i(\boldsymbol{\chi}, \mathbf{p}) &\leq 0, i \in \mathcal{I}, \\ \mathbf{y}^T c(\boldsymbol{\chi}, \mathbf{p}) &= 0, \\ \mathbf{y}_i &\geq 0, i \in \mathcal{I}. \end{aligned} \tag{8}$$

We denote active inequality constraints as a set $\mathcal{A}(\boldsymbol{\chi}, \mathbf{p}) = \{c_i(\boldsymbol{\chi}, \mathbf{p}) = 0, i \in \mathcal{I}\}$. For a given multiplier $\mathbf{y}$ that satisfies (8) the active inequality set $\mathcal{A}(\boldsymbol{\chi}, \mathbf{p})$ has two subsets, which are a weakly active set $\mathcal{A}_0(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p}) = \{i \in \mathcal{A}(\boldsymbol{\chi}, \mathbf{p}) \mid \mathbf{y}_i = 0\}$ and a strongly active set $\mathcal{A}_+(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p}) = \{i \in \mathcal{A}(\boldsymbol{\chi}, \mathbf{p}) \mid \mathbf{y}_i > 0\}$.

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

$$H(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p}) = \nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} F(\boldsymbol{\chi}, \mathbf{p}) + \sum_{i=1}^{n} \left(\nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, \mathbf{p})\right) \mathbf{y}_i. \tag{9}$$

Definition 1. Strong second-order sufficient conditions (ssosc) holds at $(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p})$ if a pair of primal-dual variable $(\boldsymbol{\chi}, \mathbf{y})$ satisfies the first-order necessary conditions (8) at $\mathbf{p}$ and

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

where the set $\mathcal{C}(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p})$ is defined as

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

Definition 2. General Strong Second-order Sufficient Optimality Conditions (GSSOSC) is satisfied if the SSOSC is satisfied for all $\mathbf{y}$ that fulfill the first-order necessary conditions (8).

We require constraint qualification for the problem (6) to ensure that the KKT conditions (8) are a necessary condition for optimality.

Definition 3. Given a point $(\boldsymbol{\chi}, \mathbf{p})$, the linear independence constraint qualification (LICQ) holds at $(\boldsymbol{\chi}, \mathbf{p})$ if the set of vectors $\{\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, \mathbf{p}), i \in \mathcal{E} \cup \mathcal{A}(\boldsymbol{\chi}, \mathbf{p})\}$ are linearly independent.

Definition 4. The Mangasarian-Fromovitz Constraint Qualification (MFCQ) holds at $(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p})$ 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 step $\mathbf{s}$ 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 LICQ implies the uniqueness of dual variables and the MFCQ implies that dual variables are bounded.

Definition 5. The Constant Rank Constraint Qualification (CRCQ) holds at $(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p})$ for a feasible point $\boldsymbol{\chi}$ if there exists a neighborhood $\mathcal{N}$ of $\boldsymbol{\chi}$ such that for all subsets $\mathcal{U} \subseteq \mathcal{E} \cup \mathcal{A}(\boldsymbol{\chi}, \mathbf{p})$, the rank of $\{\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, \mathbf{p}), i \in \mathcal{U}\}$ is equal to the rank of $\{\nabla_{\boldsymbol{\chi}} c_i(\bar{\boldsymbol{\chi}}, \mathbf{p}), i \in \mathcal{U}\}$ for all $\bar{\boldsymbol{\chi}} \in \mathcal{N}$.

Finally, we define the optimality residual as

$$\eta(\boldsymbol{\chi}, \mathbf{y}, \mathbf{p}) = \left\| \begin{pmatrix} \nabla_{\boldsymbol{\chi}} F(\boldsymbol{\chi}, \mathbf{p}) + \nabla_{\boldsymbol{\chi}} c(\boldsymbol{\chi}, \mathbf{p}) \mathbf{y} \\ c(\boldsymbol{\chi}, \mathbf{p})_{\mathcal{E}} \\ [\min(c(\boldsymbol{\chi}, \mathbf{p}), \mathbf{y})]_{\mathcal{I}} \end{pmatrix} \right\|_\infty. \tag{10}$$

3.2 Predictor-Corrector Path-Following

The path-following method described in Kungurtsev and Jäschke (2017) consists of three steps: Corrector Step, Predictor Step, and Multiplier Jump Step. These three steps are run repeatedly to follow the path of optimal solutions, starting from initial parameter value $\mathbf{p}_0$ until final parameter $\mathbf{p}_f$. The parameter $\mathbf{p}$ is updated according to $\mathbf{p}(t_k) = (1 - t_k)\mathbf{p}_0 + t_k \mathbf{p}_f$, where $t_0 = 0$ until it reaches $t_k = 1$, that is $t_0 = 0 < t_1 < t_2 \ldots < t_k = 1$. We denote the primal and dual variables during the course of path-following iteration as $\boldsymbol{\chi}_k$ and $\mathbf{y}_k$ respectively, where k represents the index of the iteration.

Now, we are ready to explain each of the steps.

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 $\mathbf{p}$. We solve the system of a linear equations

$$\boldsymbol{A} \begin{pmatrix} \Delta_c \boldsymbol{\chi} \\ \Delta_+ \mathbf{y} \end{pmatrix} = -\boldsymbol{B}, \tag{11}$$

where $\boldsymbol{A} = \begin{pmatrix} H(\boldsymbol{\chi}_k, \mathbf{y}_k, t) & \nabla_{\boldsymbol{\chi}} c_{\mathcal{A}_{+,k}}(\boldsymbol{\chi}_k, t) \\ \nabla_{\boldsymbol{\chi}} c_{\mathcal{A}_{+,k}}(\boldsymbol{\chi}_k, t)^T & 0 \end{pmatrix}$,

and $\boldsymbol{B} = \begin{pmatrix} \nabla_{\boldsymbol{\chi}} F(\boldsymbol{\chi}_k, t) + \nabla_{\boldsymbol{\chi}} c(\boldsymbol{\chi}_k, t)\mathbf{y}_k \\ \nabla_{\boldsymbol{\chi}} c_{\mathcal{A}_{+,k}}(\boldsymbol{\chi}_k, t) \end{pmatrix}$.

Since LICQ may not hold, where the Jacobian $\nabla_{\boldsymbol{\chi}} c_{\mathcal{A}_{+,k}}(\boldsymbol{\chi}_k, t)$ is not always full rank, the initial dual variables $\mathbf{y}_k$ are chosen from vertex of a polytope region. This can be done by solving a linear program. Otherwise, the matrix $\boldsymbol{A}$ may become singular.

The approximate dual variables for the strongly active constraint are obtained from the solution, i.e., $[\Delta_c \mathbf{y}]_{\mathcal{A}_{+,k}} = \Delta_+ \mathbf{y}$ and the remaining constraints are set to zero, namely, $[\Delta_c \mathbf{y}]_{\{1,\ldots,n\} \setminus \mathcal{A}_{+,k}} = 0$.

2.Predictor Step. We solve the following predictor QP

$$\begin{aligned} \min_{\Delta_p \boldsymbol{\chi}} \; & \frac{1}{2} \Delta_p \boldsymbol{\chi}^T H(\boldsymbol{\chi}_k, \mathbf{y}_k, t + \Delta t) \Delta_p \boldsymbol{\chi} \\ \text{subject to} \quad & \beta + \alpha^T \Delta_p \chi = 0, \, i \in \mathcal{A}_{+,k} \\ & \beta + \alpha^T \Delta_p \chi \leq 0, \, i \in \mathcal{A}_k \backslash \mathcal{A}_{+,k} \end{aligned} \tag{12}$$

where

$$\beta = \nabla_t c_i\left(\boldsymbol{\chi}_k, t\right)\Delta t,$$

$$\alpha = \left(\nabla_{\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_k, t+\Delta t\right) + \nabla^2_{\boldsymbol{\chi}\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_k, t+\Delta t\right)\Delta_c \boldsymbol{\chi}\right).$$

Note that since the parameter $\mathbf{p}$ enters linearly to the problem, the derivatives with respect to the parameter $\mathbf{p}$ are zero. We obtain the primal and dual solution in this step $(\Delta_p \boldsymbol{\chi}, \Delta_p \mathbf{y})$. Combining solution from the corrector step, we get $(\Delta\boldsymbol{\chi}, \Delta\mathbf{y}) = (\Delta_c\boldsymbol{\chi} + \Delta_p\boldsymbol{\chi}, \Delta_c\mathbf{y} + \Delta_p\mathbf{y})$. Here, we update the primal and dual variables solutions, i.e., $\boldsymbol{\chi}_{k+1} = \boldsymbol{\chi}_k + \Delta\boldsymbol{\chi}$, $\mathbf{y}_{k+1} = \mathbf{y}_k + \Delta\mathbf{y}$, and consequently the strongly active set $\mathcal{A}_{+,k+1}$.

3.Multiplier Jump Step. Under the assumptions MFCQ and CRCQ, there may be a discontinuity in the dual variable $\mathbf{y}$. In order to allow this discontinuity, we compute the dual variable solutions by solving the following LP,

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

where

$$\Omega = \nabla_{\boldsymbol{\chi}} L(\boldsymbol{\chi}_k + \Delta\boldsymbol{\chi}, \mathbf{y}_k + \Delta\mathbf{y}, t+\Delta t),$$

$$\vartheta = \nabla_{\boldsymbol{\chi}} F(\boldsymbol{\chi}_k + \Delta\boldsymbol{\chi}, t+\Delta t) + \sum_{i\in\mathcal{A}_{k+1}} \nabla_{\boldsymbol{\chi}} c_i\left(\boldsymbol{\chi}_k + \Delta\boldsymbol{\chi}, t+\Delta t\right)\mathbf{y}_i.$$

The solution $(\mathbf{y}_{LP})$ redefines the dual variable solutions $\mathbf{y}_{k+1} = \mathbf{y}_{LP}$ and the strongly active set $\mathcal{A}_{+,k+1} = \{i : [\mathbf{y}_{k+1}]_i > 0\} \cup \mathcal{E}$. Finally, we increase the iteration index $k = k+1$ and repeat from the beginning until the final value of $t = 1$ is reached. We describe the three steps in Algorithm 1 below.

3.3 Path-following NMPC (pf-NMPC)

We now include the predictor-corrector path-following method in the online step of the asNMPC controller. We describe the pf-NMPC controller in Algorithm 2 and the predictor-corrector path-following method is invoked in the function `MFCQ_PC_PF`.

4. NUMERICAL CASE EXAMPLE

In this section we test our proposed method, i.e., the pf-NMPC controller and compare its results against the iNMPC controller. All simulations are done in `MATLAB` using `CasADi` algorithmic differentation tool (Andersson (2013)) version 3.2.0, which includes `IPOPT` as NLP solver. We use MINOS QP (Murtagh and Saunders (1982)) solver from TOMLAB and CPLEX as LP solver.

4.1 Process Description

We implement the pf-NMPC controller for a CSTR, taken from Diehl et al. (2011), with first order reaction $A \rightarrow B$ and modified objective function as well as different bound constraint values. The dynamic model, derived from mass balance, is

Algorithm 1 Predictor-corrector path-following method

Input: t, $\boldsymbol{\chi}$, $\mathbf{y}$ close to solution $(\boldsymbol{\chi}^*(t), \mathbf{y}^*(t))$ such that $\{\nabla_{\boldsymbol{\chi}} c_i(\boldsymbol{\chi}, t)\}_{\{i\in\mathcal{I}:\, \mathbf{y}_i>0\}\cup\mathcal{E}}$ is linearly independent, and Δt

Output: $\boldsymbol{\chi}$ and $\mathbf{y}$ at $\mathbf{p}_f$

function MFCQ_PC_PF($\boldsymbol{\chi}, \mathbf{y}, \mathbf{p}_0, \mathbf{p}_f, \Delta t$)
- Define parameter γ satisfying $0 < \gamma < 1$.
- Define $\mathcal{A}_+$.
- Set parameter $\eta_{max} < 1$.
- Set $k \leftarrow 0$.
- Set $t_k = 0$.
- **while** $t_k < 1$ **do**
  - Compute $\eta_k := \eta(\boldsymbol{\chi}_k, \mathbf{y}_k, t_k)$.
  - **if** $\eta_k > \eta_{max}$ **then**.
    - Decrease Δt.
    - $k \leftarrow k+1$.
  - **else**
    - Solve (**CorrectStep**) for $(\Delta_c\boldsymbol{\chi}, \Delta_+\mathbf{y})$.
    - Solve (**QPPredict**) for $(\Delta_p\boldsymbol{\chi}, \Delta_p\mathbf{y})$.
    - Set $(\Delta\boldsymbol{\chi}, \Delta\mathbf{y}) = (\Delta_p\boldsymbol{\chi}, \Delta_p\mathbf{y}) + (\Delta_c\boldsymbol{\chi}, \Delta_c\mathbf{y})$.
    - Compute $\eta_{k+\Delta} := \eta(\boldsymbol{\chi}_k + \Delta\boldsymbol{\chi},\, \mathbf{y}_k + \Delta\mathbf{y},\, t_k + \Delta t)$.
    - **if** $\eta_{k+\Delta} < \eta_{max}$ **then**
      - $\boldsymbol{\chi}_{k+1} \leftarrow \boldsymbol{\chi}_k + \Delta\boldsymbol{\chi}$
      - $\mathbf{y}_{k+1} \leftarrow \mathbf{y}_k + \Delta\mathbf{y}$
      - $t_{k+1} \leftarrow t_k + \Delta t$
      - $\mathbf{p}(t_k) = (1-t_k)\mathbf{p}_0 + t_k\mathbf{p}_f$
      - **if** $\eta_{k+\Delta} < \eta_k^{1+\gamma}$ **then** ▷ very good step
        - Increase Δt.
      - **end if**
      - Update $\mathcal{A}_+$.
      - Solve (**JumpLP**) to redefine $\mathbf{y}_{k+1}$.
      - Let $\mathcal{A}_+ = \{i : [\mathbf{y}_{k+1}]_i > 0\} \cup \mathcal{E}$.
    - **else**
      - Decrease Δt.
    - **end if**
    - $k \leftarrow k+1$.
  - **end if**
- **end while**
- Return $\boldsymbol{\chi}$

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**
- $[\boldsymbol{\chi}^*, \mathbf{y}^*] \leftarrow$ 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 $\mathbf{p}_0 = \mathbf{z}_{k+1}$
  - Set $\mathbf{p}_f = \mathbf{x}_{k+1}$
  - $\boldsymbol{\chi}^* \leftarrow$ MFCQ_PC_PF($\boldsymbol{\chi}^*, \mathbf{y}^*, \mathbf{p}_0, \mathbf{p}_f, \Delta t$)
  - Inject the first input move of $\boldsymbol{\chi}^*$ to the plant
  - Update initial state $\mathbf{x}_0 \leftarrow \mathbf{x}_{k+1}$
  - Set $k+1 \leftarrow k$
- **end if**

end for

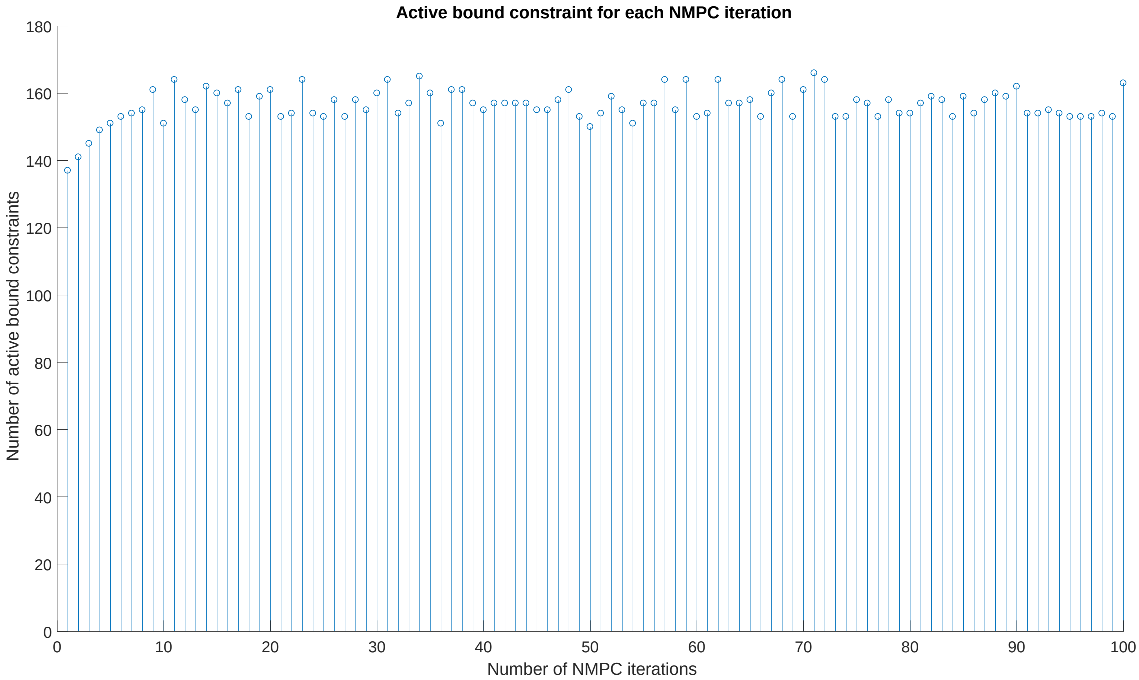

Fig. 1. Number of active bound constraints during the open-loop optimizations. The active bound constraints come from the lower bound of concentration c_B.

$$\frac{dc_A}{dt} = \frac{Q}{V}(c_{Af} - c_A) - kc_A \tag{14}$$
$$\frac{dc_B}{dt} = \frac{Q}{V}(-c_B) - kc_A,$$

where c_A and c_B are the concentration of components A and B, respectively. The rate constant is $k = 1.2 \frac{L}{mol\, minute}$, the reactor volume is $V = 10\, L$, and the feed concentration is $c_{Af} = 1 \frac{mol}{L}$. The control input is denoted by Q with unit $\frac{L}{minute}$ and the state variables are the concentration c_A and c_B. The economic objective function is

$$J = -Q, \tag{15}$$

which then is incorporated with the regularization term as in the equation (5)

$$\psi_m := J + \|c_A - 0.5\|^2 + 10^{-6}\|c_B - 0.4999\|^2 + \|Q - 12\|^2. \tag{16}$$

The weights are computed using Gershgorin bound criteria, giving 10^{-6} for the product B concentration. Bound constraints on the control and state are

$$10 \leq Q \leq 20,$$
$$0.49 \leq c_B \leq 1.$$

We run NMPC controllers with prediction horizon N = 50 for 100 minutes simulation time with sampling time 1 minute. We set initial $\Delta t = 0.5$ for the pf-NMPC controller and $\eta_{max} = 0.001$. We use direct collocation method to transcribe the open-loop optimal control problem yielding 452 optimization variables and 402 nonlinear equality constraints from the discretized state equations (14) as well as bound constraints for the control input and state variables. The real plant model is simulated by 'ode15' solver in `MATLAB`. We add noise to the measurement data where the noise is taken to have a normal distribution with zero mean and a variance of one percent of the steady state values.

The number of active bound constraints during the course of MPC iteration is shown in Figure 1. We check the linear independence of the Jacobian of the active bound constraints and equality constraints (as in the Definition 3 (LICQ)) and we find that the Jacobian is rank deficient, which implies LICQ is not satisfied. Another way to check, if LICQ does not hold, is that the the number of active constraints (equality constraints plus active bound constraints) exceeds the number of optimization variables.

We continue to check whether MFCQ holds. As in the Definition 4 (MFCQ), there exists a step **s** that satisfies the active inequality constraints as well as satisfying the the inactive inequality constraints, in addition to the linear independence of the equality constraints. The existence of the step **s** can be verified by solving an LP problem (Forsgren et al., 2002, Section 2.2). The path-following predictor-corrector method succeeds in using such step because it can give a feasible control input Q such that the concentration c_B stays in the feasible region. Moreover, we compute the Jacobian of the equality constraints and it always has full-row rank.

4.2 Comparison of Open-loop Optimization Results

Here, we compare open loop optimization solutions (predictive state and control) at iteration number two. The open loop solutions are the optimized solution given by the internal model used in the NMPC algorithm. The results are depicted in Figure 2 in which the solutions of

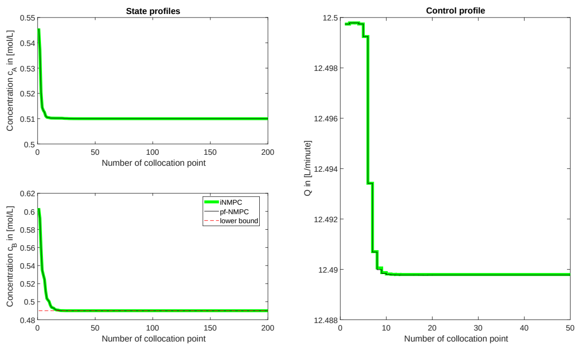

Fig. 2. Open loop solutions (state variables and control input) comparison at iteration number 2 for iNMPC and pf-NMPC controllers.

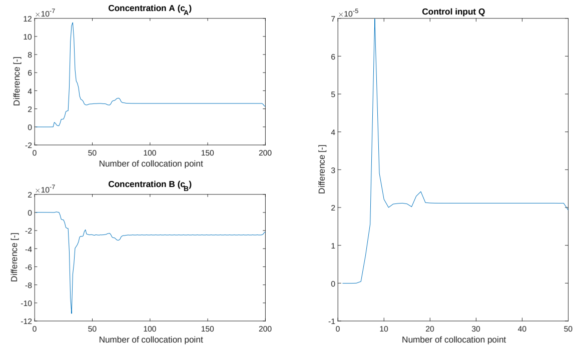

Fig. 3. The difference in predicted state variables and control input between iNMPC and pf-NMPC at iteration number 2.

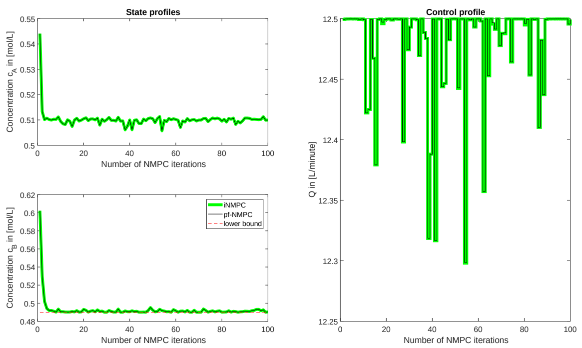

Fig. 4. Closed-loop solutions (state variables and control input) comparison of iNMPC and pf-NMPC controllers.

both controllers are overlapped. The resulted trajectories of state and control inputs of pf-NMPC follow precisely the iNMPC solutions, where the differences between the solutions are shown in Figure 3. The differences in the state variables and control input are in the order of 10^{-7} and 10^{-5}, respectively.

4.3 Closed-loop Results

We compare the closed-loop responses of pf-NMPC controller, which are obtained from the plant measurement data after injecting the first move of the optimized control

online optimization runtime (in sec.)			
	min	max	average
iNmpc	0.0680	0.1191	0.0884
pf-Nmpc	0.0416	0.1022	0.0479

Table 1.

input. As can be seen in Figure 4, again, the pf-NMPC solutions track accurately those of iNMPC. Furthermore, we compare runtime between iNMPC and pf-NMPC approaches in Table 1, where on average pf-NMPC approach gives almost two times speedup factor.

5. CONCLUSION

We have proposed the use of predictor-corrector path-following method, consisting of the three steps (corrector, predictor, and multiplier jump), for solving open-loop optimal control problem in NMPC controller setting, referred the method to as the pf-NMPC controller. We have shown that the pf-NMPC works as expected in the case example, which accurately track the solutions of iNMPC controller.

REFERENCES

Andersson, J. (2013). *A general-purpose software framework for dynamic optimization*. Ph.D. thesis, Arenberg Doctoral School, KU Leuven.

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

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

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

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

Forsgren, A., Gill, P.E., and Wright, M.H. (2002). Interior methods for nonlinear optimization. *SIAM review*, 44(4), 525–597.

Guddat, J., Vazquez, F.G., and Jongen, H.T. (1990). *Parametric optimization: singularities, pathfollowing and jumps*. Springer.

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

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

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

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