The Pennsylvania State University
The Graduate School
College of Engineering

CARLEMAN LINEARIZATION-BASED NONLINEAR MODEL PREDICTIVE CONTROL

A Thesis in
Chemical Engineering
by
Yizhou Fang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2015

The thesis of Yizhou Fang was reviewed and approved* by the following:

Antonios Armaou
Associate Professor of Chemical Engineering
Thesis Advisor

Themis Matsoukas
Professor of Chemical Engineering

Robert Rioux
Friedrich G. Helfferich Associate Professor of Chemical Engineering

Hosam Fathy
Associate Professor of Mechanical Engineering

Janna Maranas
Professor of Chemical Engineering
Graduate Program Chair

*Signatures are on file in the Graduate School.

Abstract

The need of tight operating conditions in chemical, pharmaceutical, and petroleum industries has given rise to the development of advanced process control. Model Predictive Control (MPC) started gaining attention three decades ago for optimal transitions between operating modes. Nonlinear MPC converts a constrained control problem of a nonlinear system into an optimization problem. This basic architecture makes Nonlinear MPC capable of handling large state-space multi-variable systems with constraints, and dealing with model-mismatches and disturbances readily.

The computation time of control policy is required to be less than one sampling time for online operation. However, this requirement is most of the times impossible to meet when the system has high nonlinearity. That becomes one of the most significant reasons holding back the application of Nonlinear MPC. As a result, there is strong motivation to develop an advanced formulation of Nonlinear MPC that demands less computational effort and thus decides the control actions faster.

The primary focus of this thesis is to develop an advanced formulation of Nonlinear MPC that decreases the amount of computational effort in order to circumvent feedback delay, to improve controller performance and to maintain stability of the system. Multiple mathematics tools combined with optimization techniques are implemented for the purpose of accelerated searching algorithms. The optimal control problem is formulated as a receding horizon one. An optimization problem is solved at each time the finite horizon moves on. Based on Carleman Linearization, the states of the system are extended to higher orders following the Kronecker product rule. The nonlinear dynamic process can thus be modeled with a bilinear representation while keeping nonlinear dynamic information. It enables analytical anticipation of system states and provides the searching algorithm with analytically computed sensitivity of the cost function to the control signals. The proposed method resembles both collocation and shooting methods for the following reasons. First, the states of the system are discretized explicitly in time while the sensitivity of the control signals is computed analytically. Second, the states are nonlinear functions of the control signals, releasing the optimization problem from equality constraints and reducing the number of design variables.

This thesis presents an introduction to MPC, Carleman Linearization and detailed derivations of the proposed method in Chapter 1 and 2. It also provides detailed description of resetting

extended states to compensate for the simulation errors caused by Carleman Linearization as an independent Chapter 3. Chapter 4 presents case-study examples to indicate the applications of the proposed method. Chapter 5 concludes the work and future plans.

A part of the work presented in this thesis has been published at American Control Conference, Chicago, IL on July 1st, 2015.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Dr. Armaou for offering this research project Carleman Linearization-based Nonlinear Model Predictive Control. Thank you for this brilliant idea and for advising my research.

I would like to thank Dr. Matsoukas, Dr. Rioux and Dr. Fathy for serving on my committee. Thank you for your time and your insights.

Many thanks to Dr. Belegendu and Dr. Fathy, who helped me acquired knowledge in Optimization and Optimal Control. Thank you for your wonderful lectures and thank you for the suggestions to my research and encouragement.

Many thanks to the National Science Foundation for the financial support to this research project.

I would love to give special thanks to my family in China for your love, trust and support.

Chapter 1 — Introduction

1.1 Model Predictive Control

Model Predictive Control (MPC) has attracted increasingly wide attention in chemical, pharmaceutical and petroleum refinery industries over the last three decades. The basic strategy of MPC is to use dynamic models to predict future behavior of a system and design inputs to manipulate the system into tracking reference trajectories [1]. The foundamental architecture of MPC is to determine the current control action by solving an open loop optimal control problem (OCP) within a finite horizon at each sampling time and implementing only the first control action in the sequence [2]. This architecture equips MPC with advantages over other control strategies such as coping with constraints, including state, input, output and process constraints, which is highly applicable in real industrial processes [3] [4]. MPC is practical for multiple-input-multiple-output systems based on its definition of converting the optimal control problem to an optimization one. The control policies adapt as the dynamic processes evolve since control actions are computed by repeatedly solving receding-horizon optimization problems. This property enables MPC to reject external disturbances and tolerate model mismatches [1] [7]. Reviews on MPC formulations, stability analysis and performance can be found in [8] [9].

Since MPC focuses on optimality rather than stability by nature, the stability of closed-loop process operation is an important open problem of MPC. To address the issue of stability, a large amount of research has been focused on Lyapunov-based formulations, which address stability issues, and the effect of initial condition on the feasibility in optimization [12] [13] [14].

Linear MPC is a relatively mature technology. It is based on a solid foundation of linear control theories and quadratic programming techniques. Over two thousand applications of Linear MPC have been reported by the end of last century [2].

More chanllenges and opportunities lie in Nonlinear MPC. Nonlinear MPC applies to nonlinear reactors and plants that vary over large regions of state space, including changeovers in continuous processes, tracking problems in startup and batch processes [2]. The development of large scale nonlinear programming (NLP) algorithms and dynamic optimization strategies further assure a promising future of Nonlinear MPC in industrial applications [11]. More reviews of Nonlinear MPC can be found in [15]. One of the most significant challenges in Nonlinear MPC is the issue of computational effort. MPC controllers require more computational effort than classical controllers. Due to the nonlinearity, optimization is non-convex for most of the cases, which leads to even greater increase in computational effort. Complex chemical processes with large number of states or high nonlinearity usually require a significant amount of computation. The resulting feedback delays, consequent loss of performance and potential stability issues become significant barriers to the industrial implementation of Nonlinear MPC [28]. In this thesis, we propose an approach to address this problem, Carleman Linearization-based Nonlinear MPC.

1.2 Carleman Linearization

In 1932, Torsten Carleman showed a finite dimensional system of nonlinear differential equation can be embedded into an infinite system of linear differential equations. This method is named as Carleman Linearization[18][19][20] [21].

To facilitate the introduction, we present a description of the Kronecker product rule, which Carleman linearization is based on. The Kronecker product of matrix $X \in C^{N\times M}$ and matrix $Y \in C^{L\times K}$ is defined as matrix $Z \in C^{(NL)\times(MK)}$.

$$X = \begin{vmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,M} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,M} \\ \cdots & \cdots & \cdots & \cdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,M} \end{vmatrix}, \quad Y = \begin{vmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,K} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,K} \\ \cdots & \cdots & \cdots & \cdots \\ y_{L,1} & y_{L,2} & \cdots & y_{L,K} \end{vmatrix}$$

$$Z = X \otimes Y \equiv \begin{vmatrix} x_{1,1}Y & x_{1,2}Y & \cdots & x_{1,M}Y \\ x_{2,1}Y & x_{2,2}Y & \cdots & x_{2,M}Y \\ \cdots & \cdots & \cdots & \cdots \\ x_{N,1}Y & x_{N,2}Y & \cdots & x_{N,M}Y \end{vmatrix}$$

We represent nonlinear dynamic systems in the following form:

$$\dot{x} = f(x) + \sum_{j=1}^{m} g_j(x)u_j \tag{1.1}$$
$$x(t_0) = x_0$$

where $x \in R^n$ is the state vector, and $u_j \in R^m, \forall 1 \leq j \leq m$ are the vectors of control inputs. $f(x)$ and $g_j(x)$ are nonlinear vector functions.

In chemical processes, exponential terms are commonly seen, which can be approximated by the definition of matrix exponential:

$$\exp(A) = \sum_{l=0}^{\infty} \frac{1}{l!} A^l \tag{1.2}$$

For the simplicity of derivation, we assume the nominal operating point of the system is at the origin $x = 0$. Nonlinear vector functions $f(x)$ and $g_j(x)$ are expanded by Maclaurin series in the following form:

$$f(x) = f(0) + \sum_{k=1}^{\infty} \frac{1}{k!} \partial f_{[k]}|_{x=0} x^{[k]}$$
$$g_j(x) = g_j(0) + \sum_{k=1}^{\infty} \frac{1}{k!} \partial g_{j[k]}|_{x=0} x^{[k]}$$

So nonlinear dynamic systems can be approximated by a polynomial form:

$$\dot{x} \cong \sum_{k=0}^{p} A_k x^{[k]} + \sum_{j=1}^{m} \sum_{k=0}^{p} B_{jk} x^{[k]} u_j$$

A_k denotes $\frac{1}{k!} \partial f_{[k]}|_{x=0}$ and B_{jk} denotes $\frac{1}{k!} \partial g_{j[k]}|_{x=0}$, $\forall k$. A_0 denotes $f(0)$ and B_{j0} denotes $g_j(0)$. The polynomial order p is supposed to be high enough to reduce truncation errors [20].

To implement Carleman linearization, the states of the system x are extended to $x_\otimes = [x^T x^{[2]^T} \cdots x^{[p]^T}]^T$, where $x^{[p]}$ denotes the p-th order Kronecker product of x. The bilinear formulation $\dot{x}_\otimes = \mathcal{A} x_\otimes + \sum_{j=1}^{m} (\mathcal{B}_j x_\otimes + \mathcal{B}_{j0}) u_j + \mathcal{C}$ carries the information of nonlinear dynamic constraints. $\mathcal{A}$, $\mathcal{B}_j$, $\mathcal{B}_{j0}$, and $\mathcal{C}$ are matrices of the form

$$\mathcal{A}=\begin{vmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,p} \\ A_{2,0} & A_{2,1} & \cdots & A_{2,p-1} \\ 0 & A_{3,0} & \cdots & A_{3,p-2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & A_{p,1} \end{vmatrix}, \mathcal{C}=\begin{vmatrix} A_{1,0} \\ 0 \\ 0 \\ \cdots \\ 0 \end{vmatrix}, \mathcal{B}_j=\begin{vmatrix} B_{j1,1} & B_{j1,2} & \cdots & B_{j1,p} \\ B_{j2,0} & B_{j2,1} & \cdots & B_{j2,p-1} \\ 0 & B_{j3,0} & \cdots & B_{j3,p-2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & B_{jp,1} \end{vmatrix}, \mathcal{B}_{j0}=\begin{vmatrix} B_{j1,0} \\ 0 \\ 0 \\ \cdots \\ 0 \end{vmatrix},$$

where $A_{k,i}=\sum_{l=0}^{k-1} I_n^{[l]} \otimes A_i \otimes I_n^{[k-1-l]}$ and $B_{jk,i}=\sum_{l=0}^{k-1} I_n^{[l]} \otimes B_{ji} \otimes I_n^{[k-1-l]}$.

One important assumption for the analysis in the following sections is that the control signals are all piecewise constant, which is generally the case in industrial process MPC. Thus, the formulation $\dot{x}_{\otimes}=\mathcal{A}x_{\otimes}+\sum_{j=1}^{m}(\mathcal{B}_j x_{\otimes}+\mathcal{B}_{j0})u_j+\mathcal{C}$ allows for analytical integration of nonlinear models. Providing the sensitivity of the cost function: $\int_{t_0}^{t_f} J(x,U)\mathrm{d}t$ to $u_{k,\mathcal{K}}$ (the $\mathcal{K}$-th control action in the sequence of the k-th design variable) also accelerates the computation of the optimal control policy [4] [5][6]. More detailed introduction to Carleman Linearization can be found in [18][19][20] [21].

Chapter 2 — Bilinear Carleman Linearization-Based MPC

2.1 General MPC Controller Design Methodology

In general MPC formulation, the optimal control problem is recast as a recursion of receding finite-horizon optimization problems at every time point t_0, which have a general form:

$$
\begin{gathered}
U^* = arg\min_{U} \int_{t_0}^{t_f} J(x,U)\mathrm{d}t \\
s.t. \\
u_j(t) = \sum_{i=1}^{N} U_{j,i} \cdot \mathbf{B}(t; T_{i-1}; T_i), \forall j = 1, \cdots, m \\
T_0 = t_0, T_N \leq t_f \\
\dot{x} - f(x) - \sum_{j=1}^{m} g_j(x) u_j(t) = 0 \\
x(t_0) = x_0 \\
f^c(x,U) \leq 0
\end{gathered}
$$

J is the cost function and x is the vector of state variables.

U denotes the matrix of control inputs that consists of $U_{j,i}, \forall j = 1, \cdots, m, \forall i = 1, \cdots, N$. $U_{j,i}$ is the i-th decision for the j-th manipulated variable, which means it is the signal of the j-th control input in its corresponding sampling time $(T_{i-1}, T_i]$.

The sampling time $(T_{i-1}, T_i]$ is also defined as the i-th action horizon, which has a length of ΔT_i.

$f(x)$ and $g_j(x)$ are nonlinear vector functions, accounting for the impacts of the states and the j-th control input respectively. The summation of the action horizons is the control horizon. N is the number of action horizons. T_0 is the beginning of the control horizon and T_N is the end of the control horizon. t_0, same as T_0 is the beginning of the prediction horizon and t_f is the end of the prediction horizon.

We define $\mathbf{B}(t; T_{i-1}; T_i) = H(t - T_{i-1}) - H(t - T_i)$ as a rectangular pulse function, where H is the standard Heaviside function. T_{i-1} and T_i denote the initiation time and the termination time respectively.

x_0 is the initial condition of system states.

f^c denotes the vector function of equality and inequality constraints.

2.2 Bilinear Carleman Linearization-based MPC (BCMPC) Formulation

With extended states $x_\otimes$ and extended coefficient matrices $\mathcal{A}$, $\mathcal{B}_j$, $\mathcal{B}_{j0}$, and $\mathcal{C}$ through Carleman Linearization, we represent the system dynamic constraint in a bilinear representation:

$$\dot{x}_\otimes = \mathcal{A}x_\otimes + \sum_{j=1}^{m}(\mathcal{B}_j x_\otimes + \mathcal{B}_{j0})U_{j,i} + \mathcal{C}, \;\; t \in (T_{i-1},\, T_i]$$

During each sampling time $t \in (T_{i-1},\, T_i]$, each $U_{j,i}$ is a piece-wise constant control action. The future state is predicted with the anayltical solution of the equation above

$$x(t)_\otimes = \exp\left[\big(\mathcal{A} + \sum_{j=1}^{m}\mathcal{B}_j x_\otimes U_{j,i}\big)\big(t - T_{i-1}\big)\right] x\big(T_{i-1}\big)_\otimes$$
$$+ \int_{T_{i-1}}^{t} \exp\left[\big(\mathcal{A} + \sum_{j=1}^{m}\mathcal{B}_j x_\otimes U_{j,i}\big)\big(t - \tau\big)\right] \mathrm{d}\tau \cdot \big(\sum_{j=1}^{m}\mathcal{B}_{j0}U_{j,i} + \mathcal{C}\big)$$

Remark 1: Deviation around a nominal point

Ideally Carleman Linearization is performed around a nominal point. This means every state variable and every control input is in the form of deviation from a nominal point. Since the optimization involves algebra of large matrices in the proposed formulation, we need to express both the states and the inputs in deviations to reduce numerical errors.

In Chapter 3, we will discuss simulation errors caused by Carleman Linearization and the algorithms to minimize those errors by resetting extended states.

Remark 2: Dimensionality issue

As the order of Carleman Linearization grows, the dimension of $x_{\otimes}$, $\mathcal{A}$, $\mathcal{B}_j$, $\mathcal{B}_{j0}$, and $\mathcal{C}$ all grow at a geometric pace respectively. A system has n state variables and is approximated with p-th order Carleman Linearization. The dimension of extened states $x_{\otimes} = [x^T x^{[2]^T} \cdots x^{[p]^T}]^T$ is $\sum_{i=1}^{p} n^i$. The dimensions of $\mathcal{A}$, $\mathcal{B}_j$ are both $(\sum_{i=1}^{p} n^i) \times (\sum_{i=1}^{p} n^i)$.

This large expansion in dimensionality will cost extra computational requirements. One solution is to merge identical terms in the state vector $x_{\otimes}$ to yield $x_{\otimes,reduced}$ [20][21]. For example, to approach a 2-state nonlinear system with 3-rd order Carleman Linearization, the original state vector $x = [x_1, x_2]^T$ with a dimension of 2 is extended to a dimension of 14.

$$x_{\otimes} = [x_1, x_2, x_1^2, x_1x_2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1^2x_2, x_1x_2^2, x_1^2x_2, x_1x_2^2, x_1x_2^2, x_2^3]^T$$

This extended state vector $x_{\otimes}$ can be reduced to the following state vector with a dimension of 9.

$$x_{\otimes,reduced} = [x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3]^T$$

The dimensions of constant matrices $\mathcal{A}$, $\mathcal{B}_j$, $\mathcal{B}_{j0}$, and $\mathcal{C}$ can also be reduced similarly.

Remark 3: Reformulation Optimizing ΔU and ΔT

The control move ΔU, which is the change of the control action with regard to the previous step and the length of each action horizon ΔT_i can both be design variables. It means the bilinear Carleman Linearization-based MPC algorithm (BCMPC) can be reformulated as optimizing the control move ΔU and optimizing the action horizon ΔT_i [23][24].

Optimizing the control move ΔU is computationally more favored than optimizing U in the cases that the set-points of control actions are unknown. Unknown set-points of control actions are commonly seen in economic-oriented MPC (EMPC). So optimizing the control move ΔU has more applications in EMPC cases.

The action horizon, or the sampling time, is the time interval that a piece-wise constant control action $U_{j,i}$ lasts for. ΔT_i is the length of the corresponding action horizon. In traditional MPC formulations, the length of the control horizon and the length of each action horizon are both setteld, and the number of control actions throughout the operating process is a fixed number. We propose an idea to optimize the length of each action horizon. In this way we make MPC a more

powerful tool to achieve the purpose of optimal control. While the length of the action horizons are changing, the length of per resetting interval can either change or remain the same throughout the simulation depending on the specific operating process.

Remark 4: Different Control Horizon and Prediction Horizon

The approaches of stabilizing systems controlled by MPC feedback laws can be divided into three major categories, (i) penalty on the deviation of terminal state from the set-point (ii) applying local control Lyapunov functions in the teminal cost, and (iii) using a long enough optimization horizon [25]. [25] [26] proved that for an optimization horizon length N, the value of cost function is bounded by some positive real number L. $N \geq [1+L\ln(\gamma(L-1))]$ ensures closed-loop stability, where γ is a real number that can be chosen as $L-1$ in the worst case.

In our proposed formulation of bilinear Carleman Linearizatrion-based MPC, we follow the third approach and choose a prediction horizon longer than the control horizon and make sure the prediction horizon is long enough to stablize the open-loop unstable system.

2.3 Detailed Formulation

Define the following notations for the purpose of simplicity in derivations:

$$
\begin{aligned}
&\tilde{\mathcal{A}}_i = \mathcal{A} + \sum_{j=1}^{m} \mathcal{B}_j U_{j,i}, \\
&\mathcal{G}x(U_i) = \exp(\tilde{\mathcal{A}}_i \Delta T_i), \\
&\mathcal{G}u(U_i) = \tilde{\mathcal{A}}_i^{-1}[\mathcal{G}x(U_i) - I], \\
&\mathcal{DG}x(U_i) = \exp(\tilde{\mathcal{A}}_i \frac{\Delta T_i}{r}), \\
&\mathcal{DG}u(U_i) = \tilde{\mathcal{A}}_i^{-1}[\mathcal{DG}x(U_i) - I], \\
&\mathcal{F}_i = \sum_{j=1}^{m} \mathcal{B}_{j0} U_{j,i} + \mathcal{C}
\end{aligned}
$$

We discretize each action horizon $(T_{i-1},\ T_i]$ into r smaller intervals $(T_{i-1},\ T_{i-1+\frac{1}{r}}, \ldots,\ T_{i-\frac{1}{r}},\ T_i]$ evenly. Then we integrate the states over each small interval of the length $\frac{\Delta T_i}{r}$ to obtain the analytical prediction of extended states $x_{\otimes,i-1+\frac{1}{r}}$, $x_{\otimes,i-1+\frac{2}{r}}$, $\cdots$, $x_{\otimes,i-\frac{1}{r}}$, $x_{\otimes,i}$ and then reset extended

states respectively using their original states $x_{i-1+\frac{1}{r}}, x_{i-1+\frac{2}{r}}, \cdots, x_{i-\frac{1}{r}}, x_i$:

$$
\begin{aligned}
x_{\otimes,i-1+\frac{1}{r}} &= \mathcal{DG}x(U_i)x_{\otimes,i-1} + \mathcal{DG}u(U_i)\mathcal{F}_i \\
x_{\otimes,i-1+\frac{2}{r}} &= \mathcal{DG}x(U_i)x_{\otimes,i-1+\frac{1}{r}} + \mathcal{DG}u(U_i)\mathcal{F}_i \\
&\cdots\cdots \\
x_{\otimes,i-\frac{1}{r}} &= \mathcal{DG}x(U_i)x_{\otimes,i-\frac{2}{r}} + \mathcal{DG}u(U_i)\mathcal{F}_i \\
x_{\otimes,i} &= \mathcal{DG}x(U_i)x_{\otimes,i-\frac{1}{r}} + \mathcal{DG}u(U_i)\mathcal{F}_i
\end{aligned}
$$

The detailed reason of resetting extended states is presented in Chapter 3.

After resetting $x_{\otimes,i-1+\frac{1}{r}}, x_{\otimes,i-1+\frac{2}{r}}, \cdots, x_{\otimes,i-\frac{1}{r}}, x_{\otimes,i}$ following the Kronecker product rule, the integral of the extended states over ΔT_i becomes:

$$
\begin{aligned}
\int_{T_{i-1}}^{T_i} x_{\otimes}\,\mathrm{d}t = \mathcal{DG}u(U_i)[&x_{\otimes,i-1(reset)} + x_{\otimes,i-1+\frac{1}{r}(reset)} + \cdots + x_{\otimes,i-\frac{1}{r}(reset)}] \\
&+ \tilde{\mathcal{A}_i}^{-1}[r \cdot \mathcal{DG}u(U_i) - \Delta T_i \cdot I]\mathcal{F}_i
\end{aligned} \tag{2.1}
$$

which is used to construct the cost function:

$$
\int_{T_0}^{T_N} J\,\mathrm{d}t \cong J_0(T_N - T_0) + \sum_{i=1}^{N}(J_A + \sum_{j=1}^{m} J_{Nj}U_{\otimes,j,i}\otimes)\int_{T_{i-1}}^{T_i} x_{\otimes}\mathrm{d}t + \sum_{i=1}^{N}\sum_{j=1}^{m} J_{Bj}U_{\otimes,j,i}\Delta T_i \tag{2.2}
$$

The sensitivity of the cost function to $U_{k,\mathcal{K}}$ is:

$$
\begin{aligned}
\frac{\partial}{\partial \Delta U_{k,\mathcal{K}}}\int_{T_0}^{T_N} J\,\mathrm{d}t &= (J_A + \sum_{j=1}^{m} J_{Nj}U_{\otimes,j,i}\otimes)\sum_{i=\mathcal{K}}^{N}\int_{T_{i-1}}^{T_i}\frac{\partial x_{\otimes}}{\partial \Delta U_{k,\mathcal{K}}}\mathrm{d}t \\
&+ J_{Nk}(\partial U_{\otimes,k,\mathcal{K}}) \otimes \int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} x_{\otimes}\mathrm{d}t + J_{Bk}(\partial(U_{\otimes,k,\mathcal{K}}))\Delta T_{\mathcal{K}}
\end{aligned} \tag{2.3}
$$

Define

$$
\sum_{i=\mathcal{K}}^{N}\int_{T_{i-1}}^{T_i}\frac{\partial x_{\otimes}}{\partial U_{k,\mathcal{K}}}\mathrm{d}t = G_k(i,\mathcal{K}) \tag{2.4}
$$

$$G_k(i,\mathcal{K}) = \begin{cases} \int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} \frac{\partial x_{\otimes}}{\partial U_{k,\mathcal{K}}} \mathrm{d}t, i = \mathcal{K} \\ \int_{T_{\mathcal{K}}}^{T_{\mathcal{K}+1}} \frac{\partial x_{\otimes}}{\partial x_{\otimes,\mathcal{K}}} \mathrm{d}t \frac{\partial x_{\otimes,\mathcal{K}}}{\partial U_{k,\mathcal{K}}}, i = \mathcal{K}+1 \\ \sum_{i=\mathcal{K}+2}^{N} \int_{T_{i-1}}^{T_i} \frac{\partial x_{\otimes}}{\partial x_{\otimes,i-1}} \mathrm{d}t (\prod_{l=\mathcal{K}+1}^{i-1} \frac{\partial x_{\otimes,l}}{\partial x_{\otimes,l-1}}) \frac{\partial x_{\otimes,\mathcal{K}}}{\partial U_{k,\mathcal{K}}}, i > \mathcal{K}+1 \end{cases}$$

and

$$\partial\, U_{\otimes,k,\mathcal{K}} = [1 \quad 2\, U_{k,\mathcal{K}} \cdots p\, U_{k,\mathcal{K}}^{p-1}] \tag{2.5}$$

$$\frac{\partial x_{\otimes,i}}{\partial x_{\otimes,i-1}} = \mathcal{G}x(U_i) \tag{2.6}$$

$$\int_{T_{i-1}}^{T_i} \frac{\partial x_{\otimes}}{\partial x_{\otimes,i-1}} \mathrm{d}t = \mathcal{G}u(U_i) \tag{2.7}$$

Define $\mathcal{E}_{\mathcal{K}} = \exp(\tilde{\mathcal{A}_{\mathcal{K}}} \Delta T_{\mathcal{K}})$ and $\mathcal{DE}_{\mathcal{K}} = \exp(\tilde{\mathcal{A}_{\mathcal{K}}} \frac{\Delta T_{\mathcal{K}}}{r})$.

$$\frac{\partial x_{\otimes,\mathcal{K}}}{\partial U_{k,\mathcal{K}}} = \frac{\partial \mathcal{E}_{\mathcal{K}}}{\partial U_{k,\mathcal{K}}} x_{\otimes_{\mathcal{K}-1}} + \tilde{\mathcal{A}_{\mathcal{K}}}^{-1} \frac{\partial \mathcal{E}_{\mathcal{K}}}{\partial U_{k,\mathcal{K}}} \mathcal{F}_{\mathcal{K}} + \mathcal{G}u(U_{\mathcal{K}}) \mathcal{B}_{k0} - \tilde{\mathcal{A}_{\mathcal{K}}}^{-1} \mathcal{B}_k \mathcal{G}u(U_{\mathcal{K}}) \mathcal{F}_{\mathcal{K}} \tag{2.8}$$

The sensitivity of the time integral of extended states $x_{\otimes}$ is the following where resetting extended states can be performed explicitly.

$$\begin{aligned} &\int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} \frac{\partial x_{\otimes}}{\partial U_{k,\mathcal{K}}} \mathrm{d}t = \int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} \frac{\partial \mathcal{DE}_{\mathcal{K}}}{\partial U_{k,\mathcal{K}}} \mathrm{d}t \cdot [x_{\otimes,\mathcal{K}-1(reset)} + \cdots + x_{\otimes,\mathcal{K}-\frac{1}{r}(reset)}] \\ &+ r \cdot \{\tilde{\mathcal{A}_{\mathcal{K}}}^{-1} \int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} \frac{\partial \mathcal{DE}_{\mathcal{K}}}{\partial U_{k,\mathcal{K}}} \mathrm{d}t \cdot \mathcal{F}_{\mathcal{K}} + \tilde{\mathcal{A}_{\mathcal{K}}}^{-1} [\mathcal{DG}u(U_{\mathcal{K}}) - \frac{\Delta T_{\mathcal{K}}}{r} \cdot I] \mathcal{B}_{k0} \\ &- \tilde{\mathcal{A}_{\mathcal{K}}}^{-1} \mathcal{B}_k \tilde{\mathcal{A}_{\mathcal{K}}}^{-1} [\mathcal{DG}u(U_{\mathcal{K}}) - \frac{\Delta T_{\mathcal{K}}}{r} \cdot I] \mathcal{F}_{\mathcal{K}}\} \end{aligned} \tag{2.9}$$

Based on the definition of matrix exponential,

$\frac{\partial \mathcal{E}_{\mathcal{K}}}{\partial U_{k,\mathcal{K}}}$ and $\int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} \frac{\partial \mathcal{D}\mathcal{E}_{\mathcal{K}}}{\partial U_{k,\mathcal{K}}} \mathrm{d}t$ can both be computed analytically:

$$\frac{\partial \mathcal{E}_{\mathcal{K}}}{\partial U_{k,\mathcal{K}}} = \sum_{l=1}^{\infty} \frac{(\Delta T_{\mathcal{K}})^l}{l!} \sum_{\lambda=1}^{l} \tilde{\mathcal{A}}_{\mathcal{K}}^{\lambda-1} \mathcal{B}_k \tilde{\mathcal{A}}_{\mathcal{K}}^{l-\lambda} \tag{2.10}$$

$$\int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} \frac{\partial \mathcal{D}\mathcal{E}_{\mathcal{K}}}{\partial U_{k,\mathcal{K}}} \mathrm{d}t = \sum_{l=1}^{\infty} \frac{(\frac{\Delta T_{\mathcal{K}}}{r})^{l+1}}{(l+1)!} \sum_{\lambda=1}^{l} \tilde{\mathcal{A}}_{\mathcal{K}}^{\lambda-1} \mathcal{B}_k \tilde{\mathcal{A}}_{\mathcal{K}}^{l-\lambda} \tag{2.11}$$

The sensitivity of the cost function to the $\mathcal{K}$-th action horizon $\Delta T_{\mathcal{K}}$ is:

$$\frac{\partial}{\partial \Delta T_{\mathcal{K}}} \int_{T_0}^{T_N} J \, \mathrm{d}t \cong J_0 + \sum_{i=\mathcal{K}}^{N} (J_A + \sum_{j=1}^{m} J_{Nj} U_{\otimes,j,i} \otimes) \int_{T_{i-1}}^{T_i} \frac{\partial x_{\otimes}}{\partial \Delta T_{\mathcal{K}}} \mathrm{d}t + \sum_{j=1}^{m} J_{Bj} U_{\otimes,j,\mathcal{K}} \tag{2.12}$$

Denote the sensitivity of the time integral of extended states as:

$$\sum_{i=\mathcal{K}}^{N} \int_{T_{i-1}}^{T_i} \frac{\partial x_{\otimes}}{\partial \Delta T_{\mathcal{K}}} \mathrm{d}t = H(i, \mathcal{K})$$

$$H(i,\mathcal{K}) = \begin{cases} \int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} \frac{\partial x_{\otimes}}{\partial \Delta T_{\mathcal{K}}} \mathrm{d}t, i = \mathcal{K} \\ \int_{T_{\mathcal{K}}}^{T_{\mathcal{K}+1}} \frac{\partial x_{\otimes}}{\partial x_{\otimes,\mathcal{K}}} \mathrm{d}t \frac{\partial x_{\otimes,\mathcal{K}}}{\partial \Delta T_{\mathcal{K}}}, i = \mathcal{K}+1 \\ \sum_{i=\mathcal{K}+2}^{N} \int_{T_{i-1}}^{T_i} \frac{\partial x_{\otimes}}{\partial x_{\otimes,i-1}} \mathrm{d}t (\prod_{l=\mathcal{K}+1}^{i-1} \frac{\partial x_{\otimes,l}}{\partial x_{\otimes,l-1}}) \frac{\partial x_{\otimes,\mathcal{K}}}{\partial \Delta T_{\mathcal{K}}}, i > \mathcal{K}+1 \end{cases}$$

and

$$\frac{\partial x_{\otimes,\mathcal{K}}}{\partial \Delta T_{\mathcal{K}}} = \tilde{\mathcal{A}}_{\mathcal{K}} \mathcal{G}x(U_{\mathcal{K}}) x_{\otimes,\mathcal{K}-1} + \mathcal{G}x(U_{\mathcal{K}}) F_{\mathcal{K}} \tag{2.13}$$

$$\int_{T_{\mathcal{K}-1}}^{T_{\mathcal{K}}} \frac{\partial x_{\otimes}}{\Delta T_{\mathcal{K}}} \mathrm{d}t = x_{\otimes,\mathcal{K}} \tag{2.14}$$

It is required to consider the effect of resetting extended states to calculate the sensitivity accu-

rately. Equation (2.6) indicates the sensitivity of extended states at the next sampling time to the accurate extended states at the current sampling time. This means $\frac{\partial x_{\otimes,l}}{\partial x_{\otimes,l-1(reset)}} = \mathcal{G}x(U_l)$.

Chain rule is applied in order to achieve a more accurate sensitivity.

$$\frac{\partial x_{\otimes,l(reset)}}{\partial x_{\otimes,l-1(reset)}} = \frac{\partial x_{\otimes,l(reset)}}{\partial x_l} \cdot \frac{\partial x_l}{\partial x_{\otimes,l}} \cdot \frac{\partial x_{\otimes,l}}{\partial x_{\otimes,l-1(reset)}} \tag{2.15}$$

$\frac{\partial x_{\otimes,l(reset)}}{\partial x_l}$ and $\frac{\partial x_l}{\partial x_{\otimes,l}}$ can be readily calculated based on the number of state variables n and the dimension of extended states, P.

$$\frac{\partial x_{\otimes,l(reset)}}{\partial x_l} = [\frac{\partial x_{l(reset)}}{\partial x_l}^T, \frac{\partial x^{[2]}_{l(reset)}}{\partial x_l}^T, \cdots, \frac{\partial x^{[p]}_{l(reset)}}{\partial x_l}^T]^T$$

For example, in a system of $n = 2$ state variables with 3rd order Carleman Linearization, the dimension P of extended states $x_\otimes$ is 14.

$$\frac{\partial x_{l(reset)}}{\partial x_l} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T$$

$$\frac{\partial x^{[2]}_{l(reset)}}{\partial x_l} = \begin{bmatrix} 2x_1 & x_2 & x_2 & 0 \\ 0 & x_1 & x_1 & 2x_2 \end{bmatrix}^T$$

$$\frac{\partial x^{[3]}_{l(reset)}}{\partial x_l} = \begin{bmatrix} 3x_1^2 & 2x_1x_2 & 2x_1x_2 & x_2^2 & 2x_1x_2 & x_2^2 & x_2^2 & 0 \\ 0 & x_1^2 & x_1^2 & 2x_1x_2 & x_1^2 & 2x_1x_2 & 2x_1x_2 & 3x_2^2 \end{bmatrix}^T$$

$\frac{\partial x_l}{\partial x_{\otimes,l}}$ has a dimension of $n \times P$. It consists of an identity matrix of a dimension $n \times n$ and the rest of the elements being zeros.

$$\frac{\partial x_l}{\partial x_{\otimes,l}} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & \dots & 0 \end{bmatrix}$$

Similarly, Chain rule is applied in the calculation of $\frac{\partial x_{\otimes,\mathcal{K}}}{\partial \Delta U_{k,\mathcal{K}}}$ and $\frac{\partial x_{\otimes,\mathcal{K}}}{\partial \Delta T_{\mathcal{K}}}$ to reset extended states.

Chapter 3 — Resetting Extended States

3.1 Reason of Resetting Extended States

In addition to the truncation errors caused by polynomial approximation to the original nonlinear system, Carleman Linearization introduces simulation errors because of the inconsistency within the original states and the extended states. This directly leads to integration errors when the bilinear representation is integrated over a period of time to predict future states.

For the purpose of symplicity, we use a nonlinear system without control actions as an example: $\dot{x} = f(x)$. Through Taylor expansion, the nonlinear system is approximated with a polynomial form:

$$\dot{x} \cong A_0 + A_1 x + A_2 x^2 + \cdots + A_p x^p$$

After extending the orginial states x to extended states $x_{\otimes}$, the next approximation is taken when the dynamics of the system are represented with a linear expression: $\dot{x}_{\otimes} = \mathcal{A} x_{\otimes}$

$$\dot{x}_{\otimes} = [\dot{x}^T \quad \dot{x}^{[2]^T} \quad \cdots \quad \dot{x}^{[p]^T}]^T$$

The dynamics of the k-th order extended states, $\dot{x}^{[k]},\ \forall\ 1 \leq k \leq p$ has the expression:

$$\begin{aligned}\dot{x}^{[k]} &= x^{[k-1]} \sum_{l=0}^{k-1} I_n^{[l]} \otimes \dot{x} \otimes I_n^{[k-1-l]} \\ &\cong x^{[k-1]} \sum_{l=0}^{k-1} I_n^{[l]} \otimes (\sum_{n=0}^{p} A_n x^{[n]}) \otimes I_n^{[k-1-l]}\end{aligned}$$

which is represented with orders of $x^{[k-1]^T}, x^{[k]^T}, \cdots, x^{[p+k-1]^T}$. But in $\dot{x}_{\otimes} = \mathcal{A} x_{\otimes}$, the highest

order is capped at $x^{[p]}$. So the expression of $\dot{x}^{[k]}$ is truncated to:

$$\dot{x}^{[k]} \cong x^{[k-1]} \sum_{l=0}^{k-1} I_n^{[l]} \otimes \Big(\sum_{n=0}^{p-k+1} A_n x^{[n]} \Big) \otimes I_n^{[k-1-l]}$$

The dynamic information carried by $x^{[p+1]^T}, \cdots, x^{[p+k-1]^T}$ is all lost in this expression. As the order k grows, more terms in the expression of $\dot{x}^{[k]}$ will be truncated and thus there remain more simulation errors. As the simulation proceeds on, the higher the order of $x^{[k]}$ is, the faster the simulation errors accumulate and that causes inconsistency between different orders of states within the full extended states $x_{\otimes}$.

We discard the terms $x^{[2]^T} \cdots x^{[p]^T}$ in $x_{\otimes}$ and use only the first order states x to re-extend the states to higher orders following the Kronecker product rule and obtain new extended states denoted as $x_{\otimes(reset)}$. This process is repeated frequently during simulation, and is defined as "re-setting the extended states".

In the design of MPC formulation, each action horizon is discretized into smaller "resetting intervals". We reset the extended states at the end of per "resetting interval" following the Kronecker product rule to minimize integration errors caused by Carleman Linearization. We discretize the i-th action horizon ΔT_i into r samller "resetting intervals", so $[x_{\otimes,i-1}, x_{\otimes,i}] \Rightarrow [x_{\otimes,i-1}, \cdots, x_{\otimes,i-\frac{2}{r}}, x_{\otimes,i-\frac{1}{r}}, x_{\otimes,i}]$. The number of resetting intervals r is a tuning parameter depending on specific cases.

3.2 Example Application

We use a classic open-loop stable example, the Van de Vusse Reactor, to discuss the effect of resetting extended states. In the worst-case scenario, we do not know the nominal operating condition and perform Carleman Linearization around the trivial steady-state, or the wash-out condition. In this isothermal CSTR, controlling the feed flow rate is an approach to control the product concentration since it changes the residence time in a constant volume reactor. The parallel reactions:

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C,$$

$$2A \xrightarrow{k_3} D$$

Open-loop Stable CSTR Parameters		
Parameter	Description	Value
k_1	Reaction Rate Constant	$\frac{5}{6}\ min^{-1}$
k_2	Reaction Rate Constant	$\frac{5}{3}\ min^{-1}$
k_3	Reaction Rate Constant	$\frac{1}{6}\ mol/iter \cdot min$
C_{Af}	Feeding Concentration of A	$10\ gmol/liter$
C_{A0}	Initial Concentration of A	$3\ gmol/liter$
C_{B0}	Initial Concentration of B	$1.117\ gmol/liter$

Table 3.1: Van de Vusse CSTR Parameters

A(=cyclopentadiene) is the reactant. B(=cyclopentenol) is the intermediate component and the desired product. C(=cyclopentanediol) and D(=dicyclopentadiene) are side products. Derived from conservation equations, the dynamic constraints are expressed by the following two ODEs:

$$\dot{C_A} = \frac{F}{V}(C_{Af} - C_A) - k_1 C_A - k_3 C_A^2$$
$$\dot{C_B} = -\frac{F}{V}C_B + k_1 C_A - k_2 C_B$$

where $\frac{F}{V}$ is the feed flow rate divided by the reactor volume, known as the dilution rate; this is the control input. C_{Af} is the concentration of the feeding reactant A, as a fixed parameter. The other parameters of the system are listed in Table 3.1 [27].

The CSTR system initiates at a steady-state of $C_{A0} = 3\ gmol/L$, $C_{B0} = 1.117\ gmol/L$. The input $\frac{F}{V}$ changes as a step function, which starts at $0.5714\ min^{-1}$, decreases by $0.025\ min^{-1}$ at $t = 2\ min$ and by another $0.025\ min^{-1}$ at $t = 4\ min$.

The solid black lines denote the numerical results simulated with Matlab ode45. They show the reference values that the prediction by Carleman Linearization-based model is supposed to track. Ideally the simulation results modeled by Carleman Linearization should be accurate enough to overlap with the solid black lines.

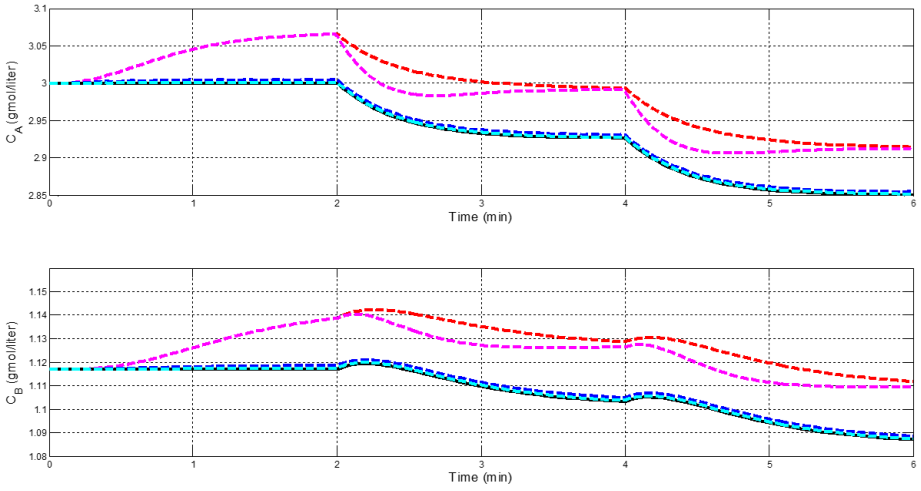

Figure 3.1: Isothermal CSTR with three parallel reactions: Comparison of the outputs C_A and C_B with different frequencies of resetting extended states during simulation

The dashed red lines are the results predicted by Carleman Linearization without resetting extended states. The integration errors accumulate as time proceeds as is shown on Figure 3.1. Resetting the extended states every $2\ min$ yields the magenta lines that are tracking the reference prediction better than the dashed red lines, but the simulation errors are still large.

The dashed dark blue lines are the prediction by Carleman Linearization-based model with a resetting interval of $0.2\ min$. That means during the simulation, the extended states are reset 10 times evenly within each control interval of $2\ min$. The profiles of C_A and C_B are tracking the numerical simulation results expressed by the solid black lines with minor differences. We reset the extended states more frequently and set the resetting interval at $0.1\ min$. This means the extended states are reset 20 times evenly per control interval. The results presented by the dashed light blue lines are exactly tracking the numerical simulation. One notable detail is the forward difference time step of ode45 selected automatically by Matlab is $0.05\ min$ in this case. This shows Carleman Linearization-based model is less demanding in computation while achieving the same accuracy. This idea has also been reported in [20].

This example indicates the worst case scenario that the nominal operating condition is unknown and large integration errors can accumulate as we anticipate future states. In terms of the design of MPC controllers, simulation errors will lead to unavoidable influence on optimization and degrade the controller perfomance. Resetting extended states compensates for this loss and reduces integration errors.

Chapter 4 — Application and Discussion

4.1 Comparison Between the Proposed MPC and Nonlinear MPC

To illustrate the applicability and computational efficiency of the proposed Carleman Linearization-based MPC formulation, a nonlinear jacketed CSTR is used as a case study example.

In the CSTR jacketed by coolant, there is an exothermic first-order reaction. The dynamic process can be described with two ODEs:

$$\dot{C_A} = \frac{q}{V}(C_{Af} - C_A) - k_0 \ \exp(-\frac{E}{R\,T_R})\ C_A$$
$$\dot{T_R} = \frac{q}{V}(T_f - T_R) - \frac{\Delta H}{\rho\, C_p}\ k_0 \ \exp(-\frac{E}{R\,T_R})\ C_A + \frac{U\,A}{V \rho\, C_p\, (T_c - T_R)}$$

The two state variables are the concentration of reactor contents C_A and the reactor temperature T_R. The control input is the coolant temperature T_c in the jacket. Table 4.1 is a list of the nominal operating conditions. The above system is nonlinear around the nominal operating condition. Any pertubation in the parameters may cause large and potentially unstable oscillations. Figure 4.1 shows the open loop response of the system to a $-10\ \%$ perturbation in the feeder concentration C_{Af}. The unstable oscillations grow larger as the operating process proceeds on.

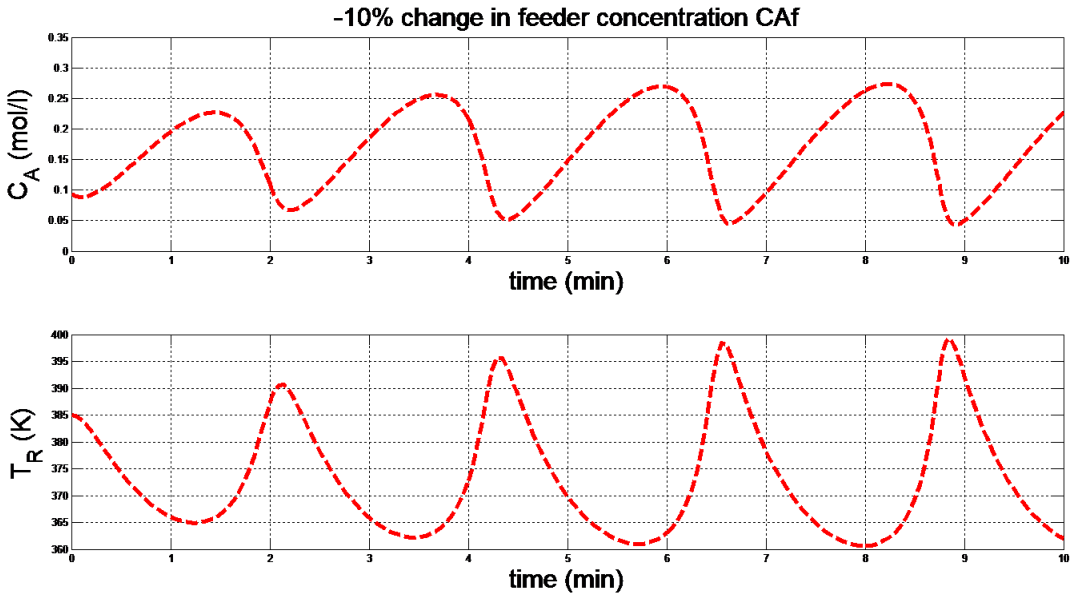

Figure 4.1: The red dashed line shows the response of the open-loop system when there is $-10\ \%$ perturbation in the feeder concentration C_{Af}.

Open-loop Unstable CSTR Parameters		
Parameter	Description	Value
q	Feed Flow Rate	$100\ liter/min$
C_{Af}	Feed Concentration of A	$1\ gmol/liter$
T_f	Feed Temperature	$350\ K$
V	Reactor Volume	$100\ liter$
UA	Heat Transfer Coefficient	$5 \times 10^4\ J/(min \cdot K)$
k_0	Exponential Factor	$7.2 \times 10^{10}\ min^{-1}$
E/R	Reduced Activation Energy	$8750\ K$
ΔH	Heat of Reaction	$-5 \times 10^4\ J/mol$
ρ	Density of Reactor Contents	$1000\ g/liter$
C_p	Heat Capacity of Reactor Contents	$0.239\ J/(g \cdot K)$
T_c	Coolant Temperature	$311.1\ K$
C_A	Nominal Concentration of Reactor Contents	$9.3413 \times 10^{-2}\ gmol/liter$
T_R	Reactor Temperature	$385\ K$

Table 4.1: Open-loop Unstable CSTR Parameters

4.1.1 Formulation and Tuning Parameters

The proposed method reformulates the optimal control problem as optimizing the piece-wise constant control inputs over a finite prediction horizon. The sensitivities of the cost function to the control inputs are provided to facilitate the searching algorithm.

A controller is designed to regulate the system at a reactor temperature of $T_R = 385\ K$ under the disturbance of $-10\ \%$ perturbation in the feeder concentration C_{Af}. We performed 4th order Carleman Linearization to represent the nonlinear system in a bilinear expression. The action horizon is set at $t = 0.03\ min$. We reset the extended states at the end of each action horizon to ensure accurate simulation. The prediction horizon is chosen at $N = 8$ and the control horizon at $N_c = 4$. In this way, we assure stablity of the system without a constraint on the terminal state.

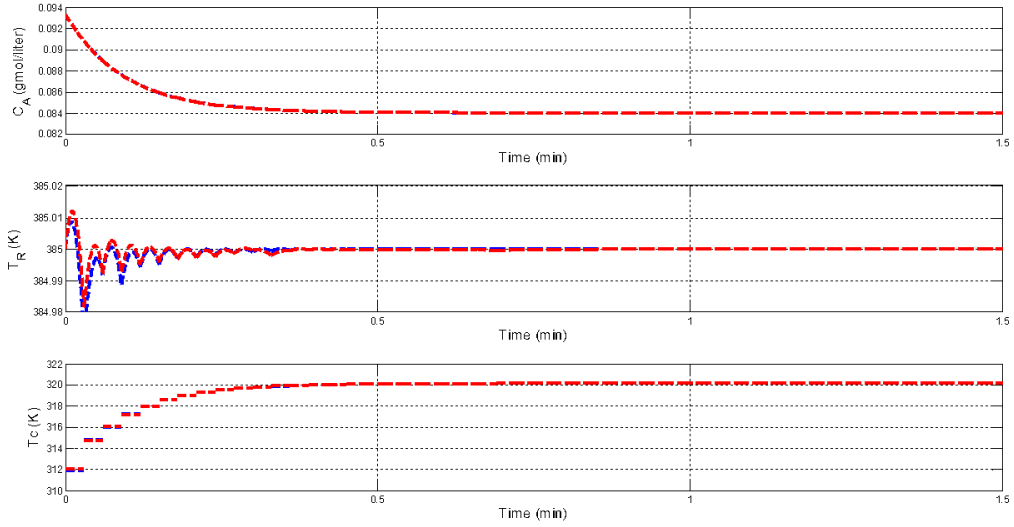

Figure 4.2: The solid blue lines show the optimal policy results of Nonlinear MPC under the disturbance of $-10\ \%$ change is the feeder concentration C_{Af}. The dashed red lines are the optimal policy results of the proposed MPC formulation.

The original cost function we try to minimize is in the following quadratic form

$$\sum_{i=1}^{N} \int_{T_{i-1}}^{T_i} (x^T Q x + u^T R u)\, \mathrm{d}t$$

where Q and R are weighting matrices, $Q = Q^T > 0$ and $R = R^T \geq 0$; x and u are the deviations of the states and the control signal from their nominal conditions respectively. Then we reformulate the cost function with extended states $x_\otimes$ and extended control signal $u_\otimes$ based on the original quadratic cost function, and decide the coefficient vectors J_0, J_A and J_B. The dimensions of these vectors are in consistency with the order of extended states $x_\otimes$ and extended control signal $u_\otimes$.

4.1.2 Simulation Results with the Proposed Formulation

Figure 4.2 shows a comparison between the performances of the proposed MPC formulation (dashed red lines) and of Nonlinear MPC (solid blue lines). They both stablize the open-loop unstable system well and regulates the reactor temperate $T_R = 385\ K$ without offsets. As is shown on Figure 4.2, the differences between the optimal control policy and the results of the system are

negligible.

Keeping other conditions identical, using the same Matlab searching function (fmincon) and the same searching algorithm (interior-point algorithm), the proposed method takes 16.1538 s CPU time to calculate the optimal control policy with Intel Core i7-3770 CPU at 3.40GHz, which is 35 % of 45.5661 s that Nonlinear MPC takes. The simulation results in Figure 4.2 demonstrated the proposed MPC formulation is more computationally efficient than Nonlinear MPC when they achieve the same control goals.

4.2 Advantage of Optimizing Action Horizons

We keep the prediction horizon fixed and set the length of each action horizon as a design variable in addition to the vector of control signals. Previously, throughout the operating process of 1.5 min, with a fixed action horizon of 0.03 min, the total number of control actions is fixed at 49. With -10% disturbance in the feeder concentration C_{Af}, the proposed formulation optimizing U and ΔT_i reduces the value of cost function to 0.0314, compared to 0.0502 of the previous BCMPC formulation optimizing only the control actions U, and 0.0703 of the Nonlinear MPC case. The formulation optimzing U and ΔT_i improves the performance of the MPC controller in regulating the system to track the desired trajectory.

4.3 Advantage of Resetting Extended States within each Action Horizon

We present another circumstance under which the control actions can not be switched as frequently as in Figure 4.2. The action horizon, or sampling time for each piece-wise constant control action is $t = 0.3\ min$. We compare the case of resetting extended states at the end of per sampling time, shown by the dashed red lines in Figure 4.4 with the case of resetting extended states 10 times evenly within each action horizon, shown by the dashed light blue lines in Figure 4.4. The optimal policy results show resetting extended states within each action horizon improves the capability of the MPC controller to track the desired trajectory. To further quantify the performance of MPC controllers, Figure 4.5 shows a comparison between the values of cost functions at each action horizon. The improvement of performance is quantified by the reduced values of the cost function.

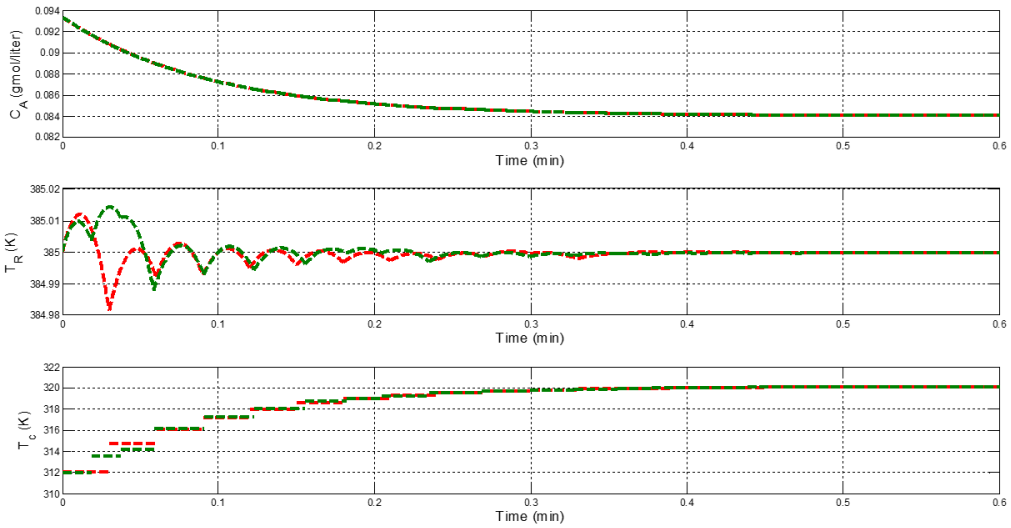

Figure 4.3: The dashed green lines are the optimal policy results with the proposed MPC formulation optimizing both the control actions U and action horizons ΔT_i, compared with the dashed red lines that only optimizes the control actions U.

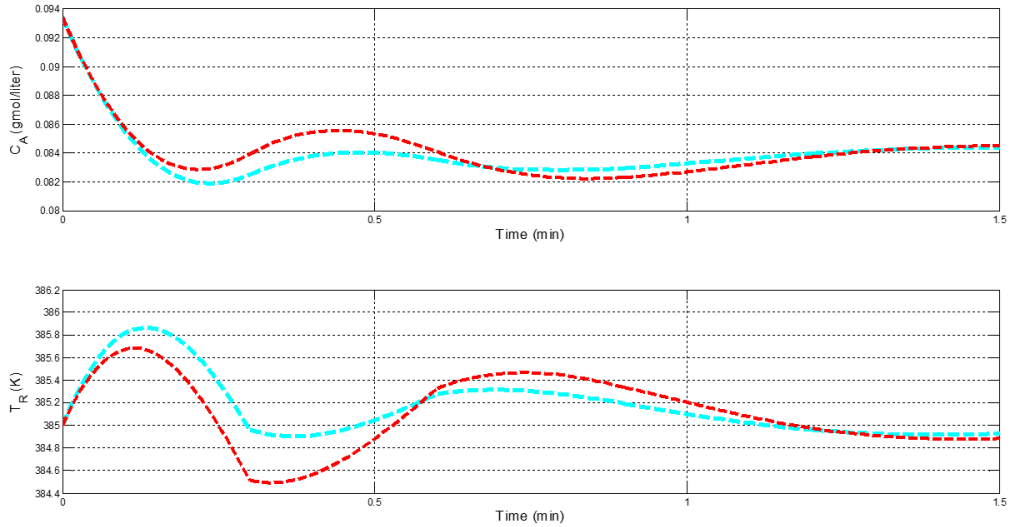

Figure 4.4: The dashed red lines show the optimal policy results of the proposed MPC when the extended states are reset every 0.3 min. The dashed light blue lines are the results when the extended states are reset every 0.03 min.

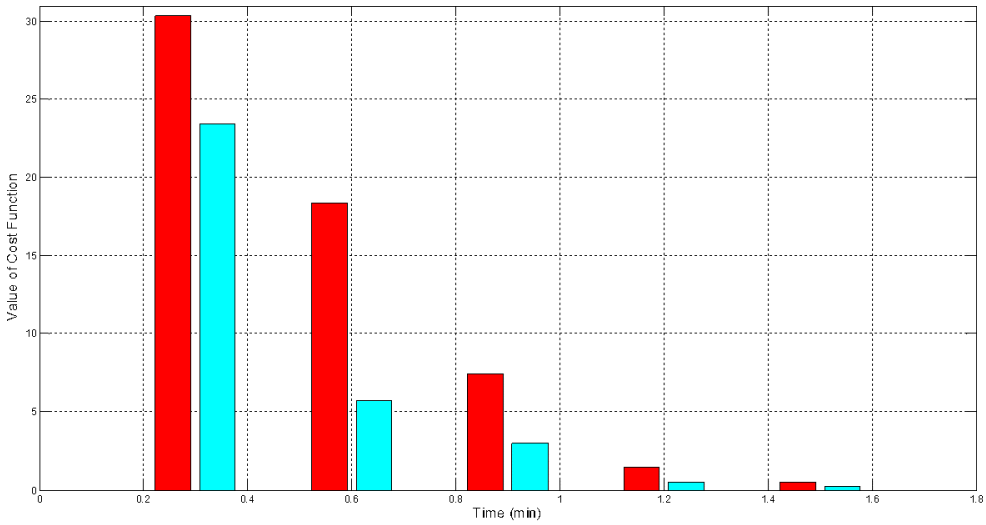

Figure 4.5: The red bars show the value of cost function at each action horizon when the extended states are reset every 0.3 min. The light blue bars are the results when the extended states are reset every 0.03 min.

Chapter 5 — Conclusion and Future Plan

5.1 Conclusion

This thesis proposed a computationally efficient method to reformulate general Nonlinear MPC controller design methodology into a Carleman Linearization-based MPC formulation. Identical to traditional MPC, it converts an optimal control problem to a receding horizon control one with dynamic constraints and performance criteria satisfied. The nonlinear dynamic process is modeled with a bilinear representation. This enables analytical anticipation of future states and analytically providing the sensitivity of the cost function to the control signals as the searching gradient. Consequently, the computation of optimal control policy is accelerated and the computational delay is removed. Resetting extended state is performed in both analytical anticipation of future states and sensitivity calculation to compensate for the simulation errors caused by Carleman Linearization. The action horizons, or sampling times are designed as manipulated variables in addition to the control signals to improve the performance of controllers designed by the proposed method.

The proposed method resembles both the collocation method and the shooting method. Because the proposed formulation discretizes the states of the system explicitly in time and calculate the sensitivity of the control signals analytically. In addition, the proposed method formulates the states as nonlinear functions of the control signals and thus it releases the optimization problem from equality constraints and reduces the number of design variables.

5.2 Future Plan

In the next step of our research, we will propose an advanced reformulation of Carleman Linearization-based Nonlinear MPC with a combination of Advanced-step Nonlinear MPC methodology published in [28] [29] [30]. Using Nonlinear Programming (NLP) sensitivity analysis to find approximations and to update solutions on-line enables this reformulation of Nonlinear MPC increased capability of dealing with measurement noises and model mismatches. It is suitable to apply Advanced-step Nonlinear MPC methodology to the method proposed in this thesis due to its bilinear formulation of the control problem and its formulation of sensitivity calculation.

The proposed formulation of Advanced-step Carleman Linearization-based Nonlinear MPC will be applied to a complex chemical system with large state-space. Our goal is to prove the proposed removes computational delay while it achieves the same stability properties as traditional Nonlinear MPC controllers do.

More recently, economic-oriented MPC (EMPC) has started to gain popularity. The primary difference of EMPC from traditional MPC is its formulation orientation towards minimizing economic costs, which naturally put more emphasis on the process paths through directly effecting economic performance [31] [32] [33]. The formulations of economic cost functions are generally non-quadratic contrary to traditional MPC, but there are formulations to guarantee stability and to improve numerical performance, including adding quadratic regularization terms in the economic cost function and using Lyapunov functions. Carleman Linearization is expected to demonstrate significant advantages in the formulation.

Bibliography

[1] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Comput. Chem. Eng.*, vol. 23, no. 4-5, pp. 667-682, May 1999.

[2] D. Q. Mayne, J. B. Rawlings, C. V Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica* , vol. 36, no. 6, pp. 789-814, Jun. 2000.

[3] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control. Eng. Pract.*vol. 11, pp. 733-764, 2003.

[4] A. Armaou and A. Ataei, "Piece-wise constant predictive feedback control of nonlinear systems," *J. Process Control*, vol. 24, no. 4, pp. 326-335, Apr. 2014.

[5] Y. Fang and A. Armaou,"Nonlinear Model Predictive Control Using a Bilinear Carleman linearization-based Formulation for Chemical Processes," *American Control Conference*, pp. 5629-5634, 2015.

[6] N. Hashemian and A. Armaou,"Fast Moving Horizon Estimation of Nonlinear Processes Via Carleman Linearization," *American Control Conference*, pp. 3379-3385, 2015.

[7] R. Huang, L. T. Biegler, and S. C. Patwardhan, "Fast Offset-Free Nonlinear Model Predictive Control Based on Moving Horizon Estimation," *Ind. Eng. Chem. Res.*, vol. 49, no. 17, pp. 7882-7890, Sep. 2010.

[8] C. Garcia, D. Preti, and M. Morari," Model Predictive Control: Theory and Practicea Survey," *Automatica*, vol. 25, no. 3, pp. 335-348, 1989.

[9] J. B. Rawlings. "Tutorial overview of model predictive control." *IEEE Control Syst. Mag.*, vol. 20, no. 3, pp. 38-52. 2000.

[10] B. Friedland, Control System Design - An Introduction to State-Space Methods, McGraw Hill, Quebec, Canada, 2009.

[11] R. Lopez-Negrete, F. J. DAmato, L. T. Biegler, and A. Kumar, "Fast nonlinear model predictive control: Formulation and industrial process applications," *Comput. Chem. Eng.*, vol. 51, pp. 55-64, Apr. 2013.

[12] P. Kokotovi and M. Arcak, "Constructive nonlinear control: a historical perspective , *Automatica*, vol. 37, no. 5, pp. 637-662, 2001.

[13] P. Mhaskar, N. H. El-farra, and P. D. Christofides, "Predictive Control of Switched Nonlinear Systems With Scheduled Mode Transitions, *IEEE Trans. Autom. Contr.*, vol. 50, no. 11, pp. 1670-1680, 2005.

[14] P. Mhaskar, N. H. El-Farra, and P. D. Christofides, "Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control, *Syst. Control Lett.*, vol. 55, no. 8, pp. 650-659, Aug. 2006.

[15] F. Allgwer and A. Zheng, Nonlinear Model Predictive Control, vol 26. Basel; Boston: Birkhuser Verlag, 2000.

[16] B. W. Bequette, "Model Predictive Control" in *Process Control: Modeling, Design, And Simulation*, Upper Saddle River, N.J: Prentice Hall PTR, 2003, ch. 16, pp. 487-520.

[17] L. T. Biegler, "Advances in nonlinear programming concepts for process control," *J. Proc. Cont.*, vol. 8, no. 5-6, pp. 301-311, 1998.

[18] W. H. Steeb and F. Wilhelm, "Non-Linear Autonomouc Systems of Differential Equations and Carleman Linearizaiton Procedure," *J. Math. Anal. Appl.*, vol. 77, no. 2,pp. 601-611, 1980.

[19] W. H. Steeb," A Note On Carleman Linearization," *Phys. Lett. A.*, vol. 140, no. 6,pp. 336-338, Oct. 1989.

[20] S. A. Svoronos, D.Papageorgiou and C. Tsiligiannis"Discretization of Nonlinear Control Systems via the Carleman Linearization," *J. Process Control*, vol. , no. , pp. 678-685, 1994.

[21] V. Hatzimanikatis, G. Lyberatos, S. Pavlou, and S. A. Svoronos, "A method for pulsed periodic optimization of chemical reaction systems," *Chem. Eng. Sci.*, vol. 48, no. 4, pp. 789-797, 1993.

[22] M. Rodriguez and D.Perez, "First Principles Model Based Control," in *European Symposium on Computer Aided Process Engineering-15*, 2005©Elserier Science B.V.

[23] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides, "Solution of a Class of Multistage Dynamic Optimization Problems. 1.Problems without Path Constraints," *Ind. Eng. Chem. Res.*, vol. 33, no. 9, pp. 2111-2122, Sep. 1994.

[24] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides, "Solution of a Class of Multistage Dynamic Optimization Problems. 2.Problems with Path Constraints," *Ind. Eng. Chem. Res.*, vol. 33, no. 9, pp. 2123-2133, Sep. 1994.

[25] S. E. Tuna, M. J. Messina, and a. R. Teel, "Shorter horizons for model predictive control," *Am. Control Conf.*, vol. 19, no. 4, pp. 678-685, Apr. 2006.

[26] A. R. Teel, "Control Is the Stability Robust Part I Asymptotic Stability Theory," pp. 3-27, 2004.

[27] B. W. Bequette, "Isothermal Chemical Reactor" in *Process Control: Modeling, Design, And Simulation*, Upper Saddle River, N.J: Prentice Hall PTR, 2003, pp. 605-617.

[28] V. M. Zavala and L. T. Biegler, "The advanced-step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, no. 1, pp. 86-93, Jan. 2009.

[29] X. Yang and L. T. Biegler, "Advanced-multi-step nonlinear model predictive control," *J. Process Control*, vol. 23, no. 8, pp. 1116-1128, Sep. 2013.

[30] R. Huang, V. M. Zavala and L. T. Biegler, "Advanced step nonlinear model predictive control for air seperation units," *J. Process Control*, vol. 19, no. 4, pp. 678-685, Apr. 2009.

[31] R. Huang and L. T. Biegler, "Economic NMPC for energy intensive applications with electricity price prediction", in *11th International Symposium on Process Systems Engineering.*, Singapore.,2012, pp.1612-1616

[32] M. Heidarinejad, J. Liu, and P. D. Christofides, "Economic model predictive control of switched nonlinear systems," *Syst. Control Lett.*, vol. 62, no. 1, pp. 77-84, Jan. 2013.

[33] R. Amrit, J. B. Rawlings, and L. T. Biegler, "Optimizing process economics online using model predictive control," *Comput. Chem. Eng.*, vol. 58, pp. 334-343, Nov. 2013.