INCREMENTAL MODEL PREDICTIVE CONTROL SYSTEM DESIGN AND IMPLEMENTATION USING MATLAB/SIMULINK

By

XIN LIN

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2013

© 2013 Xin Lin

To all chemical engineers

ACKNOWLEDGMENTS

The author is sincerely thankful to Prof. Oscar D. Crisalle, *Distinguished Teaching Scholar and Professor* in the Chemical Engineering Department of University of Florida, for helpful advice throughout the stage of this work and doctoral candidates M. Rafe Biswas and Shyam P. Mudiraj for their valuable feedback on the revisions of this thesis .

TABLE OF CONTENTS

page

ACKNOWLEDGEMENTS 4

LIST OF TABLES 7

LIST OF FIGURES 8

ABSTRACT 10

CHAPTER

1 INTRODUCTION 11

1.1 Background 11
1.1.1 Integral Controller and Offset-free Performance 11
1.1.2 Overview of MPC 12
1.2 Description of Contents 14

2 LITERATURE REVIEW 16

2.1 Discrete Integral Controller 16
2.2 Standard MPC Controller 17
2.3 Offset Performance in MPC Control Systems 20
2.4 Offset-free Performance Considering Inconstant Setpoint Tracking 20
2.5 Challenges 22

3 CONTROL SYSTEM DESIGN AND IMPLEMENTATION USING DISCRETE INTEGRAL CONTROLLER 24

3.1 Integral Controller Structure 24
3.2 Offset-free Performance Analysis 25
3.3 Control Validation 27
3.3.1 Example 1: Discrete System from Ogata [1] 27
3.3.2 Example 2: CSTR Plant from Seborg [2] 31
3.3.2.1 Modeling equations 31
3.3.2.2 Linearization of the system 33
3.3.2.3 Offset-free performance 35
3.3.2.4 Control Performance on the linear continuous and nonlinear CSTR systems 36

4 CONTROL SYSTEM DESIGN AND IMPLEMENTATION USING INCREMENTAL MPC 41

4.1 Basic Equations for Incremental MPC 41
4.2 Control Validation 44
4.2.1 Example 1: CSTR Plant from Seborg [2] 44
4.2.2 Example 2: Quadruple Tank System 51

5 CONTROL SYSTEM DESIGN AND IMPLEMENTATION USING INTEGRAL MPC . . . 55

5.1 Basic Equations for Integral MPC . . . 55
5.1.1 First Integral State $z(k)$ . . . 55
5.1.2 Second Integral State $w(k)$ . . . 57
5.1.3 Control Law for the Integral MPC Controller . . . 58
5.2 Control Validation . . . 58

6 CONCLUSIONS AND FUTURE WORK . . . 63

6.1 Conclusions . . . 63
6.2 Future Work: Constrained Incremental MPC . . . 63

APPENDIX

A DERIVATION OF K_1 AND K_2 FOR THE DISCRETE INTEGRAL CONTROLLER . . . 66

REFERENCES . . . 70

BIOGRAPHICAL SKETCH . . . 71

LIST OF TABLES

Table page

3-1 Plant parameters of the CSTR model 34

4-1 Parameter values for the quadruple tank system 52

4-2 Initial values of the quadruple tank system 52

LIST OF FIGURES

Figure page

1-1 A typical block diagram of the closed-loop system with a state feedback integral controller. 12

1-2 A block diagram of an MPC controller. 14

3-1 Closed-loop MATLAB/Simulink model from Ogata's book, two different discrete controllers(controller implemented by connected SIMULINK blocks and by S-Function) are used 30

3-2 Offset-free control performance of the closed-loop system with a discrete integral controller. The model is example 6-12 in Ogata [1]. 32

3-3 The diagram of CSTR system in the book 33

3-4 Offset-free performance of the linear discrete closed-loop CSTR system. There are step changes in setpoint, state disturbance and output disturbance at different time. 37

3-5 Control performance on the linear continuous CSTR system using identical step changes in setpoint, state disturbance and output disturbance to those graphs in figure 3-4 39

3-6 Control performance on the nonlinear CSTR system using the identical step changes in setpoint, state disturbance and output disturbance as subsection 3.3.2.3. 40

4-1 Control diagram of a closed-loop system with the incremental MPC controller 45

4-2 MATLAB/Simulink diagram of the closed-loop system with the incremental MPC controller 47

4-3 Simulation results for the linear discrete CSTR system with the incremental MPC controller 48

4-4 Simulation results for the linear continuous CSTR system with incremental MPC controller 49

4-5 Simulation results for the nonlinear CSTR system with incremental MPC controller 50

4-6 Process diagram for the quadruple tank system from Åkesson [3]. . . . 52

4-7 Simulation results of incremental MPC on the linearized quadruple tank system. The control variable with its setpoint which is presented in the top graph is the level for tank 1. Two input variables denoted as u_1 and u_2 are shown in the bottom graph. 54

5-1 Closed-loop simulation result for the linear discrete CSTR system with an integral MPC controller. The CSTR is given by subsection 3.3.2 and the step changes are presented in subsection 3.3.2.3 59

5-2 Closed-loop simulation result for the linear continuous CSTR system with an integral MPC controller. The CSTR is given by subsection 3.3.2 and the step changes are presented in subsection 3.3.2.3 61

5-3 Closed-loop simulation result for the nonlinear CSTR system with an integral MPC controller. The CSTR is given by subsection 3.3.2 and the step changes are presented in subsection 3.3.2.3 62

Abstract of a Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

INCREMENTAL MODEL PREDICTIVE CONTROL SYSTEM DESIGN AND IMPLEMENTATION USING MATLAB/SIMULINK

By

Xin Lin

May 2013

Chair: Oscar. D. Crisalle
Major: Chemical Engineering

The integral and model predictive controller (MPC) drive controlled outputs to their desired targets, and this thesis addresses the problem of integral controller, incremental and integral MPC when tracking the constant or inconstant references. Design and implementation of the MPC under MATLAB/Simulink environment are discussed both in incremental and integral form. Also one CSTR example is presented to compare the control performances among different integral controller and MPCs.

CHAPTER 1
INTRODUCTION

This chapter is a guide to the topics covered by this thesis. The objectives of this work include the design and implementation of integral and MPC controllers using S-functions under the MATLAB/Simulink environment. Verification and comparison of the control performance among different controllers are presented.

Several controllers and the challenges associated with their design are described in Subsection 1.1. An example for the plant and its detailed information is provided in Subsections 1.2 and 1.3, and the organization of this thesis is explained in the description of contents section.

1.1 Background

1.1.1 Integral Controller and Offset-free Performance

A closed-loop control diagram of an integral controller is shown in figure1-1. The integral controller utilizes the plant states and plant outputs to calculate its output. Such information can be directly measured or estimated by using an observer. The design of an integral controller is simple and straightforward. Since it uses an integral state, it can track constant setpoints without offset. If all the closed-loop poles are placed properly and the plant model is accurate enough, the integral controller achieves the goals of both setpoint tracking and disturbance rejection.

A disadvantage of the integral controller is that when there is strong interaction among different control loops, its performance will degrade significantly. Moreover, since it is designed based on a linear model, it may fail to obtain satisfactory control performance on the plant if there is serious model mismatch or nonlinearity inside the plant.

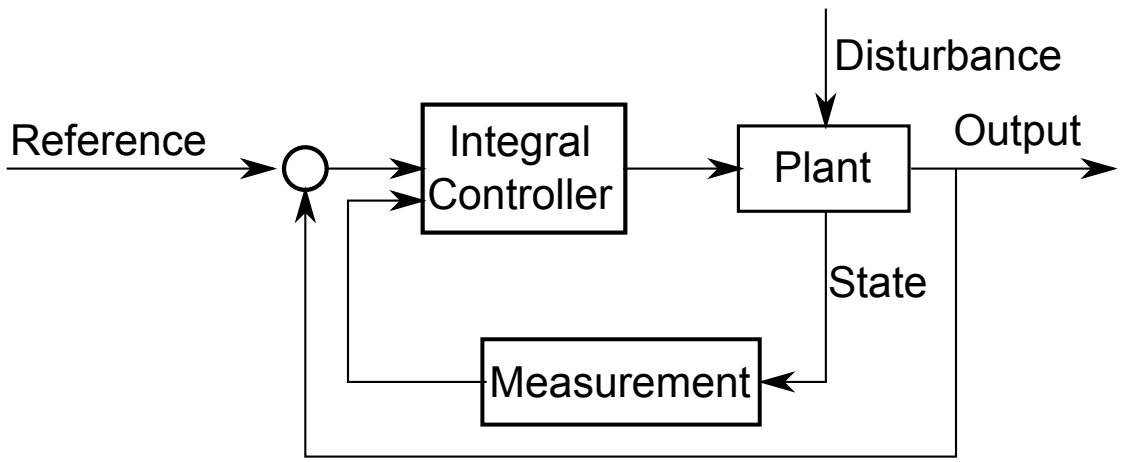

Figure 1-1. A typical block diagram of the closed-loop system with a state feedback integral controller.

1.1.2 Overview of MPC

MPC is an optimal controller based on real-time numerical optimization. A typical MPC control diagram is given in figure 1-2. The plant output is predicted by using an estimated system model. The plant input is optimized at each time instantance according to penalty function and constraints. The main ideas of MPC originally come from a computational technique used to improve control performance in process industries. Since then, predictive control has became the most widespread advanced control strategy in chemical engineering. An MPC controller can achieve desired control performance in large-scale multivariable systems, and provide a systematic method of dealing with states and inputs constraints with simple design and tuning.

The general goal of MPC is to calculate a trajectory of future manipulated variable u to optimize the future behaviour of the plant output y. The optimization is carried out within a limited time instance by using the plant information at the start of the time interval.

There are three fundamental concept in the design of MPC. The first is how to predict the future states and outputs (model); the second is the way on how to obtain the current information of the plant (measurement) and the third is the approach on the implementation of future activities (realization of control). The key issues in the design are:

1. The time interval for the design is a constant;
2. People need to have access to the current states before the control design;
3. People take the constraints into consideration, and the optimization is performed in real-time with a time window that moves foward and with the latest plant information available.

Other concepts that are used frequently in the design of MPC are the following: the moving horizon window, prediction horizon, receding horizon control, and control objective. They are discussed here as following:

1. The moving horizon window is referred to the time-dependent window from an arbitrary time t_i to $t_i + T_p$. The length of the window T_p is a constant. However, t_i , which is the beginning of the window, depends on time and increases as time evolves.
2. The prediction horizon determines how long into the future states and outputs are to be predicted for. This parameter is defined as the length of the moving horizon window, T_p .
3. Receding horizon control is used to describe the control strategy that although the optimal series of future controller outputs are calculated within the moving horizon window, the actual input to the plant is only the first value of the series, and the rest of the controller outputs are discarded.
4. For all processes, people need the state information at time t_i to predict the future behavior of the system. This information is defined as $x(t_i)$ which is a vector containing the current process data, and is either directly measured or estimated.
5. A model describing the dynamics of the system is extremely important in predictive control. A well-designed dynamic model can predict the future performance of the system accurately.

6. To reach the best decision, a mathematical criterion is used to describe the control objective. The objective is usually presented as a function based on the difference between the desired and the actual responses. This objective function is often called the penalty or cost function, J, and the optimal controller output is obtained by minimizing the penalty function within the given optimization window.

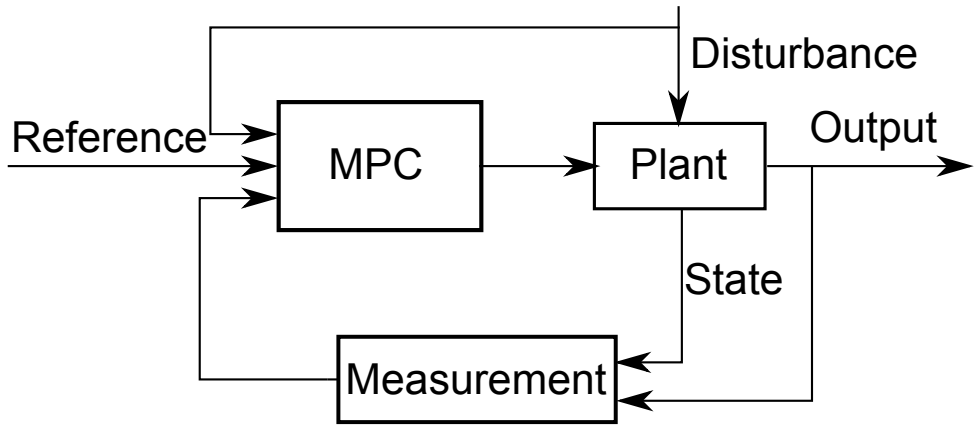

Figure 1-2. A block diagram of an MPC controller.

1.2 Description of Contents

The thesis material has evolved at University of Florida over the last one and a half years. The thesis is divided into five chapters that address the design and implementation of the integral and IMPC controllers.

Chapter 1 provides an introduction to this thesis, including the brief description about discrete integral controller, and the MPC control method.

Chapter 2 is concerned with a literature review. The conditions for offset-free performance with discrete controllers and MPC controllers obtained by previous researchers are provided and discussed. Several disturbance models used by the pioneers to eliminate the influence of existing disturbances and plant model mismatch are presented and discussed.

Chapter 3 through 5 addresses the analysis and design of discrete integral controllers and IMPC controllers under the requirement of offset-free performance for both constant and non-constant setpoint. Simulations of the plant system are used to verify the theory discussed in this part.

Chapter 6 presents conclusions and recommendations for future work.

Finally one appendix includes the controller gain derivation related to the thesis.

CHAPTER 2
LITERATURE REVIEW

2.1 Discrete Integral Controller

Chemical engineers have used discrete integral controllers for a very long time to achieve offset-free performance. If all the closed-loop poles are properly placed, the integral controller can also reject an unmeasured constant disturbance. Ogata [1] considers a discrete system with state and output equation

$$x(k+1) = A_d x(k) + B_d u(k) \tag{2–1}$$

$$y(k) = C_d x(k) \tag{2–2}$$

$x(k) \in \mathbb{R}^n$ is the state vector, $y(k) \in \mathbb{R}^m$ is output vector and $u(k) \in \mathbb{R}^p$ is the manipulated variable. The matrices $A_d \in \mathbb{R}^{n\times n}$, $B_d \in \mathbb{R}^{n\times p}$ and $C_d \in \mathbb{R}^{m\times n}$ are the state matrix, input matrix and output matrix, respectively. The pair $(A_d,\ B_d)$ is assumed to be controllable. An integral state equation is defined as

$$z(k) = z(k-1) + r(k) - y(k) \tag{2–3}$$

where $r(k) \in \mathbb{R}^m$ is the setpoint vector, $z(k) \in \mathbb{R}^m$ is the integral state vector and the controller output $u(k)$ is given by

$$u(k) = -K_2 x(k) + K_1 z(k) \tag{2–4}$$

For the purpose of design, it is convenient to define the augmented state vector

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}$$

The setpoint is assumed to be a constant or a step function. Under closed-loop condition $u(\infty)$ and $x(\infty)$ exist if the following conditions are satisfied:

1. The pair $(A_d,\ B_d)$ is controllable.

2. The matrix
$$\begin{bmatrix} A_d - I & B_d \\ C_d & 0 \end{bmatrix}$$
is full rank.

3. All the closed-loop poles are properly placed.

Since according to equation (2–12) $u(k)$ makes use of both the feedback state $x(k)$ and the integral state, if $z(\infty)$ exists for a given input, the vectors $x(k)$, $u(k)$, and $z(k)$ will converge to their steady state values denoted $x(\infty)$, $u(\infty)$ and $z(\infty)$ respectively. Then the following equation at the steady state can be obtained from (2–3)

$$z(\infty) \quad = \quad z(\infty) + r(\infty) - y(\infty)$$

which implies that

$$r(\infty) = y(\infty)$$

Therefore, the integral controller can achieve offset-free performance when the setpoint is a constant or a step function.

2.2 Standard MPC Controller

The MPC controller is used in a wide variety of industrial systems, including chemical reactors, automobiles, robots and so on. Rawlings [4] discusses the design of a predictive control system in two parts: prediction and optimization. For prediction, denote the future controller output as

$$u(k_i),\ u(k_i+1),\ \ldots,\ u(k_i+N_c-1),$$

where N_c is the control horizon. Also denoting the future state as

$$x(k_i+1|k_i),\ x(k_i+2|k_i),\ \ldots,\ x(k_i+m|k_i),\ x(k_i+N_p|k_i),$$

where N_p is the prediction horizon that determines the number of the future state people would like to calculate. In general $x(k_i + m|k_i)$ is called the future state at time $k_i + m$ predicted at instant k_i and based on the given system state $x(k_i)$. For a state space model, calculate the future states and outputs as follows

$$\begin{aligned}
x(k_i+1|k_i) &= Ax(k_i) + B\Delta u(k_i) \\
x(k_i+2|k_i) &= Ax(k_i+1|k_i) + B\Delta u(k_i+1) \\
&= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i+1) \\
&\vdots \\
x(k_i+N_p|k_i) &= A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) + \ldots + A^{N_p-N_c}B\Delta u(k_i+N_c-1)
\end{aligned} \tag{2–5}$$

and the future output as

$$\begin{aligned}
y(k_i+1|k_i) &= CAx(k_i) + CB\Delta u(k_i) \\
y(k_i+2|k_i) &= CA^2x(k_i+1|k_i) + CAB\Delta u(k_i+1) + CB\Delta u(k_i+1) \\
&\vdots \\
y(k_i+N_p|k_i) &= CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) + \ldots + CA^{N_p-N_c}B\Delta u(k_i+N_c-1)
\end{aligned} \tag{2–6}$$

Define

$$\begin{aligned}
Y &= [y(k_i+1|k_i)\, y(k_i+2|k_i) \ldots y(k_i+N_p|k_i)]^T \in \mathbb{R}^{N_p p \times p} \\
U &= [u(k_i)\, u(k_i+1) \ldots u(k_i+N_c-1)]^T \in \mathbb{R}^{N_c p \times p}
\end{aligned}$$

the output equation becomes

$$Y = Fx(k_i) + \Phi U \tag{2–7}$$

where

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix} \in \mathbb{R}^{N_p m \times n}$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix} \in \mathbb{R}^{N_p m \times N_c p}$$

For the optimization part, define

$$R_s^T = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} r(k_i)$$

then the penalty function is written as

$$\begin{aligned} J &= (R_s - Y)^T Q (R_s - Y) + U^T R U \\ &= (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2U^T \Phi^T (R_s - Fx(k_i)) \\ &\quad + U^T (\Phi^T \Phi + R) U \end{aligned}$$

where Q and R are the output weight matrix and velocity weight matrix, respectively. These two diagonal matrices are used to tune the desired closed-loop control performance.

To find optimal U that will minimize the penalty function J , calculate the first derivative of J with respect to U

$$\frac{\partial J}{\partial U} = -2\Phi^T (R_s - Fx(k_i)) + 2((\Phi^T \Phi + R)\Delta U). \tag{2–8}$$

The necessary condition for minimal J is that (2–8) equals to zero. Assuming that $(\Phi^T\Phi + R)^{-1}$ exists, then U can be sovled to minimize J , shown as (2–9)

$$U = (\Phi^T\Phi + R)^{-1}\Phi^T(R_s - Fx(k_i)) \tag{2–9}$$

Define

$$\bar{R}_s = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T$$

then U can be determined by the following equation

$$U = (\Phi^T\Phi + R)^{-1}\Phi^T(\bar{R}_s r(k_i) - Fx(k_i)), \tag{2–10}$$

which is the control law of the standard MPC controller.

2.3 Offset Performance in MPC Control Systems

For an MPC controller given by figure 1-2, Maeder [5] demonstrates that if the number of mesurements is greater than the number of manipulated variables, not all cntrolled vairables can have offset-free performance. According to Rawlings [4] MPC controllers can achieve offset-free control performance by adding an integral disturbance to the system. This additional disturbance eliminates the mismatch between real system and the model by its corresponding integral action. Since the integration routine is independent of the controller itself, it may cause wind up problems in constrained system.

To solve this problem, one can combine the controller with a disturbance estimator. Then the state equation for the prediction is augmented by the setpoint and disturbance as Rawlings [4] describes. With this improvement MPC controller can solve the wind up problem.

2.4 Offset-free Performance Considering Inconstant Setpoint Tracking

Qian [6] states that the existing control algorithms assume the disturbance and setpoint are constants. Hence, for non-constant disturbances and setpoints

such as ramps and sinusoid functions, these algorithms fail to achieve offset-free performance. Maeder [7][8] provide a generalized approach to track an arbitrary setpoint.

The state and reference equation for a discrete system

$$\begin{aligned} x_r(k+1) &= A_r x_r(k+1) \\ r(k) &= C_r x_r(k) \end{aligned}$$

where $A_r \in R^{n_r \times n_r}$ and $C_r \in R^{n_y \times n_r}$, and the matrix A_r may be unstable. If there is a linear system such that

$$s(k) = C_s x_s(k), \quad x_s(k+1) = J_{\lambda,p} x_s(k), \quad k = 0, 1, \ldots$$

where

$$J_{\lambda,p} = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 1 \\ 0 & 0 & 0 & 0 & \lambda \end{bmatrix} \in \mathbb{R}^{p \times p}$$

is a Jordan block matrix for λ with order p, the closed-loop system with a predictive controller can attain offset-free performance with the following assumptions:

1. The state estimator is stable.
2. The following expansions for y and u exist

$$y = \sum_{i=1}^{m} y_{p_i}^{\lambda_i} \tag{2–11}$$

$$u = \sum_{i=1}^{m} u_{p_i}^{\lambda_i} \tag{2–12}$$

where λ_i is the i-th eigenvalue of A with the multiplicity p_i.

However, this apporach increases the computational cost of the trajectory optimization. Moreover, there is very limited improvement on the offset performance for certain references if the expansions (2–11) and (2–12) above do not exist.

2.5 Challenges

The control law for a discrete integral controller is designed as $u(k) = -K_2 x(k) + K_1 z(k)$. When the system reaches steady state, one can have the following equation

$$u(\infty) = -K_2 x(\infty) + \lim_{k\to\infty} K_1 z(k)$$

which means $\lim_{k\to\infty} K_1 z(k)$ is a constant vector. The problem is that this limit does not ensure that a constant vector $z(\infty)$ exists.

The key is that whether the square matrix K_1 is full rank. From Ackermann's formula,

$$K = \begin{bmatrix} 0 & 0 & \cdots & 0 & I_m \end{bmatrix} \begin{bmatrix} H & GH & \cdots & G^{n-1}H \end{bmatrix}^{-1} \phi(G)$$

where $K = \begin{bmatrix} K_2 & K_1 \end{bmatrix} \in \mathbb{R}^{1\times n}$ and $\phi(G)$ is the characteristic equation of G, where

$$\begin{aligned} G &= \begin{bmatrix} A & 0 \\ -CA & I_m \end{bmatrix} \\ H &= \begin{bmatrix} I_n \\ -C \end{bmatrix} B \end{aligned}$$

Assuming $\begin{bmatrix} A - I_n & B \\ CA & CB \end{bmatrix}$ is invertible, solve for $\begin{bmatrix} K_2 & K_1 \end{bmatrix}$ to get

$$\begin{bmatrix} K_2 & K_1 \end{bmatrix} = \left[K + \begin{bmatrix} 0 & I_m \end{bmatrix} \right] \begin{bmatrix} A - I_n & B \\ CA & CB \end{bmatrix}^{-1}$$

Solve for K_1 to obtain

$$
\begin{aligned}
K_1 &= \left[\begin{bmatrix} 0 & 0 & \cdots & 0 & I_m \end{bmatrix}\begin{bmatrix} H & GH & \cdots & G^{n-1}H \end{bmatrix}^{-1}\phi(G) + \begin{bmatrix} 0 & I_m \end{bmatrix}\right] \\
&= \begin{bmatrix} A - I_n & B \\ CA & CB \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ I_m \end{bmatrix}
\end{aligned} \tag{2–13}
$$

From the above equation it is hard to know whether the matrix K_1 is full rank or not since the invertibility of the matrix $\begin{bmatrix} A - I_n & B \\ CA & CB \end{bmatrix}$ is not ensured.

In Pannocchia [9] , this approach is proven effective for certain squares cases only in Rawlings [10]. Moreover, this approach augments the original state space equations with a integrating disturbance model and hence the computational cost has been significantly increased. Another two improved methods by using the incremental and integral MPC controllers are applied in this thesis in Chapter 4 and 5.

CHAPTER 3
CONTROL SYSTEM DESIGN AND IMPLEMENTATION USING DISCRETE INTEGRAL CONTROLLER

3.1 Integral Controller Structure

A discrete-time, linear, time invariant model

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) \\ x(0) &= x_0 \end{aligned} \tag{3–1}$$

is given, where $x(k) \in \mathbb{R}^n$ is the plant state vector, $y(k) \in \mathbb{R}^m$ is output vector and $u(k) \in \mathbb{R}^p$ is the control vector. Furthermore $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, and $C \in \mathbb{R}^{m\times n}$. The pair $(A,\ B)$ is assumed to be controllable. The purpose is to design an integral state-space controller. The closed-loop control diagram is shown in figure 1-1. The controller output adopted is proposed in Ogata [1] as follows:

$$u(k) = -K_2 x(k) + K_1 z(k) \tag{3–2}$$

where

$$\begin{aligned} z(k) &= z(k-1) + r(k) - y(k) \\ z(0) &= 0 \end{aligned} \tag{3–3}$$

is an integral state belonging to the controller and where the $r(k) \in \mathbb{R}^m$ is the setpoint. $K_1 \in \mathbb{R}^{p\times n}$ and $K_2 \in \mathbb{R}^{p\times m}$ are the controller gains.

Since $u(k)$ is a linear combination of state vectors $x(k)$ and $z(k)$, it is possible to define an augmented vector consisting of $x(k)$ and $u(k)$ instead of $x(k)$ and $z(k)$. Then the following augmented state equation can be obtained.

$$\begin{bmatrix} x(k+1) \\ u(k+1) \end{bmatrix} = \begin{bmatrix} A_d & B_d \\ K_2 - K_2 A_d - K_1 C_d A_d & I_m - K_2 B_d - K_1 C_d B_d \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 \end{bmatrix} r(k+1) \tag{3–4}$$

The output equation of system (3–1) then can be rewritten as

$$y(k) = \begin{bmatrix} C_d & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \tag{3–5}$$

3.2 Offset-free Performance Analysis

The closed-loop poles are determined by the characteristic of the system itself and is independent of the setpoint $r(k)$. Only the eigenvalues of the augmented state matrix are used in placing the poles of the system.

Assume the constant setpoint $r(k)$ is applied to the system such that

$$r(k) = r$$

Then the augmented state equation can be written as

$$\begin{bmatrix} x(k+1) \\ u(k+1) \end{bmatrix} = \begin{bmatrix} A_d & B_d \\ K_2 - K_2 A_d - K_1 C_d A_d & I_m - K_2 B_d - K_1 C_d B_d \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 r \end{bmatrix} \tag{3–6}$$

Assuming that $z(k)$ approaches a constant vector, denoted as, $z(\infty)$, at steady state (5–2) becomes

$$z(\infty) = z(\infty) + r - y(\infty)$$

or

$$y(\infty) = r$$

which indicates offset-free performance. The analysis related to the controllability of the augmented state matrix (A–4) is given in Chapter 7. Assuming that the matrix $\begin{bmatrix} A_d - I_n & B_d \\ C_d A_d & C_d B_d \end{bmatrix}$ is invertible, K_1 and K_2 can be obtained with the following form

$$\begin{bmatrix} K_2 & K_1 \end{bmatrix} = \left[\bar{K} + \begin{bmatrix} 0 & I_m \end{bmatrix}\right] \begin{bmatrix} A_d - I_n & B_d \\ C_d A_d & C_d B_d \end{bmatrix}^{-1}$$

It is also found that if the input $u(k)$ is an m-vector with $m > 1$, there are more than one solutions to the matrix $\bar{K}$ using the pole-placement technique, which means that more than one pair of matrices K_1 and K_2 can be found. In this situation, choose the pair of K_1 and K_2 that obtains the optimal offset-free control performance.

The integral controller designed above can also be used for a plant model combined with a disturbance model. The dynamics and output equation of the disturbance model are shown as

$$x(k+1) = A_d x(k) + B_d u(k) + d(k) \tag{3–7}$$

$$d(k+1) = d(k) \tag{3–8}$$

$$y(k) = C_d x(k) + p(k) \tag{3–9}$$

$$p(k+1) = p(k) \tag{3–10}$$

where $d(k)$ and $p(k)$ cannot be directly measured in most situations.

3.3 Control Validation

3.3.1 Example 1: Discrete System from Ogata [1]

An example here used to show the offset-free performance of the integral controller is the Example 6-12 in Ogata [1]. The discrete system is characterized with discrete matrices

$$A_d = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.12 & -0.01 & 1 \end{bmatrix} \tag{3–11}$$

$$B_d = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and

$$C_d = \begin{bmatrix} 0.5 & 1 & 0 \end{bmatrix}$$

The controllability matrix for the pair$(A_d,\ B_d)$ is

$$
\begin{aligned}
Q_c &= \begin{bmatrix} B_d & A_d B_d & \cdots & A_d^{n-1} B_d \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0.99 \end{bmatrix}
\end{aligned}
$$

which is full rank.

The augmented state and output equations defined in chapter 7 are

$$
\bar{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.12 & -0.01 & 1 & 0 \\ 0 & -0.5 & -1 & 1 \end{bmatrix} \tag{3–12}
$$

$$
\bar{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}
$$

The controllability matrix for the pair $\left(\bar{A},\ \bar{B}\right)$ of the augmented state space system is

$$
\begin{aligned}
\bar{Q}_c &= \begin{bmatrix} \bar{B} & \bar{A}\bar{B} & \cdots & \bar{A}^{n-1}\bar{B} \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0.99 \\ 1 & 1 & 0.99 & 0.86 \\ 0 & -1 & -2.5 & -3.99 \end{bmatrix}
\end{aligned}
$$

which is a full-rank matrix. Then using the method described in Ogata [1], the gain matrix $\bar{K}$ can be obtained

$$\bar{K} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{B} & \bar{A}\bar{B} & \bar{A}^2\bar{B} & \bar{A}^3\bar{B} \end{bmatrix}^{-1} \varphi(\bar{A}) \tag{3–13}$$

where

$$\varphi(\bar{A}) = \bar{A}^4$$

Calculate $\bar{K}$ from (3–13)

$$\bar{K} = \begin{bmatrix} -0.12 & -0.13 & 0.99 & 1 \end{bmatrix} \tag{3–14}$$

and thus

$$\begin{aligned} \begin{bmatrix} K_2 & K_1 \end{bmatrix} &= \left[\bar{K} + \begin{bmatrix} 0 & 1 \end{bmatrix}\right] \begin{bmatrix} A_d - I_3 & B_d \\ C_d A_d & C_d B_d \end{bmatrix}^{-1} \\ &= \begin{bmatrix} -0.12 & 0.3233 & 2 & 0.6667 \end{bmatrix} \end{aligned}$$

Then obtain the integral and state feedback gains K_1 and K_2 as

$$\begin{aligned} K_1 &= 0.6667 \\ K_2 &= \begin{bmatrix} -0.12 & 0.3233 & 2 \end{bmatrix} \end{aligned} \tag{3–15}$$

Using the control law (3–2) - (3–3) with the gains given by (3–15), offset-free control performance can be achieved.

A closed-loop MATLAB/Simulink model for the example figure is shown in figure 3-1. There are two discrete integral controllers in the diagram labelled as controller 1 and controller 2. All of their parameters are the same except the implementation is different. The first controller is implemented by connected simulink

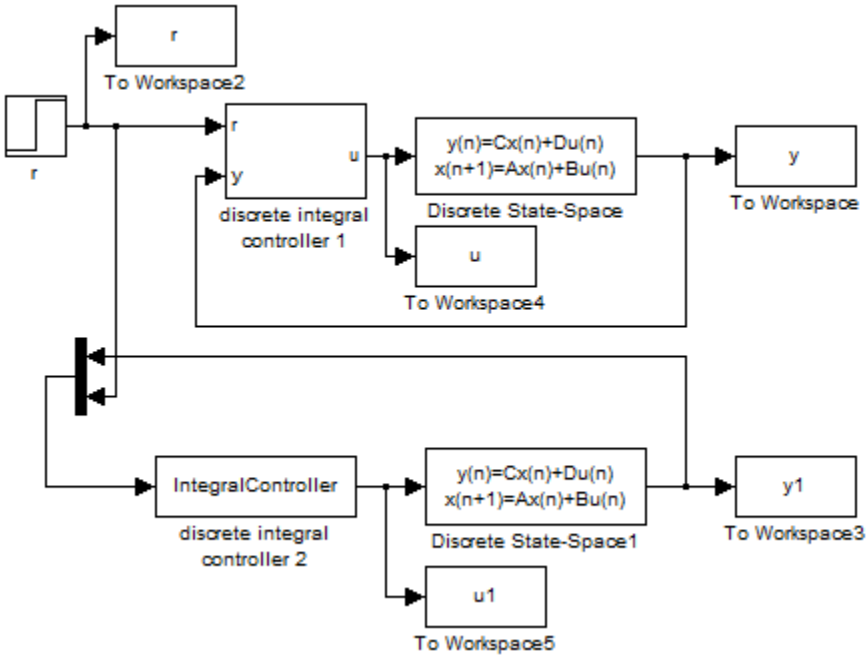

Figure 3-1. Closed-loop MATLAB/Simulink model from Ogata's book, two different discrete controllers(controller implemented by connected SIMULINK blocks and by S-Function) are used

blocks and the second is implemented by an S-function with the approach introduced in MathWorks [11]. They have the identical control performance for the same plant.

The offset-free control performance is shown in figure 3-2. Since these two integral controller have the same control performance, only the performance of the first controller is shown. The top graph in the figure presents the setpoint and plant output as a function of time. The x-axis is the time and the y-axis represents the values for setpoint and plant output. The second graph is the controller output as a function of time.

A constant setpoint $r(k) = 1$ and intitial condition

$$\begin{aligned} x_1(0) &= 1 \\ x_2(0) &= -0.5 \\ x_3(0) &= 0 \\ z(0) &= 1 \end{aligned}$$

Offset-free performance is obtained at $k = 4$, which verifies the offset elimination property of the discrete integral controller.

3.3.2 Example 2: CSTR Plant from Seborg [2]

3.3.2.1 Modeling equations

A continuous stirred-tank reactor (CSTR) as shown in Figure 3-3 is discussed here. The CSTR model is put forward by Seborg [2]. An irreversible, first-order reaction,

$$A \rightarrow B$$

is carried out in the liquid phase in the reactor, and the reactor temperature is controlled by external cooling. It is assumed that the liquid level is a constant. Mass and energy balances lead to the following nonlinear state-space model:

Molar balance equation:

$$V\frac{dC_A}{dt} = q(C_{Ai} - C_A) - Vkc_A \tag{3–16}$$

Energy balance equation:

$$V\rho C\frac{dT}{dt} = wC(T_i - T) + (-\Delta H_R VkC_A) + UA(T_c - T) \tag{3–17}$$

where C_A is the molar concentration, T is the reactor temperature, q is the outlet flow rate, T_c is the coolant liquid temperature.

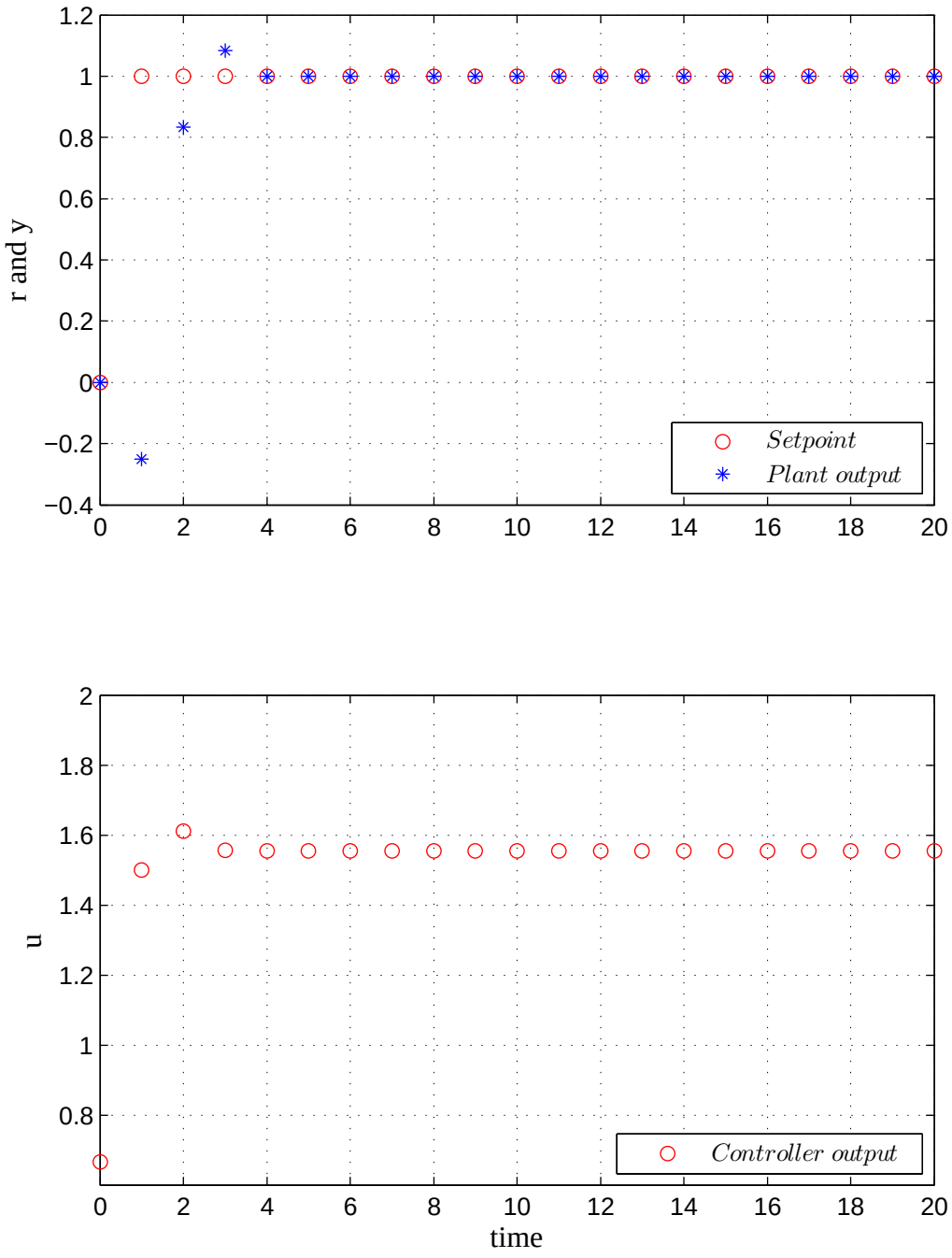

Figure 3-2. Offset-free control performance of the closed-loop system with a discrete integral controller. The model is example 6-12 in Ogata [1].

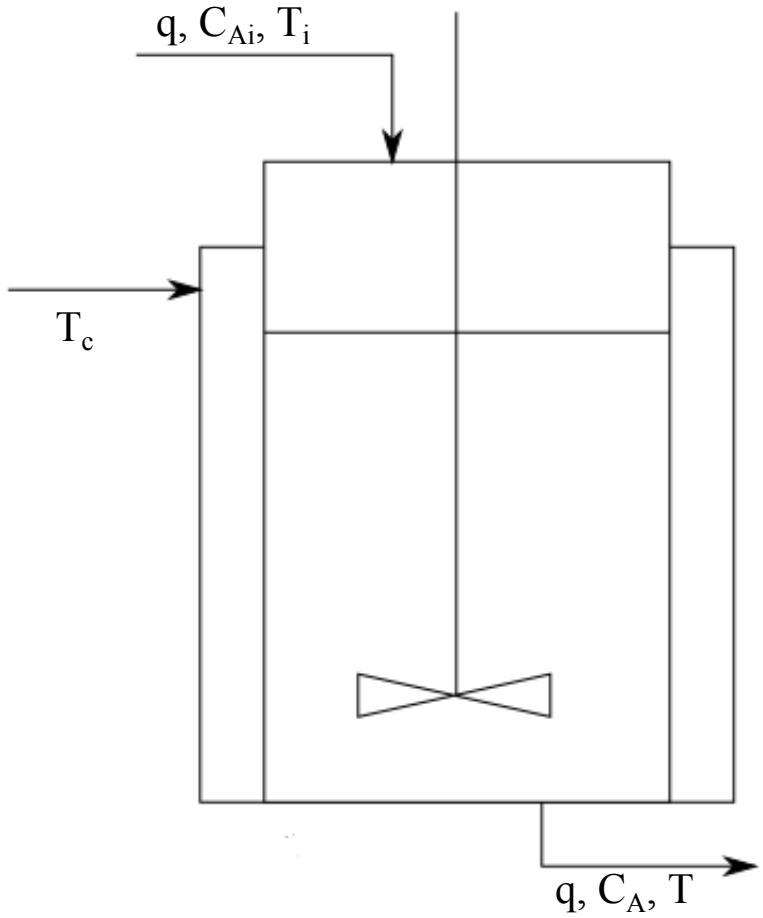

Figure 3-3. The diagram of CSTR system in the book

The controlled variable is defined as the outlet molar concentration C_A. The state variables are the reactor temperature T, and C_A, while the manipulated variables are the inlet molar concentration C_{Ai}, inlet temperature T_i, and the coolant liquid temperature, T_c. Moreover, there are state and output disturbances in the system.

3.3.2.2 Linearization of the system

The open-loop steady-state operating conditions are the following

$$C_{Ai} = 10 mol/L,\ T_i = 298.2K,\ T = 339.4282K,\ C_A = 5.5 mol/L,\ T_c = 298.05K.$$

Using a sampling period time of $1\,\mathrm{min}$, a linearized discrete state-space model is developed in terms of the deviation states, inputs, and outputs

$$x = \begin{bmatrix} T - T^s \\ C_A - C_A^s \end{bmatrix}, \quad u = \begin{bmatrix} C_{Ai} - C_{Ai}^s \\ T_i - T_i^s \\ T_c - T_c^s \end{bmatrix}, \quad y = \begin{bmatrix} T - T^s \\ C_A - C_A^s \end{bmatrix}$$

The parameters needed for process modeling is given in Table 3-1.

Table 3-1. Plant parameters of the CSTR model

Parameter	Nominal Value
q_i/V	$1\ min^{-1}$
T_i	$298.2\ K$
C_{Ai}	$10\ mol/L$
k_0	$34930800\ min^{-1}$
$-\Delta E/R$	$-5963.6\ K$
$UA/(V\rho C_p)$	$0.3\ min^{-1}$
$\Delta H/\rho C_p$	$-11.92\ L{\cdot}K/mol$

After linearization, augment the linear state space equation by state disturbance $d(k)$ and $p(k)$. The CSTR model becomes

$$x(k+1) = A_d x(k) + B_d u(k) + d(k) \tag{3–18}$$

and the linear output equation is given by

$$y(k) = C_d x(k) + D_d u(k) + p(k) \tag{3–19}$$

where

$$A_d = \begin{bmatrix} 1.4765 & 9.7527 \\ -0.2329 & -1.8182 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0 & 1 & 0.3 \\ 1 & 0 & 0 \end{bmatrix}$$

$$C_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad D_d = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

d is the unmeasured state disturbance and p is the output disturbance.

3.3.2.3 Offset-free performance

The controllability matrix of the pair (A_d, B_d) is full-row rank, and as is the controllability matrix of the pair $(\bar{A}, \bar{B})$, where

$$\begin{aligned} \bar{A} &= \begin{bmatrix} A_d & 0 \\ -A_d & I \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.12 & -0.01 & 1 & 0 \\ 0 & -0.5 & -1 & 1 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \bar{B} &= \begin{bmatrix} I_n \\ -C \end{bmatrix} B_d \\ &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \end{aligned}$$

which indicates that the system can achieve offset-free performance with proper gain matrices K_1 and K_2. Related simulation results are given in figure 3-4. Each graph in the figure shows certain variables as a function of time. These variables

include the system input, setpoint and output, state and output diturbances. The controlled variable is the outlet concentration of the CSTR and its setpoint decreases from 0 to -1 at $k = 30\,\mathrm{min}$ in order to examine the control performance under the condition of switching the reaction conversion from a low rate to a relatively high one. There is a step change from 0 to 1 in state disturbance 2 at $k = 100\,\mathrm{min}$ and a change from 0 to 1 in output disturbance 2 at $k = 150\,\mathrm{min}$.

Figure 3-4 shows the offset-free performance for the closed-loop linear discrete CSTR system. The discrete controller designed here can deliver satisfactory performance on both setpoint tracking and disturbance rejection.

3.3.2.4 Control Performance on the linear continuous and nonlinear CSTR systems

Use the same discrete integral controller on the linear continuous and nonlinear CSTR systems. Applying the same step changes in setpoint, state and output disturbances as subsection 3.3.2.3, the system responses are shown in figure 3-5. The graphs for state disturbance and output disturbance are the same as in figure 3-4, so only the graphs on the CSTR input, output and setpoint are given in the figure.

From figure 3-5 it is reasonable to conclude that offset-free control performance for the linear continuous CSTR system can be achieved using the same integral controller which tracks the setpoint and rejects disturbance very well.

Figure 3-6 is obtained while testing controller on the nonlinear system. Using the identical step changes on setpoint and disturbance as subsection 3.3.2.3, only the graphs on the CSTR input, output and setpoint are shown in the figure. The x-axis is the time, and the y-axis represents the CSTR input in the first graph and the output with its setpoint in the second graph.

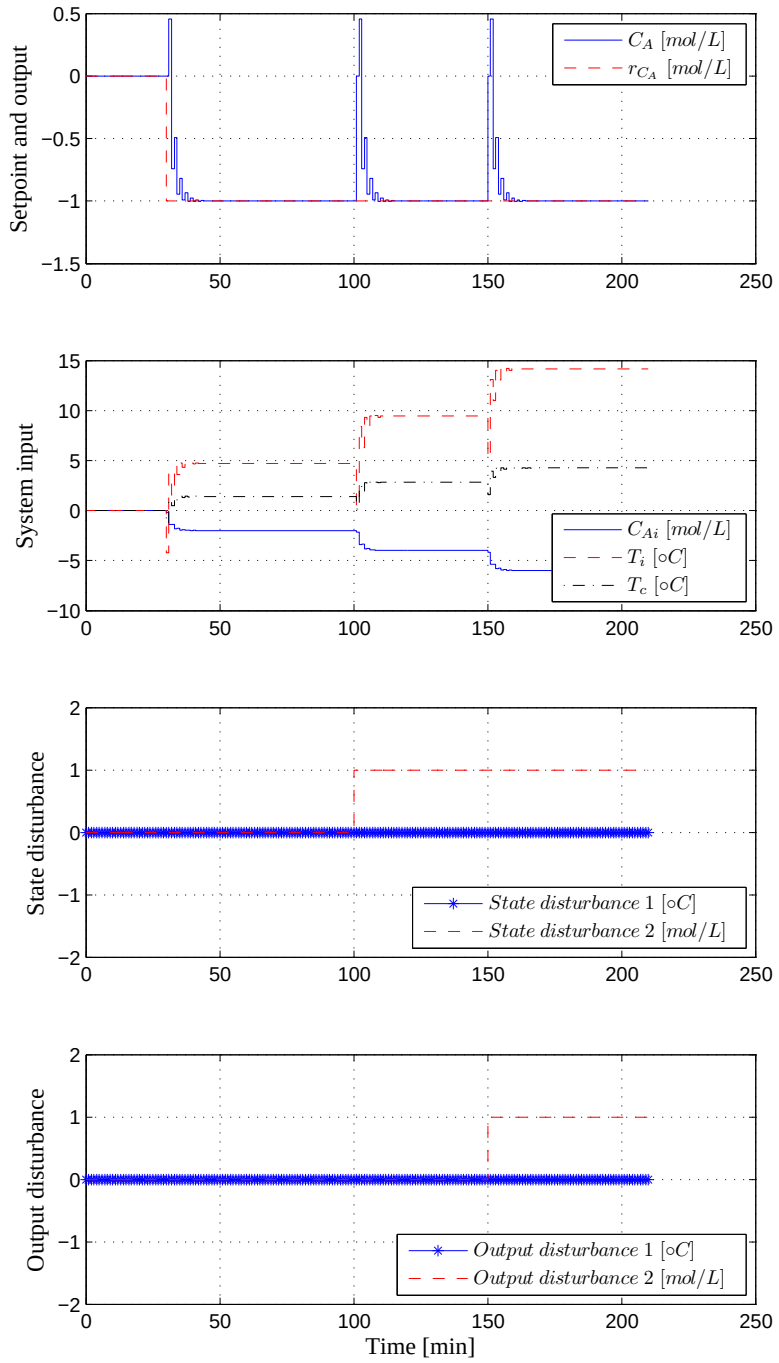

Figure 3-4. Offset-free performance of the linear discrete closed-loop CSTR system. There are step changes in setpoint, state disturbance and output disturbance at different time.

As is shown in figure 3-6, due to model mismatch, although the controller successfully achieves the offset-free control performance, strong oscillation exists in the closed-loop system. For setpoint tracking the transient process takes about 50 min, which is too long for the system to be stabilized.

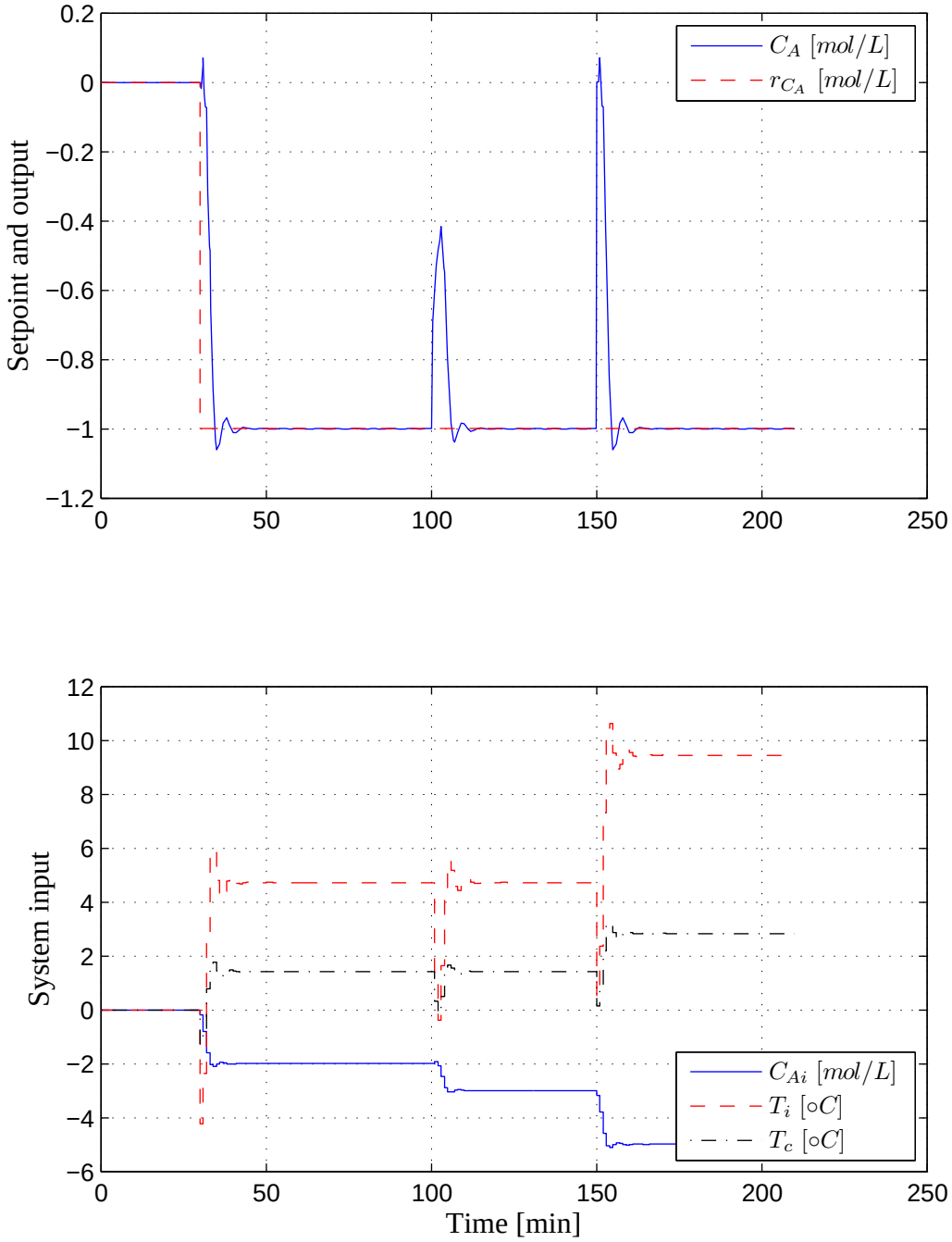

Figure 3-5. Control performance on the linear continuous CSTR system using identical step changes in setpoint, state disturbance and output disturbance to those graphs in figure 3-4 .

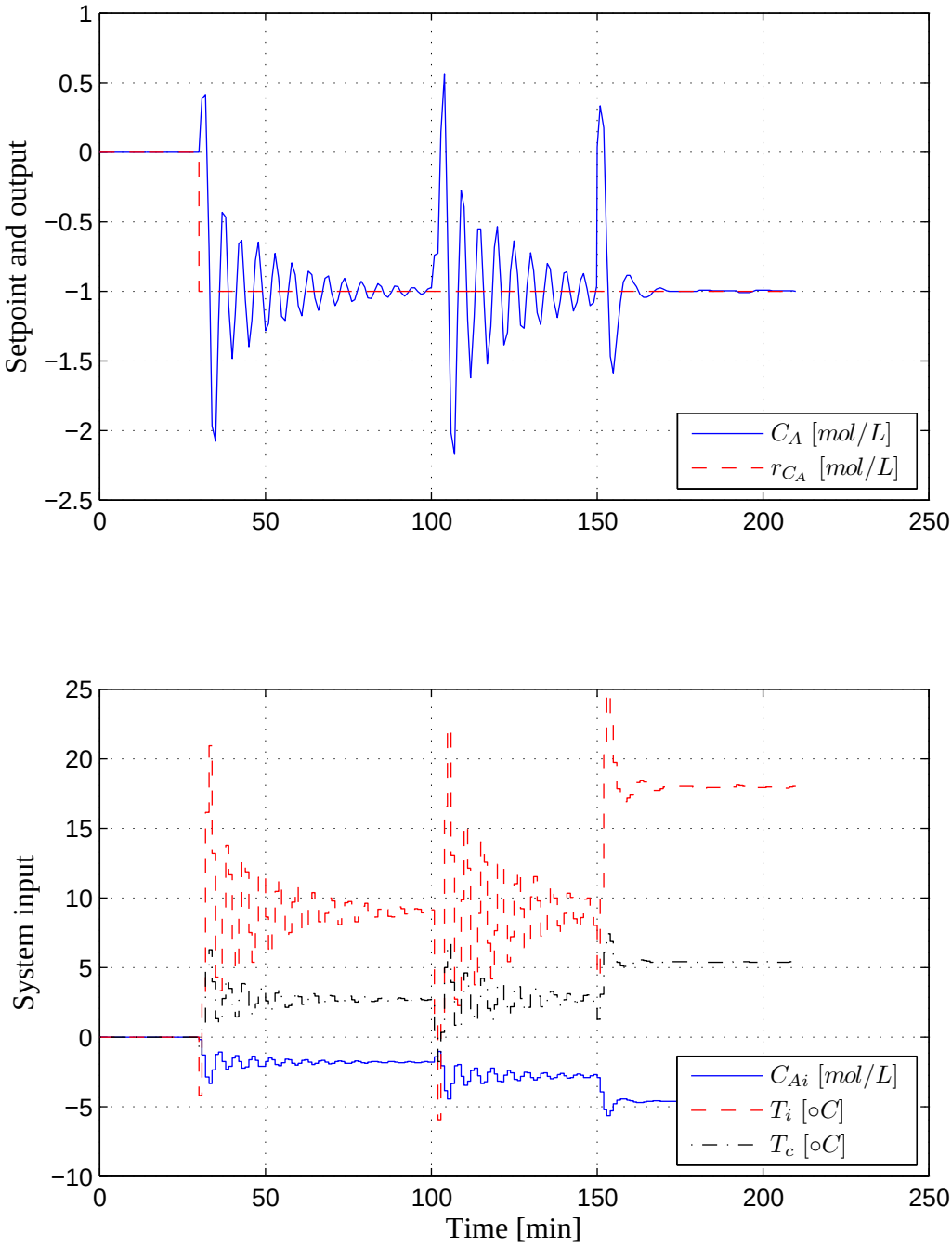

Figure 3-6. Control performance on the nonlinear CSTR system using the identical step changes in setpoint, state disturbance and output disturbance as subsection 3.3.2.3.

CHAPTER 4
CONTROL SYSTEM DESIGN AND IMPLEMENTATION USING INCREMENTAL MPC

4.1 Basic Equations for Incremental MPC

Crisalle [12] describes the design procedure of the incremental MPC. The incremental MPC is an offset-free MPC controller based on the incremental state-space prediction model

$$\begin{aligned}
\hat{x}(k+j+1) &= (I+A)\hat{x}(k+j) - A\hat{x}(k+j-1) + B\Delta u(k+j) + \Delta\hat{d}(k+j) \\
\hat{y}(k+j+1) &= C\hat{x}(k+j+1) + \hat{p}(k+j+1) \\
\hat{x}(0) &= x_0
\end{aligned}$$

where

$j = 0,\ 1, \ldots,\ N_p,\ \Delta u\,(k) \in \mathbb{R}^p,\ \hat{x}(k) \in \mathbb{R}^n,\ \hat{y}\,(k) \in \mathbb{R}^m,\ A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{m\times n}$

Assume that the current state and output vector can be directly measured. *i.e.*,

$$\begin{aligned}
\hat{x}(k) &:= x\,(k) \\
\hat{y}(k) &:= y\,(k)
\end{aligned}$$

For constant state and output disturbance, the disturbance model can be written as

$$\begin{aligned}
d(k+1) &= d(k) \\
p(k+1) &= p(k) \\
\Delta d(k) &= 0 \\
\Delta p(k) &= 0
\end{aligned}$$

The incremental MPC control objective is given by

$$J(k) = (r-\hat{y})^T Q_a (r-\hat{y}) + \Delta u^T T_a \Delta u \tag{4–1}$$

where

$$r = \begin{bmatrix} r(k+1) \\ r(k+2) \\ \vdots \\ r(k+N_c) \\ \vdots \\ r(k+N_p) \end{bmatrix}$$

and

$$\Delta u = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-2) \\ \Delta u(k+N_c-1) \end{bmatrix}$$

By minimizing (4–1), the following augmented-incremental control law is obtained.

$$\Delta u = K(r - \hat{y}^{cf}) \tag{4–2}$$

where

$$K = \left[D_l^T Q_a D_l + T_a\right]^{-1} D_l^T Q_a \in \mathbb{R}^{pN_c \times mN_p}$$

is the controller gain and

$$\hat{y}^{cf} = U_{\hat{y}} y(k) + C_l[(\hat{x}(k) - \hat{x}(k-1)] + G_l \Delta \hat{d}(k) + I_l \Delta \hat{p}(k) \tag{4–3}$$

is the constant-forcing response. The notation in (4–3) is as follows

$$U_{\hat{y}} = \begin{bmatrix} I \\ I \\ \vdots \\ \vdots \\ I \end{bmatrix}, \; I \in \mathbb{R}^{m \times m} \tag{4–4}$$

$$C_l = \begin{bmatrix} C(\sum_{i=0}^{1} A^i - I) \\ C(\sum_{i=0}^{2} A^i - I) \\ \vdots \\ C(\sum_{i=0}^{N_p} A^i - I) \end{bmatrix}$$

$$D_l = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ C\sum_{i=0}^{1} A^i B & CB & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \dots & 0 \\ C\sum_{i=0}^{N_p-2} A^i B & C\sum_{i=0}^{N_p-3} A^i B & C\sum_{i=0}^{N_p-4} A^i B & \dots & C\sum_{i=0}^{N_p-N_c-1} A^i B \\ C\sum_{i=0}^{N_p-1} A^i B & C\sum_{i=0}^{N_p-2} A^i B & C\sum_{i=0}^{N_p-3} A^i B & \dots & C\sum_{i=0}^{N_p-N_c} A^i B \end{bmatrix}$$

$$G_l = \begin{bmatrix} C \\ C(A+I) \\ C\sum_{i=0}^{2} A^i \\ \vdots \\ C\sum_{i=0}^{N_p-1} A^i \end{bmatrix}$$

and

$$I_l = \begin{bmatrix} I \\ 2I \\ \vdots \\ N_p I \end{bmatrix}$$

Defining

$$\begin{aligned} k_l^T &= U_{u|u(k-1)}^T K \\ &= \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix} K \end{aligned} \tag{4–5}$$

where $U_{u|u(k-1)}^T \in \mathbb{R}^{pN_c \times p}$ and the augmented-incremental MPC control law for the current incremental vector $\Delta u(k)$ is

$$\Delta u(k) = k_l^T \left[r - U_{\hat{y}} y(k) - C_l[(\hat{x}(k) - \hat{x}(k-1)] - G_l \Delta \hat{d}(k) - I_l \Delta \hat{p}(k) \right]$$

Noting $u(k) = u(k-1) + \Delta u(k)$, the incremental MPC control law is

$$\Delta u(k) = k_l^T [r - U_{\hat{y}} y(k)] - k_l^T C_l[(\hat{x}(k) - \hat{x}(k-1)] - k_l^T G_l \Delta \hat{d}(k) - k_l^T I_l \Delta \hat{p}(k)$$

or

$$u(k) = u(k-1) + k_l^T [r - U_{\hat{y}} y(k)] - k_l^T C_l[(\hat{x}(k) - \hat{x}(k-1)] - k_l^T G_l \Delta \hat{d}(k) - k_l^T I_l \Delta \hat{p}(k)$$

4.2 Control Validation

4.2.1 Example 1: CSTR Plant from Seborg [2]

Use the linear discrete system discussed in subsection 3.3.2 to design the incremental MPC. The incremental MPC control diagram is shown as 4-1. The inputs to the MPC controllers are the outputs and state of the plant. Such

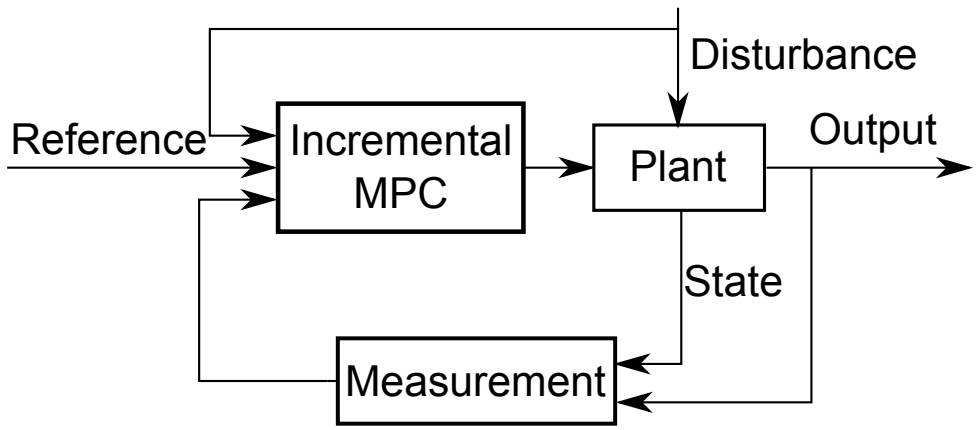

Figure 4-1. Control diagram of a closed-loop system with the incremental MPC controller

information is used to minimize the cost function and finally to find the optimal controller output which is also the input to the plant system.

Design the controller in the MATLAB/Simulink environment, the closed-loop control diagram is given in figure 4-2. The plant model is the linear discrete CSTR described in subsection 3.3.2 which contains unmeasured state disturbance d and output disturbance p. The incremental MPC controller utilizes the current state and output to calculate the controller output. The loop switch in the control diagram is used to set the plant to work under either the open-loop and the closed-loop mode.

Run the closed-loop model, using the the idential step changes as 3.3.2.3 in setpoint and disturbance. The simulation results are shown in figure 4-3. Since all the step changes regarding unmeasured state and output disturbance are the same as those in subsection 3.3.2.3, only the graphs related to the system input, output and setpoint are given in the figure. The top graph shows the input to the CSTR system as a function of time and the second graph presents the system output and setpoint as a function of time. It can be concluded that the incremental MPC controller can achieve offset-free performance on the discrete

CSTR system. The controller successfully tracks the setpoint as well as rejects disturbances without offset.

Use the identical controller on the linear contiunuous system to test its control performance. The simulation results are given in figure 4-4. As in the case of the simulation of the linear discrete system, only the graphs on the CSTR system input, output and setpoint are shown in the figure. The top graph depicts system input, where the change on the inlet concentration is larger than the input for the discrete system of figure 4-3. The bottom graph shows the outlet concentration with its setpoint as a function of time. The output control performance is better than figure 4-3 since there is no oscillation on the outlet concentration in the second graph. It also can be seen from figure 4-4 that offset-free performance is achieved by using the same incremental MPC controller, and the performance for setpoint tracking and disturbance rejection is satisfactory.

Further test of the incremental MPC controller on the non-linear CSTR system is carried out. The control performance is shown in figure 4-5. The designed controller is based on the linear discrete CSTR model and now it is applied to the nonlinear model. However, the control performance is still satisfactory. Compared to figure 3-6, the incremental MPC tracks the setpoint and rejects disturbance with much better performance than the discrete integral controller.

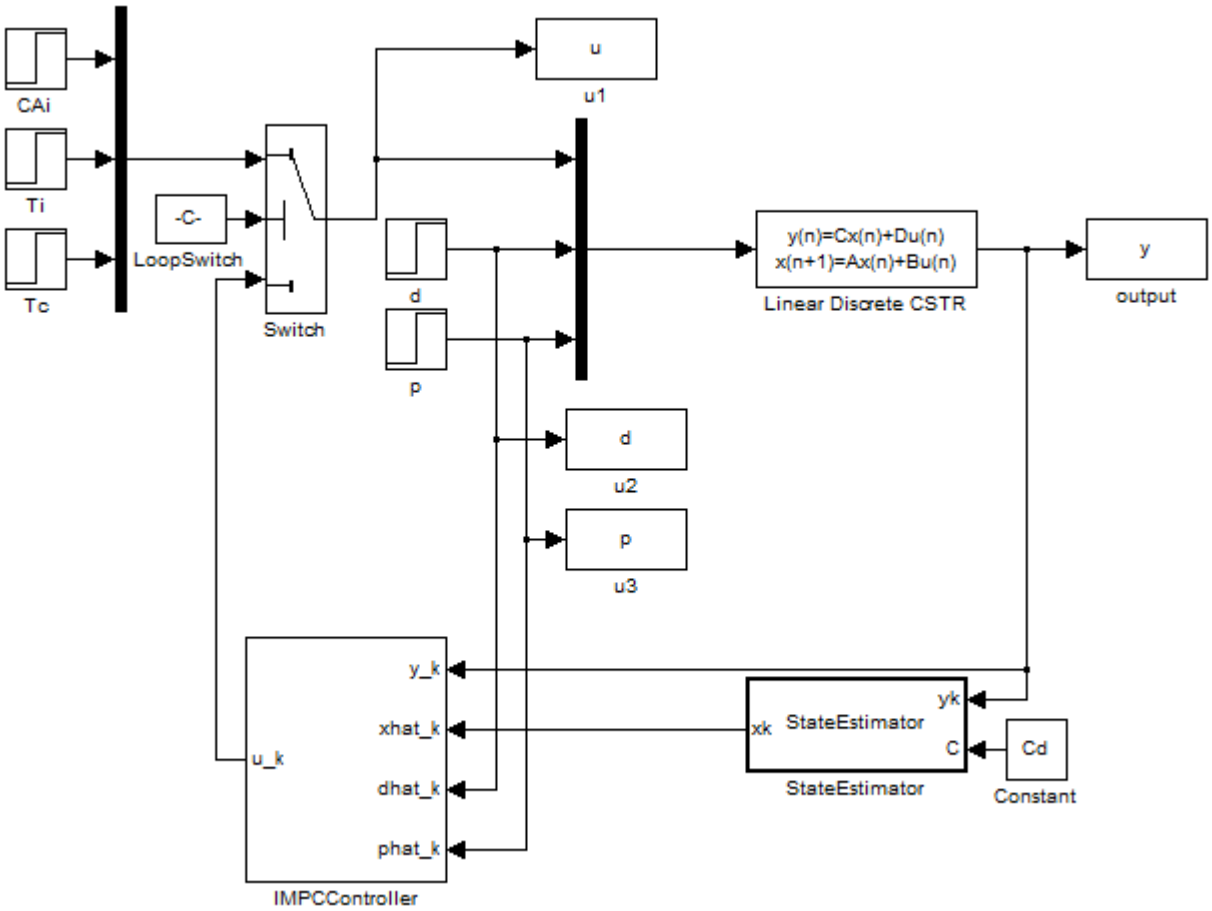

Figure 4-2. MATLAB/Simulink diagram of the closed-loop system with the incremental MPC controller

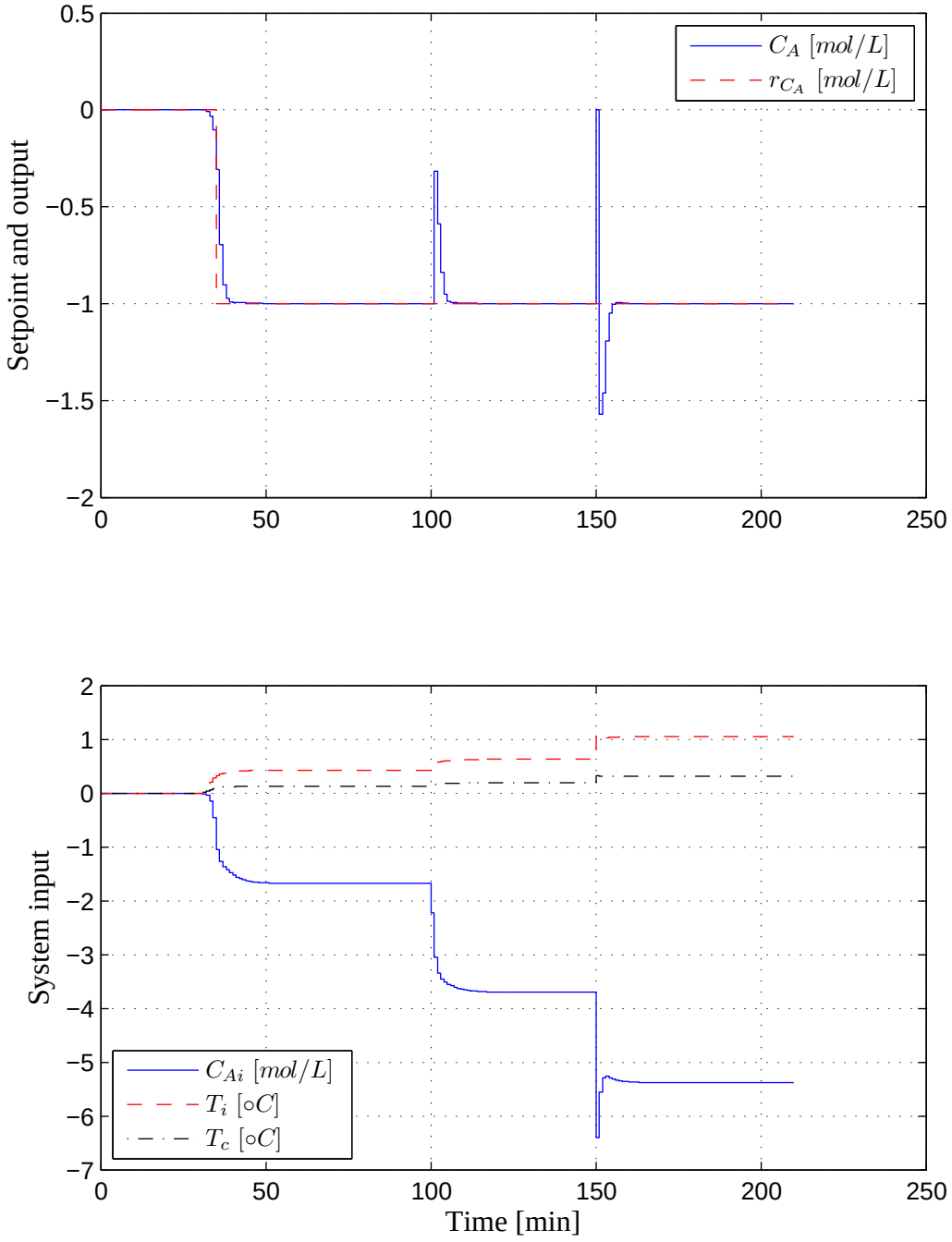

Figure 4-3. Simulation results for the linear discrete CSTR system with the incremental MPC controller

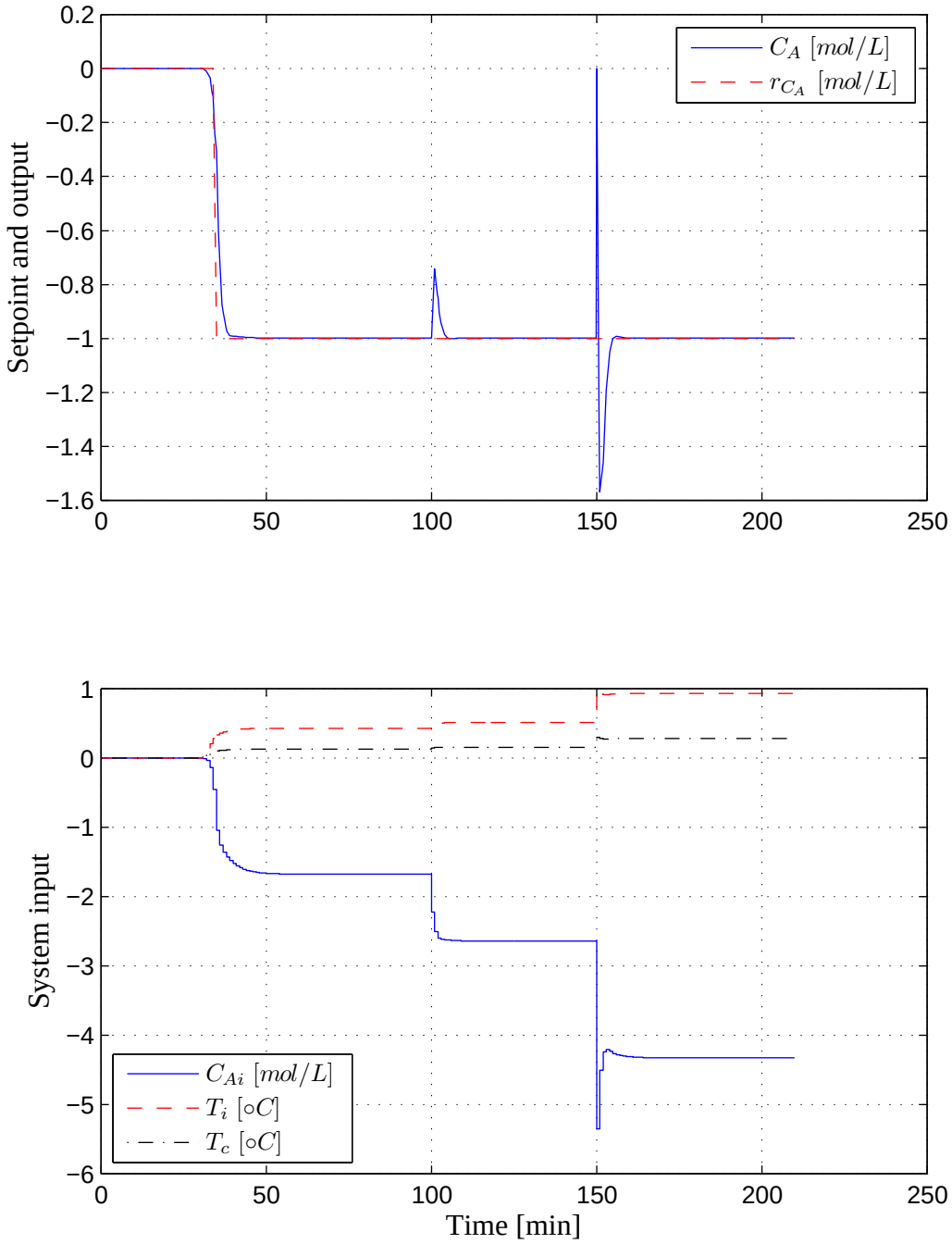

Figure 4-4. Simulation results for the linear continuous CSTR system with incremental MPC controller

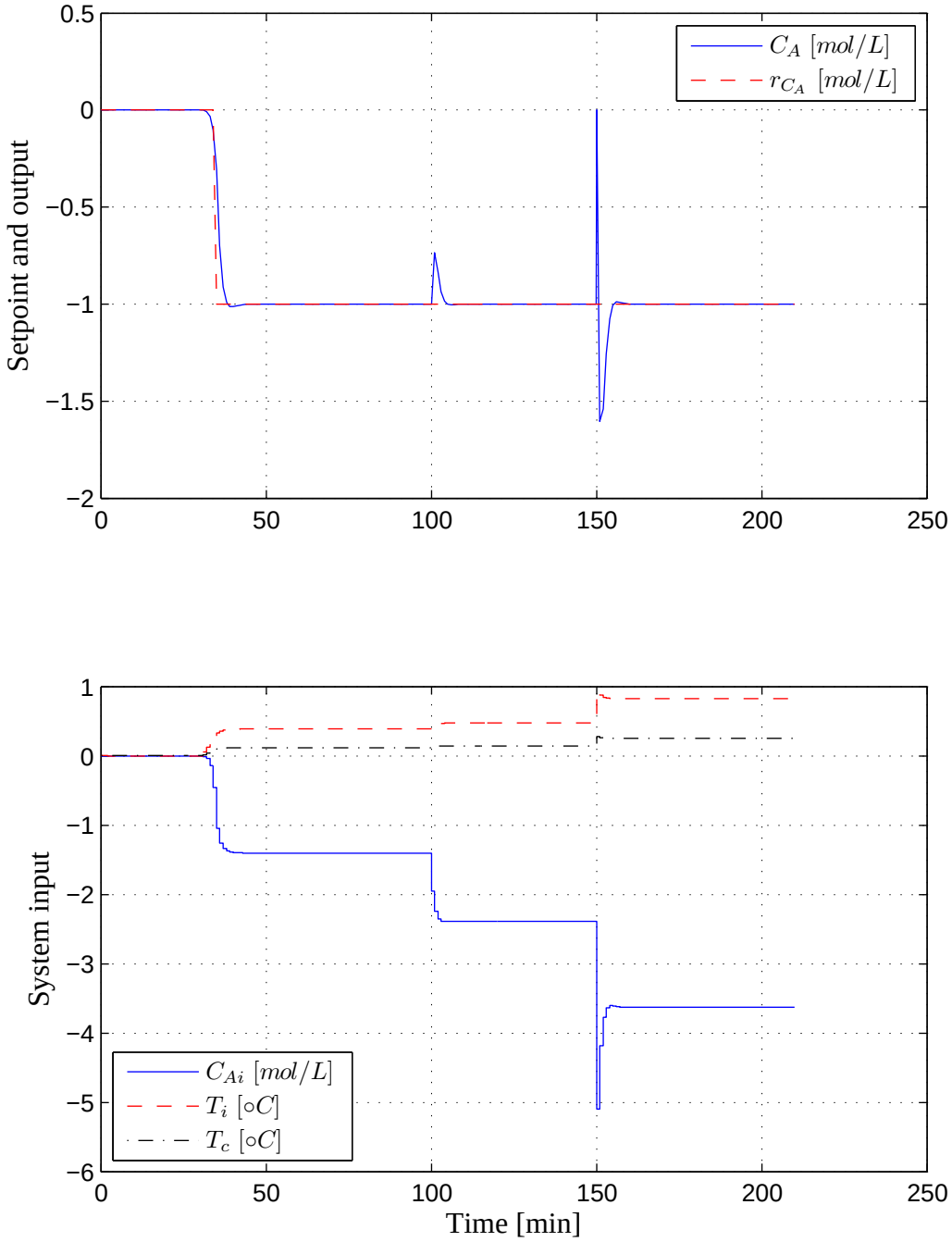

Figure 4-5. Simulation results for the nonlinear CSTR system with incremental MPC controller

4.2.2 Example 2: Quadruple Tank System

The quadruple-tank laboratory process is used to test the control performance of MPC designed by Åkesson [3]. The system consists of four tanks and their positions are arranged as a two-by-two matrix, where water from the two top tanks flows into the two bottom tanks. Two pumps are used. One is adopted to transport water into the top left tank and the bottom right tank. The other one is used to transport water into the top right tank and bottom left tank. A valve is used to adjust pump capacity to the top and bottom tank respectively. The control variables are the pump voltages. Let the states of the system be defined by the water levels of the tanks (expressed in cm) x_1, x_2, x_3 and x_4 respectively. The maximum level of each tank is $20\,\mathrm{cm}$. System dynamics are given as

$$\dot{x}_1 = -\frac{a_1}{A_2}\sqrt{2gx_1} + \frac{a_3}{A_1}\sqrt{2gx_3} + \frac{\gamma_1 k_1}{A_1}u_1 \tag{4–6}$$

$$\dot{x}_2 = -\frac{a_2}{A_2}\sqrt{2gx_2} + \frac{a_4}{A_2}\sqrt{2gx_4} + \frac{\gamma_2 k_2}{A_2}u_2 \tag{4–7}$$

$$\dot{x}_3 = -\frac{a_3}{A_3}\sqrt{2gx_3} + \frac{(1-\gamma_2)k_2}{A_3}u_2 \tag{4–8}$$

$$\dot{x}_4 = -\frac{a_4}{A_4}\sqrt{2gx_4} + \frac{(1-\gamma_1)k_1}{A_4}u_1 \tag{4–9}$$

where the A_is and the a_is represent the cross section area of the tanks and the tubes, respectively. γ_is are defined as the position of the valves which control the flow rate to the top and bottom tanks respectively. The control variables are the u_is. The objective is to control the level of the two bottom tanks denoted as x_1 and x_2. The process diagram is presented in figure 4-6. The parameter values are shown in table 4-1 and the initial values of the system are given in table 4-2.

Incremental MPC is used here to test its control performance on the linearized quadruple tank system. A step change of the setpoint of level for tank 1 of $6\,\mathrm{cm}$ is applied at $t = 60\,\mathrm{s}$, while the setpoint of tank level 2 is held constant. At $t = 600\,\mathrm{s}$, a unit step disturbance is applied to the second input

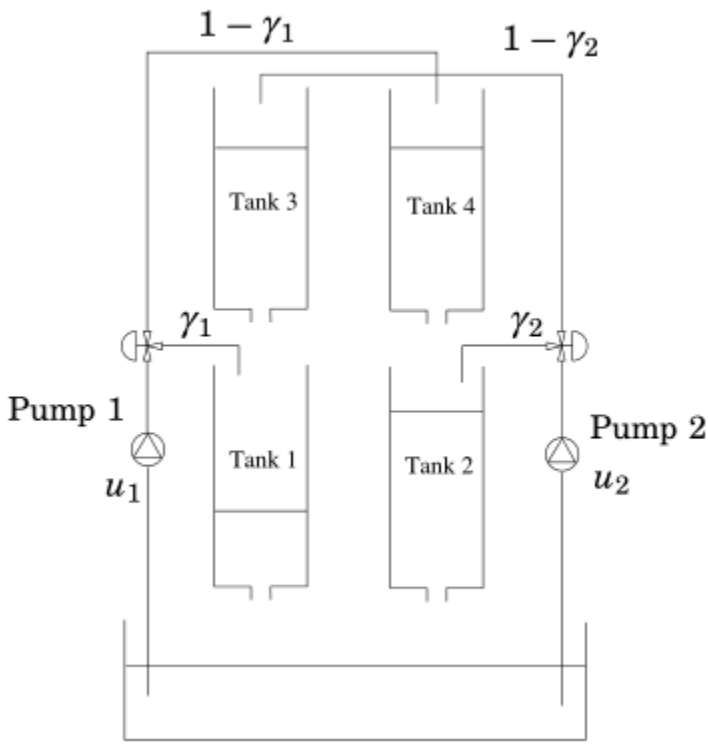

Figure 4-6. Process diagram for the quadruple tank system from Åkesson [3].

Table 4-1. Parameter values for the quadruple tank system

Parameter	Nominal Value
A_1, A_2	28 cm^2
A_3, A_4	32 cm^2
a_1, a_2	0.071 cm^2
a_3, a_4	0.057 cm^2
k_1	3.33 cm^3/Vs
k_2	3.35 cm^3/Vs
g	981cm/s^2

Table 4-2. Initial values of the quadruple tank system

Parameter	Initial Value
x_1^0	8.2444 cm
x_2^0	19.0163 cm
x_3^0	4.3146 cm
x_4^0	8.8065 cm
u_1^0, u_2^0	3 V

channel. The simulation results are shown in figure 4-7. The top graph presents the level for tank 1 with its setpoint and the bottom graph shows the system input. The controller achieves offset-free performance on both setpoint tracking and disturbance rejection with satisfactory transient process. The level for tank 1 dose not exceed $20\,\mathrm{cm}$, which ensures that system is stable.

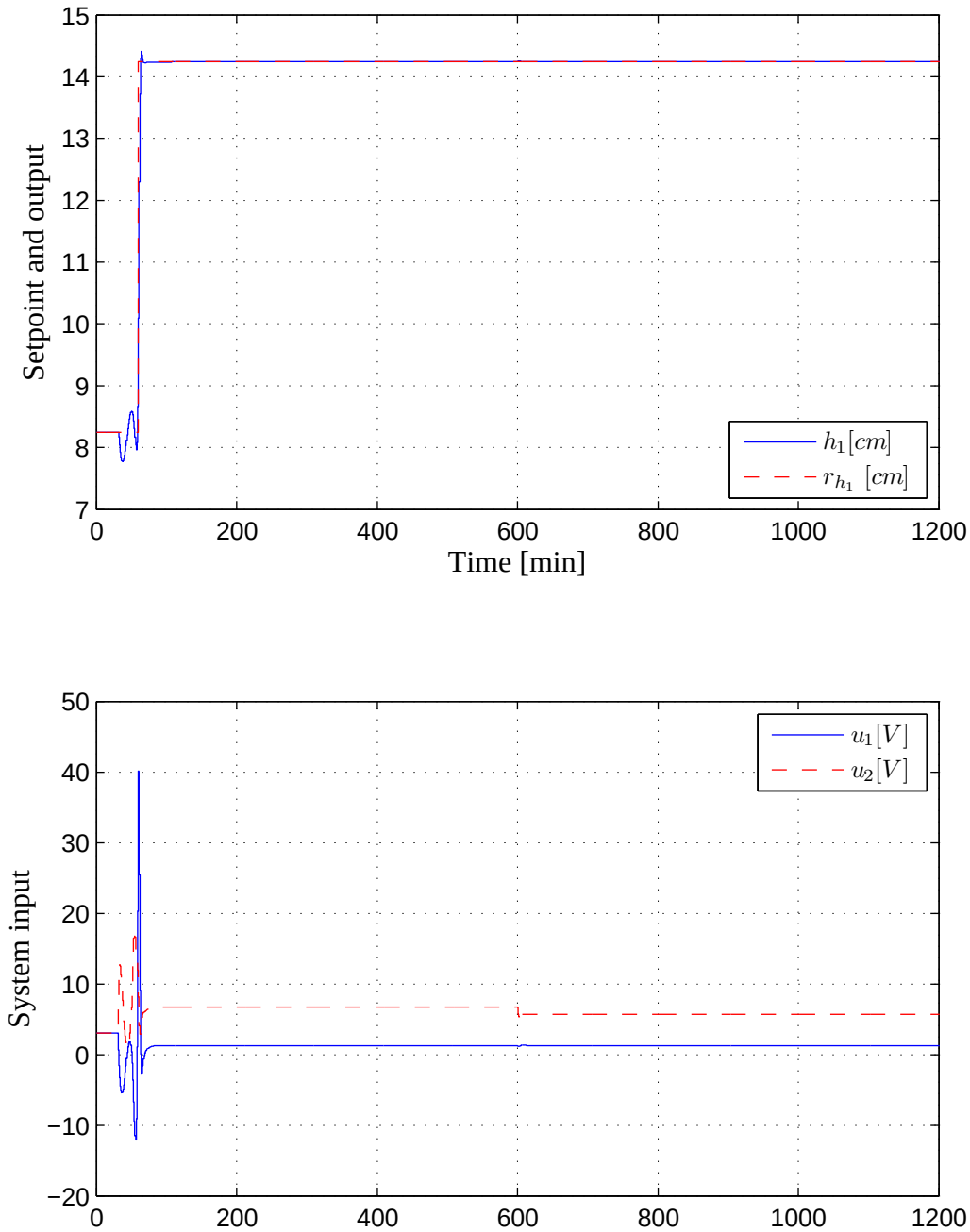

Figure 4-7. Simulation results of incremental MPC on the linearized quadruple tank system. The control variable with its setpoint which is presented in the top graph is the level for tank 1. Two input variables denoted as u_1 and u_2 are shown in the bottom graph.

CHAPTER 5
CONTROL SYSTEM DESIGN AND IMPLEMENTATION USING INTEGRAL MPC

5.1 Basic Equations for Integral MPC

Crisalle [12] indicates that the standard state-space equation and the incremental state-space state equation can be augmented with one or more integral-state equations. In this thesis, the first two integral-states $z(k) \in \mathbb{R}^m$ and $w(k) \in \mathbb{R}^m$ are introduced as

$$z(k+1) = z(k) + r(k) - y(k) \tag{5–1}$$

$$w(k+1) = w(k) + z(k) \tag{5–2}$$

where $y(k) \in \mathbb{R}^m$ is the current system output vector and $r(k) \in \mathbb{R}^m$ is the current setpoint vector.

5.1.1 First Integral State $z(k)$

The first integral state equation (5–1) can be used to generate prediction equations

$$\begin{aligned}
z(k+1) &= z(k) + r(k) - y(k) \\
z(k+2) &= z(k+1) + r(k+1) - \hat{y}(k+1) \\
&\vdots \\
z(k+N_p-1) &= z(k+N_p) + r(k+N_p-1) - \hat{y}(k+N_p-1) \\
z(k+N_p) &= z(k+N_p) + r(k+N_p) - \hat{y}(k+N_p)
\end{aligned}$$

Apply a recursive substitution routine to the prediction equations to obtain *First Integral State Predictor* as

$$\begin{bmatrix} z(k+1) \\ z(k+2) \\ z(k+3) \\ \vdots \\ z(k+N_p-1) \\ z(k+N_p) \end{bmatrix} = \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \\ I \end{bmatrix} z(k) + \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \\ I \end{bmatrix} [r(k) - y(k)] + \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ I & 0 & 0 & \cdots & 0 & 0 \\ I & I & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ I & I & I & \cdots & 0 & 0 \\ I & I & I & \cdots & I & 0 \end{bmatrix}$$
$$\times \left\{ \begin{bmatrix} r(k+1) \\ r(k+2) \\ r(k+3) \\ \vdots \\ r(k+N_p-1) \\ r(k+N_p) \end{bmatrix} - \begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \\ \vdots \\ \hat{y}(k+N_p-1) \\ \hat{y}(k+N_p) \end{bmatrix} \right\} \tag{5–3}$$

which can be rewritten as the following compact form that is referred as the *First Integral State Prediction Equation*

$$z = U_{\hat{y}} [r(k) - y(k)] + U_{\hat{y}} z(k) + V_{\hat{y}} r - V_{\hat{y}} \hat{y} \tag{5–4}$$

where $U_{\hat{y}}$ is defined as (4–4) and $V_{\hat{y}}$ is the augmented matrix as

$$V_{\hat{y}} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ I & 0 & 0 & \cdots & 0 & 0 \\ I & I & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ I & I & I & \cdots & 0 & 0 \\ I & I & I & \cdots & I & 0 \end{bmatrix} \tag{5–5}$$

which consists of the matrices $I \in \mathbb{R}^{m \times m}$ and $0 \in \mathbb{R}^{m \times m}$.

5.1.2 Second Integral State $w(k)$

The second integral state equation (5–2) can be shifted forward to generate the prediction equations

$$\begin{aligned} w(k+1) &= w(k) + z(k) \\ w(k+2) &= w(k+1) + z(k+1) \\ &\vdots \\ w(k+N_p-1) &= w(k+N_P) + z(k+N_p-1) \\ w(k+N_p) &= w(k+N_p) + z(k+N_p) \end{aligned}$$

Apply a recursive substitution routine to the prediction equations to obtain *Second Integral State Predictor*

$$\begin{bmatrix} w(k+1) \\ w(k+2) \\ w(k+3) \\ \vdots \\ w(k+N_p-1) \\ w(k+N_p) \end{bmatrix} = \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \\ I \end{bmatrix} z(k) + \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \\ I \end{bmatrix} w(k) + \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ I & 0 & 0 & \cdots & 0 & 0 \\ I & I & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ I & I & I & \cdots & 0 & 0 \\ I & I & I & \cdots & I & 0 \end{bmatrix} \times \begin{bmatrix} z(k+1) \\ z(k+2) \\ z(k+3) \\ \vdots \\ z(k+N_p-1) \\ z(k+N_p) \end{bmatrix}$$

which can be rewritten as the following compact form

$$w = U_{\hat{y}} z(k) + U_{\hat{y}} w(k) + V_{\hat{y}} z \tag{5–6}$$

where $U_{\hat{y}}$ and $V_{\hat{y}}$ are the augmented matrices defined as (4–4) and (5–5). Substitute the first integral state vector z in the right hand side of (5–7) by (5–4) to obtain the following *Second Integral State Prediction Equation*

$$w = V_{\hat{y}} U_{\hat{y}} \left[r(k) - y(k) \right] + (V_{\hat{y}} U_{\hat{y}} + U_{\hat{y}}) z(k) + U_{\hat{y}} w(k) + V_{\hat{y}}^2 r - V_{\hat{y}}^2 \hat{y} \tag{5–7}$$

5.1.3 Control Law for the Integral MPC Controller

Considering the integral MPC performance objective

$$J(k) = (r - \hat{y})^T Q_a (r - \hat{y}) + \Delta u^T T_a \Delta u + u^T R_a u + (r - \hat{y})^T S_a \Delta u$$

the control law for the integral MPC is given by

$$\Delta u(k) \quad = \quad k_l^T \left[r - U_{\hat{y}} y(k) \right] + \; k_z z + k_w w$$

or

$$u(k) = u(k-1) + k_l^T \left[r - U_{\hat{y}} y(k) \right] + k_z z + k_w w \tag{5–8}$$

as is described in Peek [13], where z and w are given by equation (5–4) and (5–7) respectively. The gain k_l^T is defined as (4–5). The gains k_z and k_w for z and w are denoted as

$$\begin{aligned} k_z &= k_l^T Q_a^{-1} V_{\hat{y}}^T S_a \\ k_w &= k_z S_a^{-1} V_{\hat{y}}^T T_a \end{aligned}$$

5.2 Control Validation

Design the integral MPC controller based on the linear discrete CSTR system 3.3.2 then test its closed-loop control performance using the idential step changes on the setpoint and disturbance as subsection 3.3.2.3. The simulation result is shown in figure 3–3. The integral MPC control diagram and Simulink diagram are similar to the incremental MPC and hence they are not given here. It can be concluded from the top graph in the figure that the integral MPC controller successfully tracks the setpoint as well as reject the state and output disturbance without offset.

Continue to test the integral MPC controller on the linear continuous CSTR system using the same step changes. The simulation result is given in figure 5-2. It can be seen from the figure that the integral MPC controller successfully

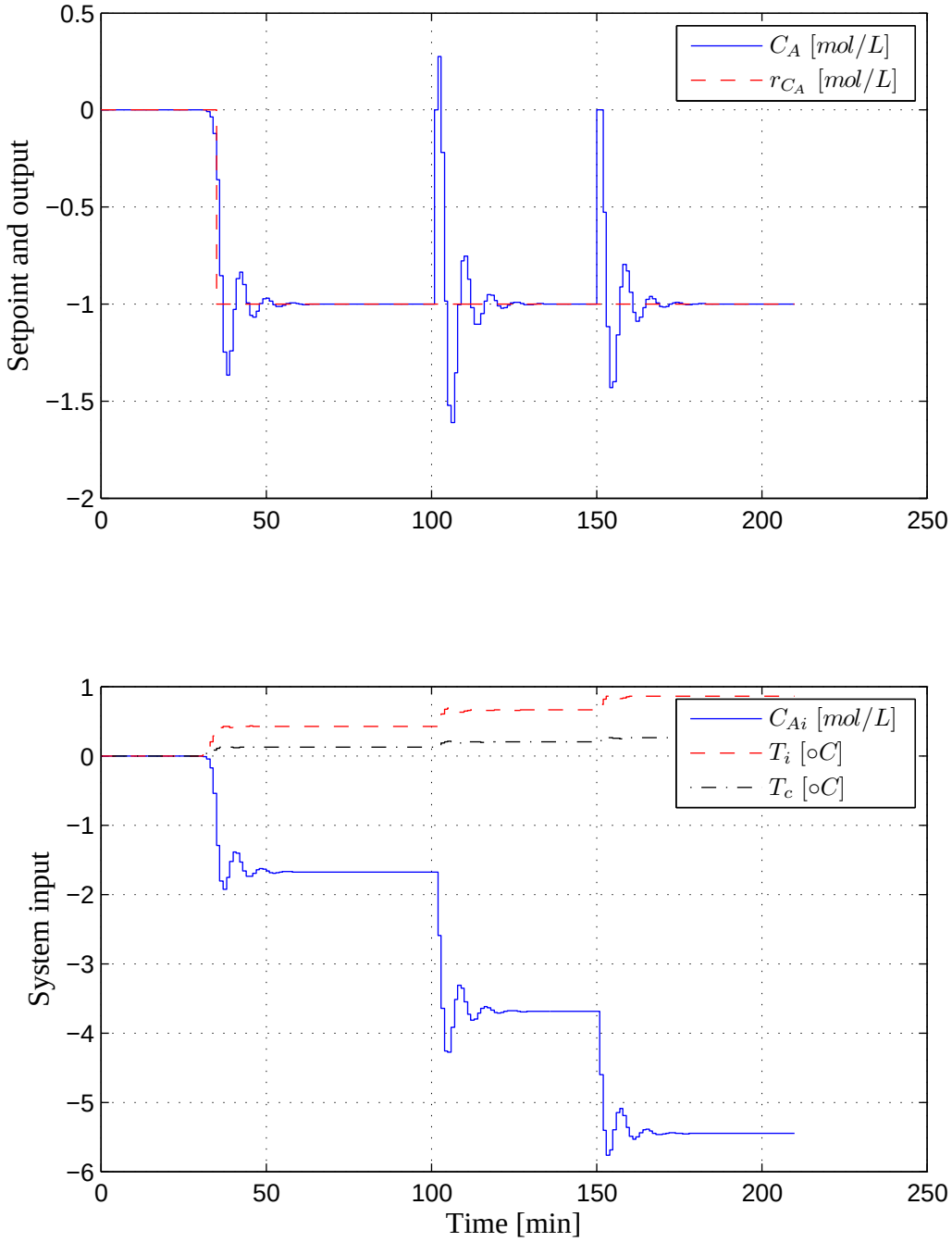

Figure 5-1. Closed-loop simulation result for the linear discrete CSTR system with an integral MPC controller. The CSTR is given by subsection 3.3.2 and the step changes are presented in subsection 3.3.2.3

achieves the satisfactory offset-free control performance on both setpoint tracking and disturbance rejection.

Finally run the simulation for the nonlinear CSTR system with the integral MPC controller using the same step changes on the setpoint and disturbance. The simulation result is presented by figure 5-3. It can be concluded that the integral MPC controller successfully tracks the setpoint and rejects disturbance. Also its offset elimination performance is satisfactory. Further validation process can be adopted to test the integral MPC controller on the real CSTR system.

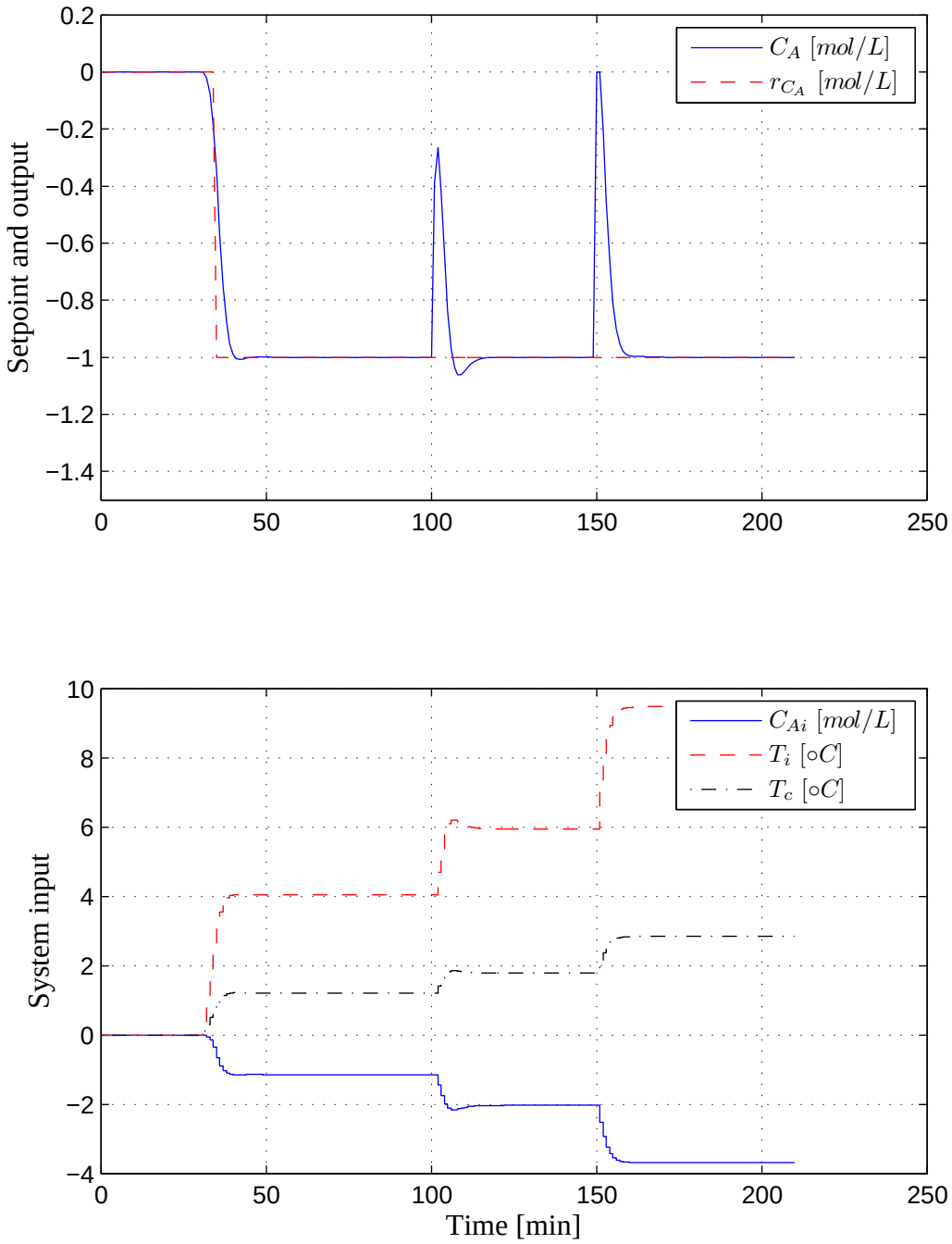

Figure 5-2. Closed-loop simulation result for the linear continuous CSTR system with an integral MPC controller. The CSTR is given by subsection 3.3.2 and the step changes are presented in subsection 3.3.2.3

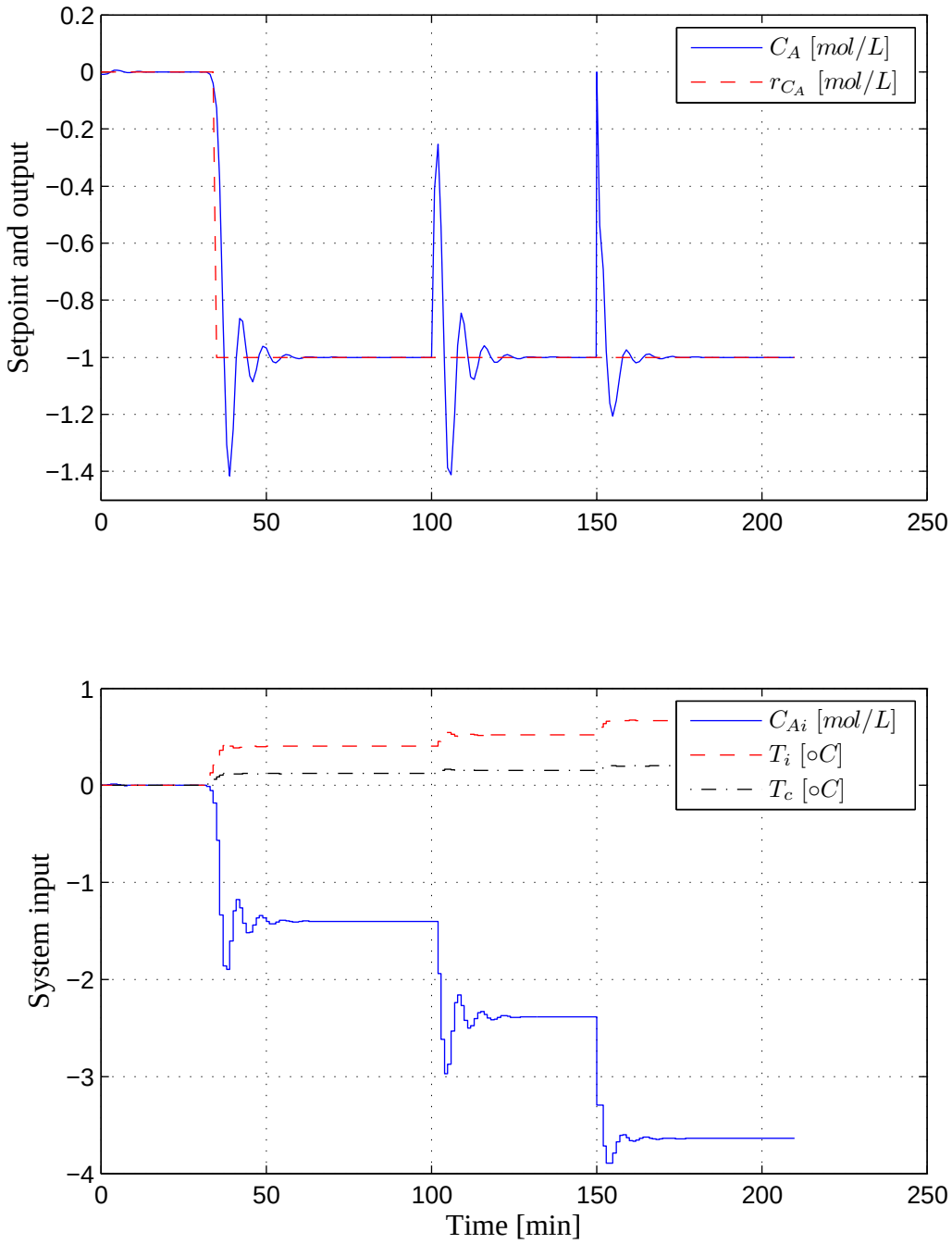

Figure 5-3. Closed-loop simulation result for the nonlinear CSTR system with an integral MPC controller. The CSTR is given by subsection 3.3.2 and the step changes are presented in subsection 3.3.2.3

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this thesis the control performance of three different controllers is examined using the same plant model and step changes on both the setpoint and disturbance. They are discrete integral controller, incremental MPC controller and integral MPC controller. All of them can successfully track the setpoint and eliminate the setpoint without offset with satisfactory closed-loop control performance on the plant system of linear discrete CSTR and linear continuous CSTR. However, due to the increased model mismatch, the control performance of the discrete integral controller deteriorates significantly when it is applied to the nonlinear CSTR system. Its simulation result on the nonlinear CSTR shows strong oscillation on the controlled variable. For the incremental and integral MPC controller, satisfactory offset-free control performance is achieved on the nonlinear CSTR system, which indicates they are more robust to model mismatch than the discrete integral controller. In the application, when the model uncertainty is significant, it is preferred to use the incremental or the integral MPC control strategy instead of the discrete integral control.

For the incremental and integral MPC controllers implemented using S-functions, there is a problem on the sigularity of the algebraic loop. This is due to when flag = 2 call, the S-function will automatically update the state vector, which will need the current plant input $u(k)$. To eliminate the algebraic loop singularity, a memory block is needed at the entry of the controller.

6.2 Future Work: Constrained Incremental MPC

The future objective is the design and implementation of the constrained incremental MPC controller under the MATLAB/Simulink environment. In this thesis, only the unconstrained incremental MPC is discussed.

Wang [14] introduces the approach to design the constrained MPC, which indicates that the constrained controll problem in the context of predictive control is actually a quadratic programming problem. The frequently used constraints in applications are the control variable $u(k)$, the output $y(k)$, the state variable $x(k)$ and the rate of change of the control variable $\Delta u(k)$. The upper and lower limit may be imposed into these constraints represented as

$$\begin{aligned} u_{min} \leq \quad & u(k) \quad \leq u_{max} \\ y_{min} \leq \quad & y(k) \quad \leq y_{max} \\ x_{min} \leq \quad & x(k) \quad \leq x_{max} \\ \Delta u_{min} \leq \quad & \Delta u(k) \quad \leq \Delta u_{max} \end{aligned}$$

The steps for the constraint MPC design are:

1. Specifying system operational limits, including control variable $u(k)$, output $y(k)$, state variable $x(k)$ and rate of change of the control variable $\Delta u(k)$
2. Expressing the limits by using the notations of minimum and maximum of $u(k)$, $y(k)$, $x(k)$ and $\Delta u(k)$
3. Expressing these minimum and maximum values in the form of inequalities that consist of the parameterized future control trajectory $\Delta u(k)$, $\Delta u(k+1)$, $...\Delta u(k+N_c-1)$
4. The design of the constrained MPC now is converted to the minimization of the original penalty function subject to the inequalities, where the parameters $\Delta u(k)$, $\Delta u(k+1)$, $...\Delta u(k+N_c-1)$ become the decision variables.
5. Using quadratic programming methods to solve the constrained optimization problem at each sampling instant to obtain the optimal decision variables.

Since the constraints are expressed as inequalities, generally speaking there is no analytical solution to the constrained control problem, unless the active constraints for each sampling instant are known. Under the condition that the active constraints are known, optimal solution of decision variables can be found

in analytical form. For the most of practical applications, instead of iteratively solving for the decision variables, Hildreth's programming routine can be used to identify the active set of constraints via Lagrange multipliers in Crisalle [15], which would make the constraints much simpler and thus only an easy iterative routine is needed in finding the optimal solution of the multipliers. This is crucial for the online implementation of the MPC controller.

Appendix A
DERIVATION OF K_1 AND K_2 FOR THE DISCRETE INTEGRAL CONTROLLER

The state and the integral state at the time $k+1$ are

$$\begin{aligned} z(k+1) &= z(k) + r(k+1) - y(k+1) \\ &= z(k) + r(k+1) - C_d\left[A_d x(k) + B_d u(k)\right] \\ &= -C_d A_d x(k) + z(k) - C_d B_d u(k) + r(k+1) \end{aligned}$$

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ &= A_d x(k) + B_d\left[-K_2 x(k) + K_1 z(k)\right] \end{aligned}$$

In this servo system, the design parameters are the gain matrices K_1 and K_2 so that the system has the desired closed-loop poles. From the state-space equation, calculate $u(k+1)$ as

$$\begin{aligned} u(k+1) &= -K_2 x(k+1) + K_1 z(k+1) \\ &= (K_2 - K_2 A_d - K_1 C_d A_d)x(k) + (I_m - K_2 B_d - K_1 C_d B_d)u(k) \\ &\quad + K_1 r(k+1) \end{aligned} \tag{A–1}$$

Considering the step input, $x(k)$ and $u(k)$ converge to the constant vector, named $x(\infty)$ and $u(\infty)$. At steady state, equation (3–4) becomes

$$\begin{bmatrix} x(\infty) \\ u(\infty) \end{bmatrix} = \begin{bmatrix} A_d & B_d \\ K_2 - K_2 A_d - K_1 C_d A_d & I_m - K_2 B_d - K_1 C_d B_d \end{bmatrix} \begin{bmatrix} x(\infty) \\ u(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 r \end{bmatrix}$$

Using the deviation vectors

$$\begin{aligned} \tilde{x}(k) &= x(k) - x(\infty) \\ \tilde{u}(k) &= u(k) - u(\infty) \end{aligned}$$

(3–4) can be rewritten as

$$\begin{bmatrix} \tilde{x}(k+1) \\ \tilde{u}(k+1) \end{bmatrix} = \begin{bmatrix} A_d & B_d \\ K_2 - K_2 A_d - K_1 C_d A_d & I_m - K_2 B_d - K_1 C_d B_d \end{bmatrix} \times \begin{bmatrix} \tilde{x}(k) \\ \tilde{u}(k) \end{bmatrix} \tag{A–2}$$

For convenience, we use the equivalent form

$$\begin{bmatrix} \tilde{x}(k+1) \\ \tilde{u}(k+1) \end{bmatrix} = \begin{bmatrix} A_d & B_d \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ \tilde{u}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ I_m \end{bmatrix} w(k) \tag{A–3}$$

where

$$w(k) = \begin{bmatrix} K_2 - K_2 A_d - K_1 C_d A_d & I_m - K_2 B_d - K_1 C_d B_d \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ \tilde{u}(k) \end{bmatrix}$$

Given

$$\begin{aligned} \xi(k) &= \begin{bmatrix} \tilde{x}(k) \\ \tilde{u}(k) \end{bmatrix} \in \mathbb{R}^{n+m} \\ \bar{A} &= \begin{bmatrix} A_d & B_d \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)} \\ \bar{B} &= \begin{bmatrix} 0 \\ I_m \end{bmatrix} \in \mathbb{R}^{(n+m)\times m} \\ \bar{K} &= -\begin{bmatrix} K_2 - K_2 A_d - K_1 C_d A_d & I_m - K_2 B_d - K_1 C_d B_d \end{bmatrix} \in \mathbb{R}^{m\times(n+m)} \end{aligned}$$

(A–3) becomes

$$\begin{aligned} \xi(k+1) &= \bar{A}\xi(k) + \bar{B}w(k) \\ w(k) &= -\bar{K}\xi(k) \end{aligned} \tag{A–4}$$

For the augmented system (A–4), the controllability matrix is definded as

$$\begin{bmatrix} \bar{B} & \bar{A}\bar{B} & \dots & \bar{A}^{n+m-1}\bar{B} \end{bmatrix} \in \mathbb{R}^{(n+m)\times m(n+m)} \tag{A–5}$$

In terms of A_d and B_d (A–5) can be expressed as

$$\begin{bmatrix} \bar{B} & \bar{A}\bar{B} & \dots & \bar{A}^{n+m-1}\bar{B} \end{bmatrix} = \begin{bmatrix} 0 & B_d & A_dB_d & \dots & A_d^{n+m-2}B_d \\ I_m & 0 & 0 & \dots & 0 \end{bmatrix} \tag{A–6}$$

Since it is assumed that (3–1) is completely state controllable, the rank of matrix

$$\begin{bmatrix} B_d & A_dB_d & \dots & A_d^{n-1}B_d \end{bmatrix}$$

equals n. Therefore the rank of (A–6) equals $n+m$, which makes it feasible to arbitrarily place desired closed-loop poles of (A–4) by using pole-placement method. First notice that

$$\begin{aligned} &\begin{bmatrix} K_2 & K_1 \end{bmatrix} \begin{bmatrix} A_d - I_n & B_d \\ C_dA_d & C_dB_d \end{bmatrix} \\ = &\begin{bmatrix} K_2A_d - K_2 + K_1C_dA_d & K_2B_d + K_1C_dB_d \end{bmatrix} \end{aligned} \tag{A–7}$$

and thus solve for (A–7) to obtain

$$\begin{aligned} \bar{K} &= \begin{bmatrix} K_2A_d - K_2 + K_1C_dA_d & -I_m + K_2B_d + K_1C_dB_d \end{bmatrix} \\ &= \begin{bmatrix} K_2 & K_1 \end{bmatrix} \begin{bmatrix} A_d - I_n & B_d \\ C_dA_d & C_dB_d \end{bmatrix} + \begin{bmatrix} 0 & -I_m \end{bmatrix} \end{aligned}$$

To find K_1 and K_2 , notice that

$$\begin{bmatrix} K_2 & K_1 \end{bmatrix} \begin{bmatrix} A_d - I_n & B_d \\ C_dA_d & C_dB_d \end{bmatrix} = \bar{K} - \begin{bmatrix} 0 & -I_m \end{bmatrix}$$

Then K_1 and K_2 are expressed as

$$\begin{bmatrix} K_2 & K_1 \end{bmatrix} = \left[\bar{K} + \begin{bmatrix} 0 & I_m \end{bmatrix}\right] \begin{bmatrix} A_d - I_n & B_d \\ C_d A_d & C_d B_d \end{bmatrix}^{-1}$$

REFERENCES

[1] K. Ogata, *Discrete-time control systems*. Prentice-Hall Englewood Cliffs, NJ, 1995, vol. 8.

[2] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III, *Process dynamics and control*. Wiley, 2010.

[3] J. Åkesson, *MPCtools 1.0 Reference Manual*, Department of Automatic Control Lund Institute of Technology, Box 118 SE-221 00 Lund Sweden, January 2006.

[4] J. B. Rawlings, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[5] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, 2009.

[6] Y. Qian, "Conditions and methods for offset-free performance in discrete control systems," Master's thesis, University of Florida, 2012.

[7] U. Maeder and Morari, "Offset-free reference tracking for predictive controllers," in *Decision and Control, 2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 5252–5257.

[8] U. Maeder and M. Morari, "Offset-free reference tracking with model predictive control," *Automatica*, vol. 46, no. 9, pp. 1469–1476, 2010.

[9] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE Journal*, vol. 49, no. 2, pp. 426–437, 2003.

[10] J. B. Rawlings, E. S. Meadows, and K. R. Muske, "Nonlinear model predictive control: A tutorial and survey," in *Preprints IFAC Symposium ADCHEM, Kyoto, Japan*, 1994, pp. 203–214.

[11] MathWorks, *Writing S-Functions*.

[12] O. D. Crisalle, *Model Predictive Control - A State-Space Analysis and Design Approach*, 2011.

[13] C. S. Peek, "High performance control theory, design, and applications," Ph.D. dissertation, University of Florida, 2008.

[14] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB®*. Springer, 2009.

[15] O. D. Crisalle, "Introduction to optimal control," Course notes, 2011.

BIOGRAPHICAL SKETCH

Xin Lin received his Master of Science degree in chemical engineering from the University of Florida in the spring of 2013 and bachelor's degree in automation at the Department of Control Science and Engineering from Zhejiang University (Hangzhou, China) in 2011. In January 2011 he joined the process control research group under the guidance of Dr. Oscar D. Crisalle. As a member of the group, his research concentration lay on the design and implementation of the incremental and integral controllers in the MATLAB/Simulink environment.