การควบคุมเชิงทำนายแบบจำลองในกรณีทั่วไปสำหรับกระบวนการควบคุมระดับของเหลวผ่านทางระบบควบคุมแบบกระจายตัว

นาย โซฟีก เฮล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

GENERALIZED PREDICTIVE CONTROL WITH EXTENDED PREDICTIVE CONTROL FOR TWO-TANK LEVEL CONTROL VIA DISTRIBUTED CONTROL SYSTEM

Mr. Sopheak Hel

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2008

Thesis Title	GENERALIZED PREDICTIVE CONTROL WITH EXTENDED PREDICTIVE CONTROL FOR TWO-TANK LEVEL CONTROL VIA DISTRIBUTED CONTROL SYSTEM
By	Mr. Sopheak Hel
Field of Study	Electrical Engineering
Advisor	Assistant Professor Manop Wongsaisuwan, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of Requirements for the Master's Degree

.... Dean of the Faculty of Engineering
(Associate Professor Boonsom Lerdhirunwong, Dr.Ing.)

THESIS COMMITTEE

.... Chairman
(Associate Professor David Banjerdpongchai, Ph.D.)

.... Advisor
(Assistant Professor Manop Wongsaisuwan, Ph.D.)

.... Examiner
(Associate Professor Waree Kongprawechnon, Ph.D.)

โซฟีก เฮส : การควบคุมเชิงทำนายแบบจำลองในกรณีทั่วไปสำหรับกระบวนการควบคุมระดับของเหลวผ่านทางระบบควบคุมแบบกระจายตัว (GENERALIZED PREDICTIVE CONTROL WITH EXTENDED PREDICTIVE CONTROL FOR TWO-TANK LEVEL CONTROL VIA DISTRIBUTED CONTROL SYSTEM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.มานพ วงศ์สายสุวรรณ, 65 หน้า.

ขั้นตอนวิธีของการควบคุมเชิงทำนายเกิดขึ้นในช่วงกลางศตวรรษ 1970 โดยเริ่มจากการประยุกต์ใช้งานด้านอุตสาหกรรม ในวิทยานิพนธ์นี้ การควบคุมเชิงทำนายแบบขยาย (Extended Predictive Control) ได้ถูกนำมาใช้ในการควบคุมระดับน้ำสองถังซึ่งควบคุมโดยใช้ระบบควบคุมแบบกระจายตัว (Distributed Control System) ขั้นแรกเราจะใช้การควบคุมเชิงทำนายแบบทั่วไป (Generalized Predictive Control) กับระบบ ซึ่งจะได้ผลตอบสนองช้า ในขณะที่ข้อกำหนดสำคัญของการควบคุมอุตสาหกรรมคือผลตอบสนองที่รวดเร็ว, ส่วนพุ่งเกินต่ำ และเงื่อนไขสภาวะดีของระบบ เพื่อลดระยะเวลาที่ใช้ในการติดตั้ง จากผลของการควบคุมเชิงทำนายแบบทั่วไปนี้ทำให้มีการศึกษาและใช้งานการควบคุมเชิงทำนายแบบขยายในช่วงต่อมา การควบคุมเชิงทำนายแบบขยายปรับปรุงเลขเงื่อนไขสภาวะดีของเมทริกซ์ระบบโดยทำให้จำนวนสภาวะ (condition number) มีค่าลดลงและเพิ่มค่าของตัวกำหนด (determinant) ของเมทริกซ์ระบบโดยการปรับค่าพารามิเตอร์เพียงตัวเดียว ในขั้นแรกสมรรถนะของการควบคุมเชิงทำนายแบบขยายและการควบคุมเชิงทำนายแบบทั่วไปได้ถูกทดสอบโดยใช้การโปรแกรมบน Matlab สำหรับแนววิถีปรับตั้ง (setpoint trajectory) กึ่งกลางของแบบจำลองนามบัญญัติ (nominal model) ผลลัพธ์แสดงให้เห็นว่าการควบคุมเชิงทำนายแบบขยายให้ผลการควบคุมระดับน้ำที่ดีโดยมีผลตอบสนองวงปิดรวดเร็วกว่า และส่วนพุ่งเกินต่ำ สำหรับเงื่อนไขเมทริกซ์น้ำหนักค่าเดียวกัน

ภาควิชา วิศวกรรมไฟฟ้า
สาขาวิชา วิศวกรรมไฟฟ้า
ปีการศึกษา 2551

ลายมือชื่อนิสิต
ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก

##4970765921: MAJOR ELECTRICAL ENGINEERING

KEY WORDS: GENERALIEZD PREDICTIVE CONTROL / ILL CONDITIONALITY / CONDITION NUMBER / EXTENDED PREDICTIVE CONTROL / GPC TUNING / TWO-TANK LEVEL CONTROL / DISTRIBUTED CONTROL SYSTEM / DETERMINENT

SOPHEAK HEL: GENERALIZED PREDICTIVE CONTROL WITH EXTENDED PREDICTIVE CONTROL FOR TWO-TANK LEVEL CONTROL VIA DISTRIBUTED CONTROL SYSTEM, ADVISOR: ASST. PROF, MANOP WONGSAISUWAN, Ph.D., 65 pp.,

Predictive control algorithms emerged in the middle of the seventies from the side of industrial applications. In this thesis, a new predictive control termed extended predictive control (EPC) is implemented on the two-tank level control to control the water level at the middle operation point of each tank via distributed control system (DCS). First the original generalized predictive control (GPC) testing is performed on the plant. The response indicated a slow response of process. On the other hand, the requirement of fast closed-loop response, minimum overshoot with well conditionality is important to the level control industrial for reducing operation setup time. From the results of generalized predictive control, EPC is studied and practically implemented. EPC drastically improved the system matrix conditionality by minimizing condition number and increasing determinant of the system matrix to evaluate a single tuning parameter. First the control performance of EPC and GPC is tested by MATLAB programming for middle setpoint trajectory of the nominal model. The results showed that the EPC provided good control of water levels with faster closed-loop response and minimal overshoot for the same weighted matrix condition.

Department : Electrical Engineering
Field of Study : Electrical Engineering
Academic Year : 2008

Student's Signature ....
Advisor's Signature ....

Acknowledgments

First, I would like to thank my advisor Assistant Professor Manop Wongsaisuwan for his precious help, support, technical guidance, and enthusiastic encourage during those years. I feel deeply indebted to him for his discussion and constructive influence on this orientation of this research throughout my educational endeavor at Control Systems Laboratory, Chulalongkorn University.

I express sincere gratitude to Associate Professor David Banjerdpongchai, Associate Professor Waree Kongprawechnon for their valuable time and service rendered as committee member. My thanks are also extended to my colleagues, Mr. Siripong Permpornsri for his several helpful discussions on implementation via DCS presented in this thesis and his friendship and support.

Many thanks go to all graduate students at Control systems research laboratory, Chulalongkorn University for their helpful discussion, encouragement, and hospitality. The JICA project for AUN-Seed Net is also acknowledged for financial support of my postgraduate studies.

Finally, I would like to express my deepest gratitude to my parents, Chanly Hul and Hel In, for their support and repeated encouragements throughout the preparation of this thesis.

Contents

Page

List of Tables

Page

List of Figures

Page

Page

List of Notations

Symbols

s	Complex variable used in Laplace transform
z^{-1}	Backward shift operator
$\|v\|^2_{\mathcal{Q}_i}$	Weighted norm of vector v with respected to $\mathcal{Q}_i$
$I_{n\times n}$	$(n \times n)$ identity matrix
$\hat{x}(k+j\|k)$	Expected value of $x(k+j)$ with available information at instant k
$u(k)$	Input variable at instant k
$y(k)$	Output variable at instant k
$e(k)$	Discrete white noise with zero mean
$w(k)$	Reference signal at instant k
$A(z^{-1})$	Process left polynomial in the unit delay operator z^{-1}
$B(z^{-1})$	Process right polynomial in the unit delay operator z^{-1}
$C(z^{-1})$	Coloring polynomial in the unit delay operator z^{-1}
N_1	Minimum output horizon
N_2	Maximum output horizon
N_u	Control horizon
$\mathcal{Q}_1$	Output weighting matrix
$\mathcal{Q}_2$	Input weighting matrix
W_{epc}	Weighting matrix for Extended Predictive Control

Acronyms

TITO	Two Input Two Output
CARIMA	Controlled Autoregressive Integrated Moving Average
DCS	Distributed Control System
EPC	Extended Predictive Control
FCS	Field Control Station
FOPDT	First Order Plus Dead Time
GPC	Generalized Predictive Control
HIS	Human Interface Station
MIMO	Multiple Input Multiple Output
MV1	Manipulated Variable 1
MV2	Manipulated Variable 2
NLCS	Network Level Control System
LV1	Level Variable 1
LV2	Level Variable 2
SISO	Single Input Single Output
TTLP	Two Tank Level Process
YFPT	Yokogawa Fieldbus Process Trainer

CHAPTER I

INTRODUCTION

In this first chapter a general background to the problem will be given. This is done by introducing the reader to the areas of predictive control. The discussion will then become more specific as the problems are presented in Section 1.1 and Section 1.2 covers literature review of predictive control and some kind of level process control. In Section 1.3 the scope for this thesis are presented, which deals with some conventions used. The objectives of this thesis is illustrated in Section 1.4. Finally the chapter provides briefly distributions from the thesis.

1.1 Motivation

Since the end of the last century in every corner of industry, the need of new enabling technologies are moving forward to embrace new automation technologies and to unleash the full potential in plant performance. Distributed Control Systems (DCS) can help the practitioners realize this change without compromising the continuity of their operations. On the other hand, some of the most interesting developments have been in the used of predictive control that has been considered as robust optimal control both in industrial application and academic research. It has emerged successfully as powerful practical control technique in process industrial such as power industrial, steam generator, distillation columns, especially refinery manufacture.

With the advantages of control design method, one of the most popular predictive control is Generalized Predictive Control (GPC) that will bring into play the sufficient poles in the industrial implement in the future. However, there are some critical problems in controlling system, where there is often a large gap between the work of academic and industrial practice. One of these criticisms is that the need for designing controller parameters that provides a specified closed-loop response. As it is known, the most industrial plants need to be controlled with high performance that provide faster closed-loop response and still provide a smooth approach to set point with minimal oscillations in the presence of changing operating conditions. Another drawback is the problem of ill conditionality in controlling process. Because of these problems, the new tuning strategy for two-tank level process control via DCS by Extended Predictive Control (EPC) is applied in this research. Finally some complete simulations and experiments of these ideas are exploited for showing the potential of designing generalized predictive controller parameters.

1.2 Literature Review

The literature concerned with the tuning strategy issues for some kind of predictive control and intended to present easy ways of implementation them in the industrial is by now quite extensive.

Therefore this section is devoted to point out some of the most useful starting into the area of tuning parameter of predictive control for some kind of level control processes. Among the more recent the following can be mentioned.

At first, it is a short description summary of level process. The level control have been wildly used in many industrial such as the oil manufacture, liquor transportation, paper making waste water treatment, hydro power plant, and so on. There are some kinds dynamic model of level control. First the Networked Level Control System (NLCS) whose structure is totally different depends on the number of vertical cylindrical tanks with the level controller on each tank. The feed liquid to the first tank comes from a reservoir unit. The output signal from each controller goes to a control valve that set the outflow rate. For NLCS whose structure has three vertical cylindrical tanks, [1] applied a new method called generalized predictive control combined with a queuing-selecting strategy. The main problem for this kind of model is to reduce the network delays which are well known to degrade the performance of control systems or even to destabilize it. A great many of real time control strategy have been developed for this kind of model, such as PID control, predictive control, intelligent control, etc. In our study, we will mention to a coupled tank process existing on the Yokogawa Fieldbus Trainer (YFT). YFT was designed by Yokogawa Ltd and is used in the Control System Research Laboratory (CSRL), Chulalongkorn University. This system will be studied as two-input and two-output (BIBO), where the feed liquids are pumped into the first and second tank come from the reservoir unit. The control problem can be sorted that how to take the level at each tank to the set point levels. For the above process tank, Permpornsri [2] has succeed in applying Model Predictive Controller (MPC) and Multiple Model Adaptive Controller (MMAC). However, few techniques for predictive control have been developed for level control in two-tank level process with reducing ill-conditioning problem in order to achievevdesirable control performance (fast response and minimal oscillation).

There are several surveys available on dealing with predictive control. After provided an application of model predictive control to control a fluid catalytic cracker in 1979 [3,4], a great amount of literature has received considerable attention following the celebrated early use of predictive control. It has been devoted extensively to the investigation on this topic by a large number of researchers and practitioners. One of the most popular predictive controls both in industrial and academic is GPC proposed by Clarke and Mohtadi [5]. It has been successfully implemented in industrial application, showing good performance. It can handle with many control problems for a wide range of plants with reasonable numbers of design variables, which have to be specified by the user depending upon a prior knowledge of the plant and control objectives. As the range of applications for predictive control increase more than 2200 successful applications in industrial [6,7], the demand on the algorithms also increase. The majority of application can be found in petrochemicals, where also consist of level control as well. After recent years of investigation, predictive control has now a strong conceptual and practical framework while several factors are still open and further research. Some of these factors are system identification, better tuning strategy, ill condition, limited control horizon, as well as applicable in new technologies control system.

Tuning of both unconstrained and constrained single-input single-output (SISO) and multiinput multi-output (MIMO) have been addressed by an array of researcher. Generalized Minimum Variance control [8], the idea of this controller is to modify slightly the criterion though the inclusion of a penalty on the control signal as well as on the output. Thus the idea is generally to keep the weighting factor as small as possible in order to remain as close as possible to the objective of maintaining the output variance minimal. The positive weighting factor is included simply to prevent control signal explosion. Properties Generalized Predictive Control [9] there exist straightforward guidelines for the selections of the tuning parameters for the prediction horizon parameters, control horizon parameter, and the weighting control factor which are called principal components. It is still not guaranteed for specific choices of weighting factor even the plant is known perfectly. An analytical expression for move suppression coefficient, for which the impact form is achieved by employing a first-order plus dead time (FOPDT) model approximation of the process dynamic, was derived by Shirdhar and Cooper [10] based on the assumption that the condition number is 500 which was the upper limit of ill conditioning of the system matrix. The extension of multivariable control system is derived by Shirdhar and Cooper. [11] investigated a variable move suppression approach by using a calculated variance of the closed-loop response in order to evaluate the change in a move suppression coefficient at every time instant k. The transient response of this approach is close to critically damping which provides fast response and minimal overshoot. However, how to evaluate and implement the weighting structure appropriately processes with well-condition system is less obvious. Because of this reason, applying EPC to the control level plant via DCS is an important issue.

Building on the work of these past researchers, [12–14], developed a new tuning strategy for tuning SISO and MIMO processes. The tuning term extended predictive control (EPC) uses the condition number and determinant of the system matrix to provide a simple and effective tuning procedure. The condition number of the system matrix represents the degree to which the inevitability of this matrix is sensitive to the parameters of the controlled variable. The structure of GPC is essentially a pseudoinverse of the system matrix of the plant, and therefore, this measure of control design is considered. The lower condition number the better the controllability and control performance of the plant. The contribution of that earlier work was an analytical expression to compute the condition number of the system matrix. This work demonstrated a fundamental change in the tuning of predictive controller. It focuses on further development of EPC for multivariable systems. The tuning strategy presented in this work addresses this important issue of robustness in the final MIMO EPC design. These parameters include sampling interval, control horizon.

The main research in this thesis, we aim to apply a simple and effective tuning strategy approach that results in a well conditioned system to control the two-tank level system by GPC and EPC. The problem of design controller is to provide well-condition systems. The main feature of design procedure uses the condition number of the system matrix to evaluate a single tuning parameter which is the well-know infinite control horizon resulting in powerful stability for nominal case. In addition it has the potential to provide real quality of application and performance benefits on level control process. The results will illustrate with simulation and experiment that the proposed technique has a relatively small computational cost.

1.3 Objectives

The primary objective of this research is to investigate and implement both generalized predictive control and a new tuning strategy for multivariable extended predictive control for two tanks level control via distributed control system. First the nominal plant model is simulated. In particular, these tuning methods are applicable to unconstrained multivariable processes.

1.4 Scope of Thesis

The system designed by this research will have specification as follows

1. This thesis deal with TTLP designed by Yokogawa Ltd via DCS
2. To study GPC and EPC tuning strategy
3. Implement both GPC and EPC to control TTLP
4. In addition, some comparisons between GPC and EPC is presented

1.5 Methodology

1. CENTUM CS3000 R3 connected to the Yokogawa Fieldbus Process Trainer (YFPT) is used to control and monitor with TTLP
2. FOPDT model approximation of the process dynamic is used to derive the tuning parameters for EPC
3. A minimized condition number and increased determinant approach of the system matrix is considered simultaneously to provide the well-condition system
4. The computational tools used in this thesis are MATLAB and SEBOL solver

1.6 Contributions

The expected contributions from this thesis are:

1. Implementation of two controller GPC and EPC on TTLP via DCS
2. Better control performance of EPC for TTLP
3. Understand how to control a process via DCS

CHAPTER II

TWO-TANK WATER LEVEL AND DCS

The topic of this chapter is to present the water process tank and distributed control system, in order to find out how it should be merged. In Section 2.2 describes the physical structure of the two tank water level that placed on YFPT. In order to deal with implementation next chapter, the identified model of the TTLP is also given. After the model of the system is given DCS is introduced briefly in Section 2.3. Finally the chapter is concluded

2.1 Introduction

Process control has become increasingly important in the process industrial application along with heat transfer, fluid mechanics and liquid level control. As a consequence of changing economic conditions and technological bases have moved rapidity over last 30 years. Process control is also played as an important role in the development of complex processes for manufacturing with safe and efficient operation. Furthermore, the rapidly increased computer speed and modern control systems have enabled high performance measurement to become an essential part of industrial plants.

In addition, for reasons of cost effectiveness, modern instrument systems in manufacturing, increasingly use network technologies and also increasingly sophisticated devices. The digital process control system or another word is Distributed Control System (DCS) is widely used in the field of industrial process. DCS refers to control system in which the controller are distributed throughout the system with each component. The entire system of controllers are connected by networks for communication and monitoring. Moreover, it come with an embedded operating system, Ethenet port and a web interface for configuration where the operator workstation is typically a window platform.

It is clear that the scope and importance of process control technology will continue to expand during the last decade. Consequently, one of the important themes that we emphasize is the need for control engineering graduate to understand some knowledge not only the traditional control theories but also the modern control technologies that is to say how to operate and design modern plants. The way the plant is successfully designed and controlled by using theories but sometime it has a large impact on how it should be controlled and what level of control performance can be obtained in the real system.

In this thesis, we provide an appropriate balance of process control theory and practice. In particular, this study will focus on controlling the ingrated two-tank water level via modern digital control system which is called distributed control system (DCS). It emphasizes physical and empirical modeling of water level plant, measurement and DCS.

2.2 Description of The Two-Tank Level

The control of liquid levels, for example in a process tank, is an important function. In our study is a process two tank water level, shown in the Fig. 1. It is an interconnected cylindrical tank system operating in parallel, that has been designed by Yokogawa for process trainer in laboratory, Chulalongkorn University. It has two control valves CV1 and CV2, and it is automatic valve operation.

The structure consists of two cylindrical tanks and two pneumatic actuated valves. The two vertical cylinders are 1.2 meters and constructed from plastic, whereas diameter and cross sectional area of each tank is 20 cm and 315.85 cm^2 respectively. These two tanks are connected to each other via a constant flow rate solenoid valve. The amount of water from tank 1 going to reservoir unit can be adjusted the outflow rate via manual valve.

Two pneumatic valves are used for the actuation of the process liquid. These actuators translate an air signal into valve stem motion by air pressure acting on a diaphragm or piston connected to the stem. The air pressure is supplied at 0.4 PSI by the air pump. Pneumatic actuators are used in throttle valves for open-close positioning where fast action is required. When air pressure closes the valve and spring action opens the valve, the actuator is termed direct acting. When air pressure opens the valve and spring action closes the valve, the actuator is termed reverse acting. The first position vale provides water flow rate to the first tank and The second position vale provides water flow rate to the second tank.

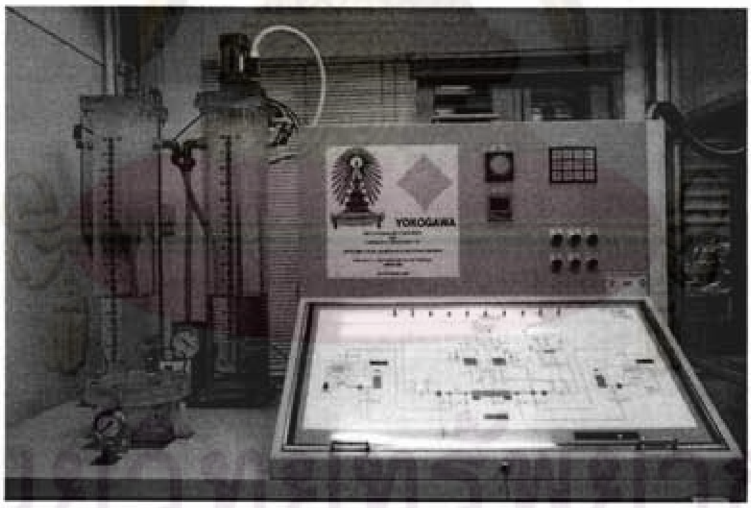

Figure 2.1: Yokogawa Fieldbus Process Trainer

2.2.1 Workspace Description

A level-control system in which the level is a major variables to be controlled, the system must be designed to satisfy the static and dynamic control objectives. These objectives can be focused on the acceptable deviations with respect to a level set points staying within the allowable maximum overshoot and undershoot and meeting system setting time requirement. Fig. 2.2 shows a sketch of the process configuration and its control system. Fig. 2.2, it consists of two vertical cylindrical tanks

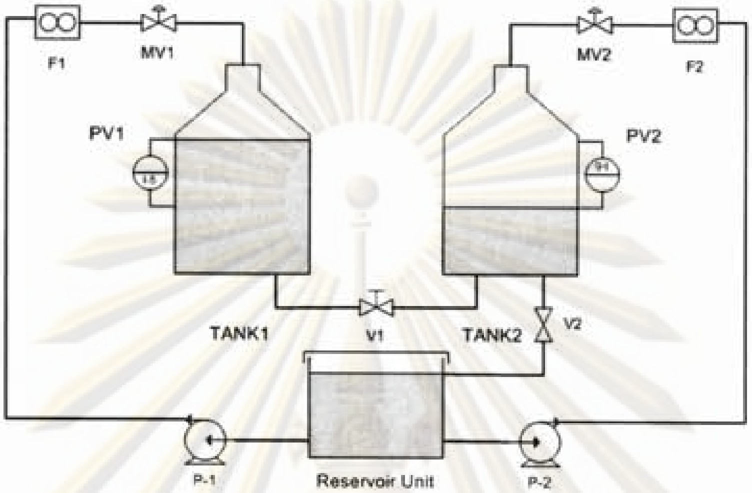

Figure 2.2: Dynamic model structure

with the level controller at each tank. The flow rates of vale V_1 and V_2 are fixed. The liquid level in each of the tank PV_1 and PV_2 are controlled by manipulating the flow rate F_1 and F_2 of feed liquid pumped from the reservoir into the first and second tank respectively. The level signal from the level transmitter on each tank is sent to a level controller. The output signal from each controller goes to a control valve that sets the outflow rates. The flow rates F_1 and F_2 are set by manipulated variable MV_1 and MV_2 from the two level controller. The process variable signal PV_1 and PV_2 from the two level transmitters depend on the two liquid levels H_1 and H_2. In this case of study we express these PV_i and MV_i signal as fractions of the full scale of the signals. The signal from transmitter and controller are voltage or current signals, which vary over standard ranges (0 to 10V or 4 to 20mA).

$$F_1 = MV_1 F_1^{max} \tag{2.1}$$

$$F_2 = MV_2 F_2^{max} \tag{2.2}$$

where F_n^{max} = flow rate when the control valve is wide open.

Numerical values of all parameters and the values of the variables at the initial steady-state conditions are given in Table 2.1.

2.2.2 Transfer Function

In the experimental system, the input are the flow rares and the output are the level of each tank. In other word, the pneumatic actuators and the sensor measurements. We define the flow rate vector $\delta\mathbf{MV}$ and the measurement vector $\delta\mathbf{PV}$. The manipulated variables and controlled variables view of the actuated system is illustrated in the Fig. 2.3.

Table 2.1: Dynamic model structure

Parameters	Value
Diameter of tank	20 cm
Cross-sectional area of tank	$315.85cm^2$
Span of level transmitters	90 cm
High of tank 1	120 cm
High of tank 2	120 cm

$$\delta\mathbf{MV} = \begin{bmatrix} \delta MV_1 \\ \delta MV_2 \end{bmatrix}, \qquad \delta\mathbf{PV} = \begin{bmatrix} \delta PV_1 \\ \delta PV_2 \end{bmatrix} \tag{2.3}$$

Figure 2.3: Input and Output of the system

In order to establish the transfer function of the process around the operating point. The actuators are independently controlled by proportional controller [2] , where small-signal analysis is used, as illustrated in the Fig. 2.4. The resulting information shall be used for the validation of the process model.

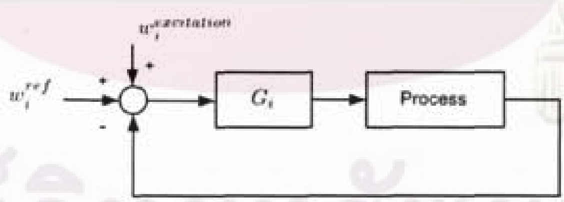

Figure 2.4: Independent SISO controller for each actuator

The model of two-tank system is

$$\begin{bmatrix} \delta PV_1 \\ \delta PV_2 \end{bmatrix} = \begin{bmatrix} \dfrac{K_{11}\, e^{-\theta_{11}\, s}}{\tau_{11}\, s+1} & \dfrac{K_{12}\, e^{-\theta_{12}\, s}}{\tau_{12}\, s+1} \\ \dfrac{K_{21}\, e^{-\theta_{21}\, s}}{\tau_{21}\, s+1} & \dfrac{K_{22}\, e^{-\theta_{22}\, s}}{\tau_{22}\, s+1} \end{bmatrix} \begin{bmatrix} \delta MV_1 \\ \delta MV_2 \end{bmatrix} \tag{2.4}$$

where K_{ij} is gain
θ_{ij} is time delay
τ_{ij} is time constant of systems
δPV_i is small change of output (liquid level)
δMV_i is small change of input (position of control valve)
i is the number of output
j is the number of input

2.2.3 Model of The Plant

In the real practice it is impossible to find the model which completely describes the true system. This is why the model validation is very important part of system identification experiment. Model validation approach focuses on the important question of finding out whether the estimated model is suitable for its intended purpose, which in this thesis is to calculate j-step ahead predictions. There are several standard methods for model validation describes.

Permpornsri achieved an experimental validation of assumed model for two-tank liquid control by Non-Parametric Method or other word Step Response Analysis. The definition of appropriate boundary conditions for each tank turned out to be controlled with three difference operating points which was classified as low, middle and high level model, as shown on the Table (2.2). In order to improve the quality of the assumed-model, he applied and validated a Residuals Analysis. On this basic, improved shapes of the assumed-modes were obtained by component-mode synthesis. Despite small discrepancies observed with respect to experimental data, the assumed-mode model was sufficiently accurate for control design.

Table 2.2: Model Parameters for TTLP

Water Level	Controlled Variable δPV_1			Controlled Variable δPV_2		
	K_{11}	τ_{11}	θ_{11}	K_{21}	τ_{21}	θ_{21}
Low	3.8	382	14	3.9	472	6
Middle	**3.2**	**452**	**7**	**3.1**	**472**	**32**
High	3.2	651	7	3.15	520	24
	K_{12}	τ_{12}	θ_{12}	K_{22}	τ_{22}	θ_{22}
Low	5.35	443	15	6.5	430	10
Middle	**4.15**	**416**	**32**	**4.9**	**443**	**4**
High	6.8	568	20	7.6	528	6

For this thesis, we consider only the middle operating point where the model transfer matrix is defined as

$$\begin{bmatrix} \delta PV_1 \\ \delta PV_2 \end{bmatrix} = \begin{bmatrix} \dfrac{3.8\,e^{-7s}}{452\,s+1} & \dfrac{4.15\,e^{-32s}}{416\,s+1} \\ \dfrac{3.1\,e^{-32s}}{472\,s+1} & \dfrac{4.9\,e^{-4s}}{443\,s+1} \end{bmatrix} \begin{bmatrix} \delta MV_1 \\ \delta MV_2 \end{bmatrix} \tag{2.5}$$

2.2.3.1 Discrete Plant Model

It is noted that the plant to be controlled is a first-order plus dead time (FOPDT) model with the transfer function of the form

$$G(s) = \frac{K}{\tau + 1} e^{-\theta s} \tag{2.6}$$

Moreover, the generalized predictive control algorithm used Auto-Regressive Moving-Average (CARIMA) model. For this reason the continuous time model above need to be discretized.

- Case 1: The dead time d is an integer multiple of the sampling time T_s

The corresponding discrete transfer function of Eq. (2.6) has the form

$$G(z^{-1}) = \frac{bz^{-1}}{1 - az^{-1}} z^{-d} \tag{2.7}$$

where a, b and d can be derived from continuous parameters of Eq. (2.6), resulting the following expressions [15]:

$$\begin{cases} a = e^{-T_s/\tau} \\ b = K(1-a) \\ d = \theta / T_s \end{cases}$$

- Case 2: The dead time d is not an integer multiple of the sampling time T_s

In this case the Padè Approximation can be used and the plant discrete transfer function can be written as:

$$G(z^{-1}) = \frac{b_0 z^{-1} + b_1 z^{-2}}{1 - az^{-1}} z^{-d} \tag{2.8}$$

The following relation can be used to obtain the discrete parameters [15]

$$\begin{cases} \tau = T_s(\theta + \epsilon) & \text{for } 0 < \epsilon < 1 \\ a = e^{-T_s/\tau} \\ b_0 = K(1-a)(1-\alpha) \\ b_1 = K(1-a)\alpha \\ \alpha = \frac{a(a^{\epsilon}-1)}{1-a} \end{cases}$$

Therefore, the discretized model of the sampling time $T_s = 14$ sec at the middle level is

$$\begin{bmatrix} \delta PV_1(k) \\ \delta PV_2(k) \end{bmatrix} = \begin{bmatrix} \dfrac{0.0492 + 0.0484\,z^{-1}}{1 - 0.9695\,z^{-1}} & \dfrac{0.0986 + 0.388\,z^{-1}}{1 - 0.9669\,z^{-1}} \\ \dfrac{0.0650 + 0.0256\,z^{-1}}{1 - 0.9708\,z^{-1}} & \dfrac{0.1094 + 0.0431\,z^{-1}}{1 - 0.9689\,z^{-1}} \end{bmatrix} \begin{bmatrix} \delta MV_1(k) \\ \delta MV_2(k) \end{bmatrix} \tag{2.9}$$

Table 2.3: Operating Point

Operation	Value
Set point signal of controller SP1	0.45 fraction of full scale
Set point signal of controller SP2	0.65 fraction of full scale
Deviation value of controller output δMV_1	$0 - 100$ fraction of full scale
Deviation value of controller output δMV_2	$0 - 100$ fraction of full scale
Small change value of level transmitter output δPV_1	$0.45 - 0.65$ fraction of full scale
Small change value of level transmitter output δPV_2	$0.45 - 0.65$ fraction of full scale

2.3 Distributed Control System

Distributed control systems (DCS) use decentralized elements or subsystems to control distributed processes or complete manufacturing systems. They provide control and supervision of production processes by connecting operator interaction and machines. With the brief introduction of reliability, flexibility and compatibility of DCS, Yokogawa integrated production control system, CENTUM CS3000 R3, is used to monitor and control the operation of process tank for this thesis. As shown in the Fig. 4, CENTUM CS300 R3 consist of a field control station (FCS), communications medium, and human interface station (HIS). The communications medium in a CENTUM CS300 R3 is a wired which connects the FCS to HIS.

From the system configuration drawing we can classify DSC structure into 3 main components as following

- Human Interface Station
- Field Control Station
- Process Input Output

2.3.1 Human Interface Station

A handy and reliable operation environment that combines the latest Windows technology with the functionality of the CENTUM CS3000 R3. The widely praised graphic window, trend window and CENTUM CS3000 R3 operation windows are provided. In the operation and monitoring environment, there are two types of screen modes. Operators can select either Multiple-window mode, like an office PC, or Full-screen mode, like a legacy operator console. As in our study we used an office PC. The collection of data for display on the HIS is event-driven, i.e., The HIS starts collecting data only when the corresponding window is called up. Thus only the dynamic data in the currently displayed windows need to be refreshed. The data sampling load on the HIS is reduced and a one-second refresh rate for the dynamic data in the windows is guaranteed.

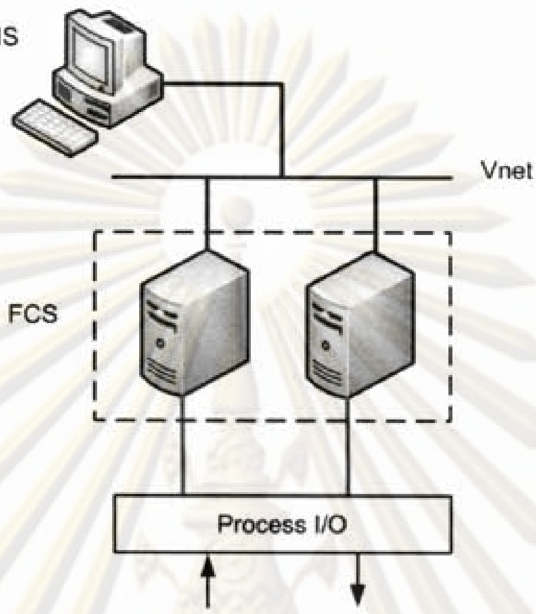

Figure 2.5: System Configuration Drawing

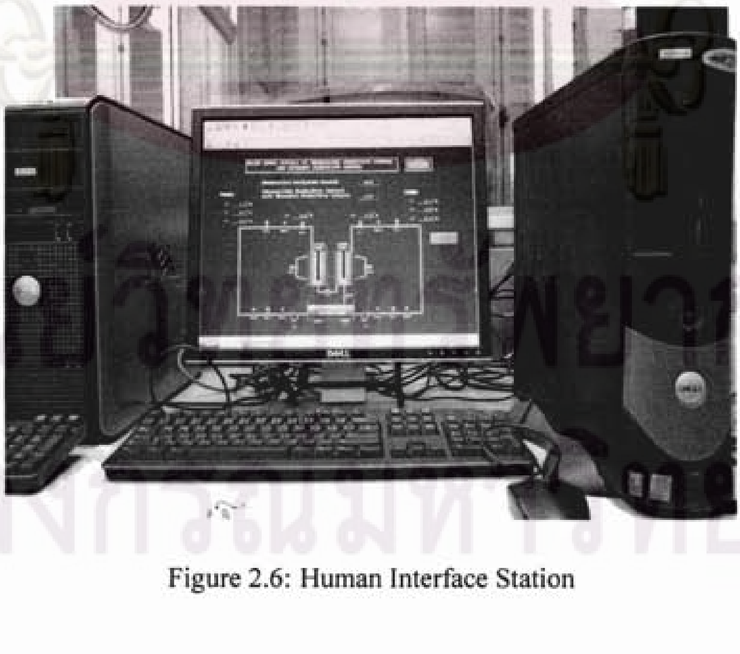

Figure 2.6: Human Interface Station

2.3.2 Field Control Station

The FCS inherits the renowned reliability of the CENTUM CS3000 R3 and is truly the most reliable FCS in the industry. The FCS features a fully redundant architecture for V-net communications, power supply, and CPU. While the FCS is running, the stand-by processor module performs control computations synchronously with the processor module in service, so control is always maintained even while switching over between the duplexed control processors (hot stand-by system). The same process I/O modules are often used with the CS3000 systems in many plants and have the highest reliability. The modules are mainly the isolated channel type.

Figure 2.7: Field Control Station

2.3.3 Process Input and Output

The same process I/O modules are often used with the CS3000 systems in many plants and have the highest reliability. The modules are mainly the isolated channel type and use M4 screw terminals.

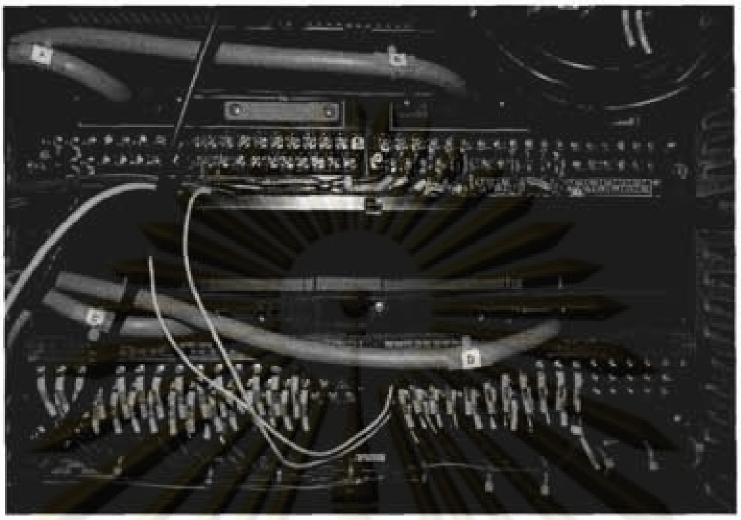

Figure 2.8: Field Control Station

2.4 Conclusion Remark

In this chapter the dynamic structure of TTLP placed on the YFPT is described. It can be seen that plant model which is very useful in next chapter is obtained by using non-parametric method. Moreover, the brief introduction of the principle components for DCS is also given. TTL will be connected to DCS in order to show the performance in controlling the system for GPC and EPC.

CHAPTER III

GPC AND EPC PREDICTIVE CONTROLLER

This chapter provides the principle to the area of predictive control. First there is an introduction discussing the idea in generalized predictive control in Section 3.1. It also gives a proper mathematical formulation to be used throughout this thesis. The non-adaptive tuning strategy will be presented in this Section. This will help to select the principle parameters. In Section 3.2 the new predictive control termed extended predictive controller is illustrated. Finally the chapter is concluded in order to illustrate the ideas of predictive control.

3.1 Generalized Predictive Control

Camacho and Bordons [15] gives a thorough treatment of the subject and explanation the general ideas of a long range input-output generalized predictive control (GPC) that is described as an effective multivariable control algorithm in which a dynamic optimization problem is solved on-line. The control signal sequences are obtained by minimizing the performance index at each sampling time to optimize future output on a finite fixed horizon. A new problem is solved for the first control of the optimal control sequences, where its theoretical analysis in order to assess the influence of the design parameters such as minimum prediction horizon, maximum prediction horizon, control horizon, and move suppression coefficient on the closed loop stability and desired performance. It can only be done for very specific choices of these design parameters.

3.1.1 Performance Index

Limitations of this thesis will only deal with the unconstrained case of predictive control. For the treatment of constraint case can see [15]. We alter the finite horizon criterion, which is the most commonly used in a quadratic criterion of some kind. In Fig. 3.1 a schematic picture of the basic ideas of predictive control is presented. In our research the quadratic cost function will be used

$$J_k(u) = \sum_{j=N_1}^{N_2} \|\hat{y}(k+j|k) - w(k+j|k)\|^2_{Q_1} + \sum_{j=1}^{N_u} \|\Delta u(k+j-1)\|^2_{Q_2} \tag{3.1}$$

where $\hat{y}(k+j|k)$ denotes an optimum j step ahead prediction of the system output on data up to time k, predicted based on the past input, output vectors and the future control sequences. $w(k+j|k)$ is the predicted future reference trajectory and $\Delta u(k+j-1)$ denotes an incremental control input of the system at time $k+j-1$, computed by minimizing (3.1) at time k. Q_1 and Q_2 are positive definite weighting matrices.

The cost function (3.1) is minimized to produce $\Delta u(k), \Delta u(k+1), \cdots, \Delta u(k+N_u-1)$. For the next sampling time the minimized criterion is repeated. This implementation is called *Receding*

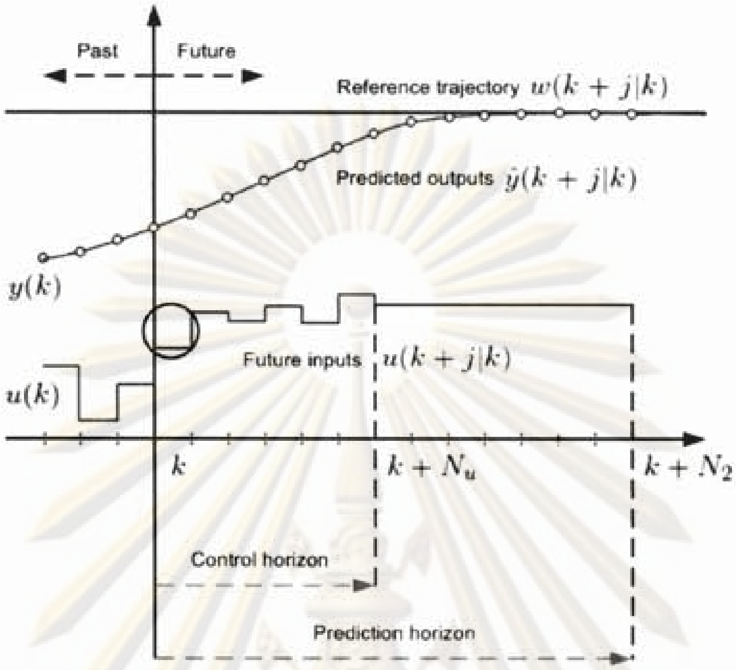

Figure 3.1: GPC Scheme

Horizon Control. The modification made to this control signal not in terms of $u(k)$ but in terms of $\Delta u(k)$ is that it is the changes in the input signal that are unwanted or another word is that the statement of optimization problem. If $\Delta = 1 - z^{-1}$ acts upon $u(k)$ the result will be

$$\Delta u(k) = u(k) - u(k-1) \tag{3.2}$$

As can be seem this cost function penalizes differences between the predicted outputs from the system, $\hat{y}(k+j|k)$ and the reference trajectory $w(k+j|k)$ over the prediction horizon, where $j = N_1, N_1+1, \cdots, N_2$. It also penalizes changes in the control signal $\Delta u(k+j-1)$ over the control horizon. The exact form of these penalizations can be controlled by using the weighting matrices $\mathcal{Q}_1$ and $\mathcal{Q}_2$ which in most case are diagonal. The control criterion is that it possesses a significant level of complexity sufficient to make it capable of producing effective controllers for enormous ranges of candidate plants while the criterion itself depends upon the specification of only main design parameters weighting matrix $\mathcal{Q}_2$. However, due to the fact that the shape of the design parameter used in this thesis requires one tuning parameter with well condition of weighted system matrix. In that case the system matrix weighted by $\mathcal{Q}_2$ need to be modified by EPC tuning procedure which is being studied in Section 3.2 and also is entirely devoted to the problem used in this thesis.

3.1.2 Model Description

The model representation to be used in this thesis will mostly be discrete-time, linear and time-invariant models with uncertainties and are called noise inputs appearing on the feedback loop. A model of the process described through the CARIMA model can be expressed as

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + \frac{1}{\Delta}C(z^{-1})e(k) \tag{3.3}$$

where $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ are polynomials in the backward shift operation z^{-1} defined as:

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b} \\ C(z^{-1}) &= 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_{n_c} z^{-n_c} \end{aligned}$$

The operator Δ is defined as $\Delta = 1 - z^{-1}$. The variables $y(k)$, $u(k)$ and $e(k)$ are the output,input and noise vector at time k respectively. The noise variable $e(k)$ affects the output and are assumed to be white stationary random noise with $E[e(k)] = 0$, $E[e(k)e(k)^T] = \Gamma_e$.

- A deterministic associated to the relationship between inputs and outputs given by the polynomial $A(z^{-1})$ and $B(z^{-1})$
- A stochastic associated to the relationship between noise variable $e(k)$ and the output given by the polynomials $A(z^{-1})$ and $C(z^{-1})$. This part is called noise model.

3.1.3 Control Law

To solve the problem posed by the minimization of (3.1) in order to obtained unconstrained GPC control law, we have to compute a set of j-step ahead predictions of the outputs over the prediction horizons based on information known at time k and on the future values of the control increments. Moreover, the following Diophantine equation need to be used

$$1 = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \tag{3.4}$$

The E_j and F_j are polynomials uniquely defined given A and the prediction interval j

$$\begin{aligned} F_j(z^{-1}) &= f_{j,0} + f_{j,1} z^{-1} + f_{j,2} z^{-2} + \cdots + f_{j,n_a} z^{-n_a} \\ E_j(z^{-1}) &= e_{j,0} + e_{j,1} z^{-1} + e_{j,2} z^{-2} + \cdots + e_{j,j-1} z^{-(j-1)} \end{aligned}$$

It can easily be seen that the initial conditions for the recursion equation are given by:

$$\begin{aligned} E_1 &= 1 \\ F_1 &= z(1 - A(z^{-1})\Delta) \end{aligned}$$

Because of the predictive nature of model predictive controllers, time delays are inherently considered by them. The dead times from input to the output are reflected in the polynomial matrix

$$B(z^{-1}) = z^{-d}B'(z^{-1}) \tag{3.5}$$

The lower limit of the prediction horizon N_1 can therefore be made equal to $d+1$. Therefore the recursion of the Diophantine Equation is used.

If equation (3.3) is multiplied by z^{-j}, we then obtain

$$y(k+j) = \frac{B}{A}u(k+j-d-1) + E_j e(k+j) + \frac{F_j}{A\Delta}e(k) \tag{3.6}$$

If we replace $e(k)$ using (3.3), this yields

$$y(k+j) = F_j y(k) + E_j B\Delta u(k+j-d-1) + E_j e(k+j) \tag{3.7}$$

where the last term contains information which is independent from signals measurable at time k. It is then obvious that the minimum variance prediction of $\hat{y}(k+j|k)$ given data know at time k is obtained by replacing the last term by zero, yielding

$$\hat{y}(k+j|k) = F_j y(k) + E_j B\Delta u(k+j-d-1) \tag{3.8}$$

In this expression $\hat{y}(k+j|k)$ is a function of know signal values at time k and also of future control inputs which have yet to be computed. We then use a second Diophantine equation to distinguish between past and future control values,

$$E_j(z^{-1})B(z^{-1}) = G_j(z^{-1}) + z^{-j}Gp_j(z^{-1}) \tag{3.9}$$

which yield the following expression for the prediction,

$$\begin{aligned}
\hat{y}(k+d+1|k) &= G_{d+1}\Delta u(k) + Gp_{d+1}\Delta u(k-1) + F_{d+1}y(k) \\
\hat{y}(k+d+2|k) &= G_{d+2}\Delta u(k) + Gp_{d+2}\Delta u(k-1) + F_{d+2}y(k) \\
&\vdots \\
\hat{y}(k+d+N_2|k) &= G_{d+N_2}\Delta u(k) + Gp_{d+N_2}\Delta u(k-1) + F_{d+N_2}y(k)
\end{aligned}$$

The vector of predicted controlled plant outputs can be written as

$$\mathbf{y} = \mathbf{Gu} + \mathbf{F}(z^{-1})y(k) + \mathbf{Gp}(z^{-1})\Delta u(k-1) \tag{3.10}$$

Equivalently,

$$\mathbf{y} = \mathbf{Gu} + \mathbf{f} \tag{3.11}$$

Note that the vector f is the free response prediction of $y(k+j)$ by assuming that control increments after time $k-1$ will be zero, where its element term can be calculated recursively by

$$f(k+j) = Gp_j(z^{-1})\Delta u(k-1) + F_j(z^{-1})y(k) \tag{3.12}$$

where

$$\begin{aligned}
\mathbf{y} &= \begin{bmatrix}\hat{y}(k+d+1|k) & \hat{y}(k+d+2|k) & \cdots & \hat{y}(k+N_2|k)\end{bmatrix}^T \\
\mathbf{f} &= \begin{bmatrix}f(k+d+1|k) & f(k+d+2|k) & \cdots & f(k+N_2|k)\end{bmatrix}^T
\end{aligned}$$

The dynamic matrix $\mathbf{G}$ is composed of the impulse response parameters, g_j, of the pant model of N_2 controlled variables and N_u manipulated variables is

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_u-1} & g_{N_u-2} & \cdots & g_0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2-1} & g_{N_2-2} & \cdots & g_{N_2-N_u} \end{bmatrix}_{N_2 \times N_u} \tag{3.13}$$

An outline is given of how to compute the GPC control law. For simplicity of notation the quadratic minimization of (3.1) now becomes a direct problem of linear algebra, with substituting the predicted output from (3.1.3) into (3.14)

$$J_k(u) = (\mathbf{Gu} + \mathbf{f} - \mathbf{w})^T(\mathbf{Gu} + \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T\mathbf{u} \tag{3.14}$$

The required minimization after expanding (3.14)

$$J_k(u) = \mathbf{u}^T(\mathbf{G}^T\mathbf{G} + \lambda I)\mathbf{u} + 2\mathbf{u}^T\mathbf{G}^T(\mathbf{f} - \mathbf{w}) \tag{3.15}$$

Note that the performance index is quadratic and hence has a unique minimum which therefore can be located by setting the first derivative to zero:

$$\begin{aligned} \frac{dJ}{d\mathbf{u}} &= 2(\mathbf{G}^T\mathbf{G} + \lambda I)\mathbf{u} + 2\mathbf{G}^T(\mathbf{f} - \mathbf{w}) \\ \frac{dJ}{d\mathbf{u}} &= 0 \end{aligned} \tag{3.16}$$

Therefore, the vector of optimum future control increments $\mathbf{u}$ can be expressed as

$$\mathbf{u} = (\mathbf{G}^T\mathbf{G} + \lambda I)^{-1}\mathbf{G}^T(\mathbf{f} - \mathbf{w}) \tag{3.17}$$

where

$$\begin{aligned} \mathbf{w} &= \begin{bmatrix} w(k+d+1|k) & w(k+d+2|k) & \cdots & w(k+N_2|k) \end{bmatrix}^T \\ \mathbf{u} &= \begin{bmatrix} \Delta u(k) \\ \Delta u(k) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix}_{N_u \times 1} \end{aligned}$$

Most industrial plant have many variables that have to be controlled and many manipulated variables or variables used to control the plant. In many cases, when one of the manipulated variables is changed, it is not only affects the corresponding controlled variables but also upsets the other variables. As in this thesis, we study about how to control the two tanks level process which consists of two inputs and two outputs (TITO). Let us define the vector of predicted controlled plant outputs

$\mathbf{y} = [\mathbf{y}_1^T \quad \mathbf{y}_2^T]^T$ or equivalent to

$$\mathbf{y} = \begin{bmatrix} \hat{y}_1(k+d_1+1) \\ \vdots \\ \hat{y}_1(k+d_1+N_2) \\ \hat{y}_2(k+d_2+1) \\ \vdots \\ \hat{y}_2(k+d_2+N_2) \end{bmatrix}_{2N_2 \times 1} \tag{3.18}$$

The following expression of prediction $\hat{y}_1(k)$ and $\hat{y}_2(k)$ are defined

$$\begin{aligned} \hat{y}_1(k+d_1+j|k) &= \sum_{i=1}^{2} G_{1i,j}(z^{-1})\Delta u_i(k+j) + \sum_{i=1}^{2} G_{1i,j}(z^{-1})\Delta u_i(j-1) \\ & \quad F_{1j}(z^{-1})y_1(k) \qquad (3.19) \\ \hat{y}_2(k+d_2+j|k) &= \sum_{i=1}^{2} G_{2i,j}(z^{-1})\Delta u_i(k+j) + \sum_{i=1}^{2} G_{2i,j}(z^{-1})\Delta u_i(j-1) \\ & \quad F_{2j}(z^{-1})y_2(k) \qquad (3.20) \end{aligned}$$

Let us define $\mathbf{f}$ as the free response of $\mathbf{y}$

$$\mathbf{f} = \begin{bmatrix} f_1(k+d_1+1) \\ \vdots \\ f_1(k+d_1+N_2) \\ f_2(k+d_2+1) \\ \vdots \\ f_2(k+d_2+N_2) \end{bmatrix}_{2N_2 \times 1} \tag{3.21}$$

Equivalently,

$$f_1(k+d_1+j) = \sum_{i=1}^{2} Gp_{1i,j}(z^{-1})\Delta u_i(k+j) + F_{1j}(z^{-1})y_1(k) \tag{3.22}$$

$$f_2(k+d_2+j) = \sum_{i=1}^{2} Gp_{2i,j}(z^{-1})\Delta u_i(k+j) + F_{2j}(z^{-1})y_2(k) \tag{3.23}$$

The optimum control law without constraint can be expressed as:

$$\mathbf{u} = (\mathbf{G}^T \mathcal{Q}_1 \mathbf{G} + \mathcal{Q}_2)^{-1} \mathbf{G}^T \mathcal{Q}_1 (\mathbf{w} - \mathbf{f}) \tag{3.24}$$

where the weighting matrix $\mathcal{Q}_1$, $\mathcal{Q}_2$ and reference signals $\mathbf{w}$ are defined as:

$$\mathbf{w} = \begin{bmatrix} \hat{w}_1(k+d_1+1) \\ \vdots \\ \hat{w}_1(k+d_1+N_2) \\ \hat{w}_2(k+d_2+1) \\ \vdots \\ \hat{w}_2(k+d_2+N_2) \end{bmatrix}^T \quad (3.25)$$

$$\mathcal{Q}_1 = \begin{bmatrix} \gamma_1 I_{N_u \times N_u} & 0 \\ 0 & \gamma_2 I_{N_u \times N_u} \end{bmatrix}_{2N_2 \times 2N_2} \quad (3.26)$$

$$\mathcal{Q}_2 = \begin{bmatrix} \lambda_1 I_{N_2 \times N_2} & 0 \\ 0 & \lambda_2 I_{N_2 \times N_2} \end{bmatrix}_{2N_u \times 2N_u} \quad (3.27)$$

This expression will be accurate if the $\mathbf{G}^T \mathcal{Q}_1 \mathbf{G} + \mathcal{Q}_2$ matrix is positive definite (minimum of the performance index). To ensure this fact, $\gamma_i I_{N_u \times N_u}$ weighting matrices must be positive definite, $\lambda_i I_{N_2 \times N_2}$ weighting matrices must be non-negative definite and dynamic matrix G should be of full rank.

Because of the receding control strategy, only $\Delta u(k)$ is need at time k. Thus only the first m rows of $\mathbf{G}^T \mathcal{Q}_1 \mathbf{G} + \mathcal{Q}_2)^{-1} \mathbf{G}^T \mathcal{Q}_1$ have to be computed.

The control law can then be expressed as:

$$\Delta u(k) = (\mathbf{G}^T \mathcal{Q}_1 \mathbf{G} + \mathcal{Q}_2)^{-1} \mathbf{G}^T \mathcal{Q}_1 (\mathbf{w} - \mathbf{f}) \quad (3.28)$$

3.1.4 Choice of parameters

For the real application the principal parameters such as N_1, N_2, N_u and λ_i for $i = 1, 2$ should best be selected in order to have desired performance.

Let consider the case where the process can now be interpreted as process with a common minimum delay d_i which is associated to the outputs y_i for $i = 1, 2$. Therefore, the lower limit of the prediction horizon $N_{1,i}$ can be made equal to $d_i + 1$. If d_i is not know or variables, then $N_{1,i}$ can be set to 1. The maximum output horizon N_2 should be chosen corresponding more closely to the rise time of the plant. The general acceptable control of $N_u = 1$ for the simple plant is set. Increasing N_u make the control and corresponding output response more active.

In addition to the specific choice of controller parameters for closed-loop stability, all the transfer function matrices consist of the nominal model and the uncertainty matrix that includes the model uncertainties (noises), must be satisfy the performance criterion. To achieve the desired performance of fast closed-loop response without oscillatory transients, we provide a modification of the weighting matrices $\mathcal{Q}_2$ technique which ensure the well-conditioned system [12].

3.2 Extended Predictive Control

In the process industry typical plants are fast reacting and need to get to set point in a relatively short time with fast settling times. In order to overcome the problem of providing fast closed-loop responses, Abu-Ayyad, Dubay and Kember [12] developed a new simple tuning strategy termed extended predictive control (EPC). This work focuses on an effective strategy of predictive control by reformulating the control law with a new structure of a weighting matrix. An exact method is to evaluate a single tuning parameter based on the optimal range of the condition number and also the increasing determinant of the system matrix. In this thesis the comparison of the control performance of EPC with GPC for level process is investigated.

The general predictive control law is the solution of optimization performed by infinite horizon stable predictive control cost (3.1) with this most of the algorithms using. It formulated with positive weighting factors λ on the manipulated variable moves. Thus the idea is generally to prevent unconstrained optimal predictive control signal explosion. The closed form solution of quadratic cost function without weighting on $\mathbf{G}^T\mathbf{G}$ is computed as

$$\mathbf{u} = (\mathbf{G}^T\mathbf{G} + \lambda I)^{-1}\mathbf{G}^T\mathbf{e} \tag{3.29}$$

where $\mathbf{e}$ is the vector of tracking difference between the trajectory reference and the prediction of the process.

3.2.1 New Structure for EPC

By introducing a new structure of the weighting factor matrix W_{new} which must be symmetric in order to keep the systemetricity of the system matrix $\mathbf{G}^T\mathbf{G}$, the beginning of reformulation of the new control strategy is considered.

$$\mathbf{u} = (\mathbf{G}^T\mathbf{G} + W_{new})^{-1}\mathbf{G}^T(\mathbf{f} - \mathbf{w}) \tag{3.30}$$

In our study the new structure of the W_{new} matrix is accommodated for the special case of strict control horizon $N_u = 3$ and can be shown to be

$$W_{new} = \begin{bmatrix} 0 & -\lambda_{11} & -\lambda_{12} \\ -\lambda_{11} & 0 & -\lambda_{21} \\ -\lambda_{12} & -\lambda_{21} & 0 \end{bmatrix} \tag{3.31}$$

To achieve fast response with minimal oscillations, Δu must be evaluated from a well conditioned system matrix. This matrix has to be weighted in such a manner as to reduce condition number and also increase the determinant. In order to increase $\|\mathbf{G}^T\mathbf{G} + W_{new}\|$ to obtain lower values of the condition number of $(\mathbf{G}^T\mathbf{G} + W_{new})$, it is assumed that the even elements of the first row of the weighting matrix W_{new} are equal. At this stage, the EPC structure of the W_{new} matrix is designed to have three parameter σ_1, σ_2 and λ, for any value of the control horizon $(N_u = 3)$. The W_{new} matrix using these parameters is now termed W_{epc}. The weighting matrix W_{epc} is

$$W_{epc} = \begin{bmatrix} 0 & -\sigma_1\lambda & -\lambda \\ -\sigma_1\lambda & 0 & -\sigma_2\lambda \\ -\lambda & -\sigma_2\lambda & 0 \end{bmatrix} \tag{3.32}$$

The formulation of the new control law is obtained

$$\mathbf{u} = (\mathbf{G}^T\mathbf{G} + W_{epc})^{-1}\mathbf{G}^T(\mathbf{f} - \mathbf{w}) \tag{3.33}$$

where σ_1 and σ_2 are the new weighting factor in the W_{epc} matrix and λ is the old weighting matrix.

With the large number of the prediction horizon N_2, the system matrix $\mathbf{G}^T\mathbf{G}$ can be expressed as a function of $a = \sum_{i=0}^{N_2-1} g_i g_i$, where g_i for $i = 0, 2, \cdots, N_2 - 1$ is the first column of the dynamic matrix $\mathbf{G}^T$. The system matrix can be defined as [15]

$$\mathbf{G}^T\mathbf{G} = \sum_{i=0}^{N_2-1} g_i g_i \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{3.34}$$

For clarity, we define C_{gpc} and C_{epc} are weighted matrices for GPC and EPC respectively, where its expression is demonstrated as

$$C_{gpc} = \begin{bmatrix} a-\lambda & a & a \\ a & a-\lambda & a \\ a & a & a-\lambda \end{bmatrix} \tag{3.35}$$

$$C_{epc} = \begin{bmatrix} a & a-\sigma_1\lambda & a-\lambda \\ a-\sigma_1\lambda & a & a-\sigma_2\lambda \\ a-\lambda & a-\sigma_2\lambda & a \end{bmatrix} \tag{3.36}$$

From Eq. (3.36) and (3.34), it is shown that the condition number of C_{epc} can be further improved by eliminating the second element of the first row and assuming that $\sigma_1\lambda = a$. Therefore, λ is no longer a tuning parameter as in GPC controller structure whereas the new tuning parameters σ_1 and σ_2 will be used instead. Thus the main question to be deal with in this stage is how to tune the new strategy parameters σ_1 and σ_2 in order of the system matrix C_{epc} is weighted with well condition. To deal with this conestone problem the simply tuning rule of σ_1 and σ_2 on EPC will be presented next section.

Therefore, the approximated condition number γ_{epc} for EPC can be represented as a function of the new weighting factor σ_1 and σ_2 as following

$$\gamma_{epc} = 1 + \sqrt{\left(\frac{\sigma_2}{\sigma_1}\right)^2 - 2\left(\frac{\sigma_2}{\sigma_1}\right) + 2 - \left(\frac{2}{\sigma_1} - \frac{1}{\sigma_1^2}\right)} \tag{3.37}$$

The determinant of C_{epc} will be calculated in terms of λ, σ_1, σ_2 as

$$|C_{epc}| = a\left[2\left(\sigma_1 + \sigma_2 + \sigma_1\sigma_2\right) - \left(\sigma_1^2 + \sigma_2^2 + 1\right)\right]\lambda^2 - 2\sigma_1\sigma_2\lambda^3 \tag{3.38}$$

3.2.2 Tuning of σ_1 and σ_2 for EPC

In this section, the guidelines on the choice of σ_1 and σ_2 are presented individually. First by setting $\sigma_2 = 1$ for the value of $N_u = 3$, $|C_{epc}|$ is thus evaluated by using only weighting factor σ_1. Second $|C_{epc}|$ is evaluated by using only weighting factor σ_2. Moreover, the expressions for determinant and condition number for GPC and EPC are given in Table (3.1). It is note that the maximum value of

Table 3.1: Determinant and condition number for $N_u = 3$

Control Scheme	Determinant	Condition Number
GPC	$N_u a\lambda^{N_u-1} + \lambda^{N_u}$	$N_u a/\lambda + 1$
EPC: $\sigma_2 = 1$	$\lambda^2\left[a\sigma_1(4-\sigma_1) - 2\sigma_1\lambda\right]$	$1 + \sqrt{2\left(1/\sigma_1^2 - 2/\sigma_1 + 1\right)}$
EPC: $\sigma_1 = 1$	$\lambda^2\left[a\sigma_2(4-\sigma_2) - 2\sigma_2\lambda\right]$	$1 + \sqrt{\sigma_2^2 - 2\sigma_2 + 3}$

$|C_{epc}|$ occurs for the reformulated structure of EPC at $\lambda = a\left[(4-\sigma_1)/3\right]$ whereas no maximum value of $|C_{gpc}|$ for the old structure of GPC. Since the determinant of the weighted matrix for EPC must be greater than GPC ($|C_{epc}| > |C_{gpc}|$) and the condition number of the weighted matrix for EPC must be smaller than GPC ($\gamma_{epc} < \gamma_{gpc}$), an inequality condition are derived as a function of a and λ

$$\begin{cases} \sigma_1(\lambda, a) > \left(2 - \frac{\lambda}{a}\right) + \frac{1}{a}\sqrt{\lambda^2 - 5a\lambda + a^2} \\ \sigma_2(\lambda, a) > \left(2 - \frac{\lambda}{a}\right) + \frac{1}{a}\sqrt{\lambda^2 - 5a\lambda + a^2} \\ \lambda < \left(5 - \sqrt{21}\right)\frac{a}{2} \\ \sigma_2 > \sigma_1 \end{cases}$$

It can be conclude that the weighting factor σ_1 and σ_2 must be greater than 1 in order to have hight determinant values for any value of $\lambda < a$.

3.2.3 EPC Tuning Strategy for TITO

The mathematical analysis used to derive the weighting matrix for unconstrained TITO EPC closely follows the SISO case presented in [15]. The dynamic matrix $\mathbf{G}$ for the TITO system of 2 controlled variables and 2 manipulated variables is

$$\mathbf{G} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \tag{3.39}$$

where $\mathbf{G}_{ij}$ is the dynamic matrix of the corresponding process.

The system matrix for 2 inputs and 2 outputs cases is given by

$$\mathbf{G}^T\mathbf{G} = \begin{bmatrix} G_{11}^T G_{11} + G_{21}^T G_{21} & G_{11}^T G_{12} + G_{21}^T G_{22} \\ G_{12}^T G_{11} + G_{22}^T G_{21} & G_{12}^T G_{12} + G_{22}^T G_{22} \end{bmatrix} \tag{3.40}$$

The complicity of the system matrix is reduced by separating as a sum of two matrices

$$Q_1 = \begin{bmatrix} G_{11}^T G_{11} + G_{21}^T G_{21} & 0 \\ 0 & G_{12}^T G_{12} + G_{22}^T G_{22} \end{bmatrix} \tag{3.41}$$

$$Q_2 = \begin{bmatrix} 0 & G_{11}^T G_{12} + G_{21}^T G_{22} \\ G_{12}^T G_{11} + G_{22}^T G_{21} & 0 \end{bmatrix} \tag{3.42}$$

As can be seem from (3.40), the lowering of all eigenvalues of the system matrix is obtained by adding the weighting matrix W_{BIBO} to the diagonal blocks matrix Q_1 which are equivalent to the corresponding column system matrix of the subprocesses. For this reason, the weighting matrix for BIBO EPC with 2 outputs is given as

$$W_{BIBO} = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \tag{3.43}$$

Equivalently,

$$W_{BIBO} = \begin{bmatrix} 0 & -\sigma_1\lambda_1 & -\lambda_1 & & & \\ -\sigma_1\lambda_1 & 0 & -\sigma_{2,1}\lambda_1 & & 0 & \\ -\lambda_1 & -\sigma_{2,1}\lambda_1 & 0 & & & \\ & & & 0 & -\sigma_1\lambda_2 & -\lambda_2 \\ & 0 & & -\sigma_1\lambda_2 & 0 & -\sigma_{2,2}\lambda_2 \\ & & & -\lambda_2 & -\sigma_{2,2}\lambda_2 & 0 \end{bmatrix} \tag{3.44}$$

where W_j is the extended move suppression matrix of the jth manipulated variable of the BIBO system. The weighting matrix W_{BIBO} which effectively modifies the condition number of the system matrix, is tuned such that the condition number is always bounded by a fixed low value.

Therefore, the overal diagonal blocks of system matrix is expressed as

$$Q_1 + W_{BIBO} = \begin{bmatrix} G_{11}^T G_{11} + G_{21}^T G_{21} + W_1 & 0 \\ 0 & G_{12}^T G_{12} + G_{22}^T G_{22} + W_2 \end{bmatrix} \tag{3.45}$$

Shridhar and Cooper [15] shown that the maximum eigenvalues of all off-diagonal terms are zeros. Therefore, to find an upper bound on the maximum eigenvalue of the overall system matrix, $Q_1 + Q_2$, it can be shown that

$$\mu_{max}(Q_1 + Q_2) \leq \mu_{max}(Q_2) \tag{3.46}$$

where μ_{max} represented the largest eigenvalue of the square matrix.

Therefore, the maximum eigenvalue comes from the diagonal blocks in portioned matrix (Q_2). Mathematically [15] $\sigma_{2,2}$ has more impact on the determinant and condition number of the overall control matrix $(Q_1 + Q_2)$ than $\sigma_{2,1}$

In summary, the tuning procedure is as follows:

- Using 2^{th} diagonal matrix $(Q_1 + W_{BIBO})$ and the inequality condition, obtain the parameters σ_1 and $\sigma_{2,2}$.
- Substituting the selected parameters above to (3.45), then repeat the above for the remaining 1^{st} diagonal matrix of (3.45). Each successive tuning of $\sigma_{2,2}$ and $\sigma_{2,1}$ is based on the corresponding diagonal matrix of $(Q_1 + W_{BIBO})$
- Using these σ values, construct $A_{exact} = (\mathbf{G}^T\mathbf{G} + W_{BIBO})$ and use in the evaluation of $\Delta u(k)$ in (3.33)

3.3 Summary and Discussion

3.3.1 Summary

The generalized predictive controller has been presented as shown in Algorithm 3.2. This algorithm has been developed as extensions of the weighting matrix of the system matrix that according to the Algorithm 3.3. The main idea which we have been illustrated in this chapter are the tuning strategy for both GPC and EPC.

3.3.2 Discussion

In this chapter, we have presented a simple improved approach of EPC for predictive controller. The performance is defined in term of reformulation the structure of the system matrix whereas the control horizon is strict. EPC introduces the idea of well-conditionality which depends on not only condition number, but also the determinant of the matrix. The technique can be efficiently implement since it removes the ill-conditioning in the system matrix by developing the new weighting structure. In next chapter, numerical example based on the two-tank liquid level system will confirm that the new extended predictive control EPC algorithm yields high performance results than the algorithm with conventional generalized predictive control GPC.

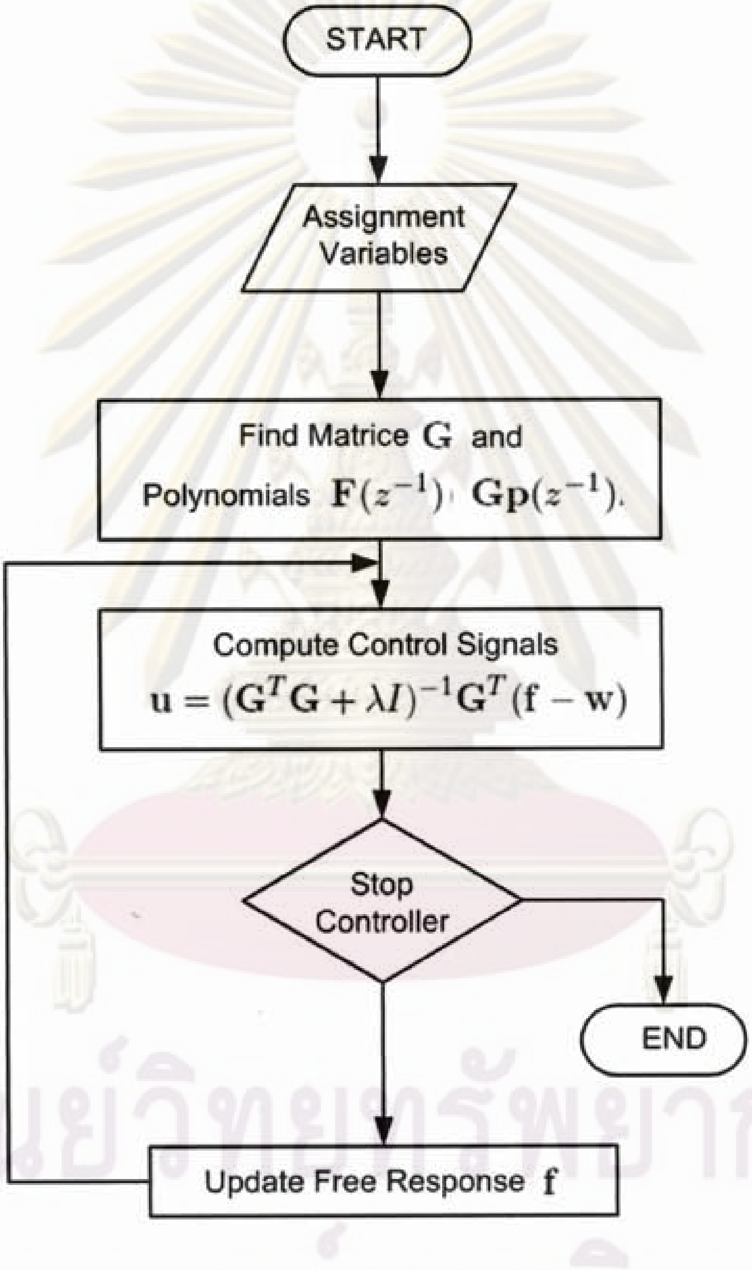

Figure 3.2: GPC Algorithm

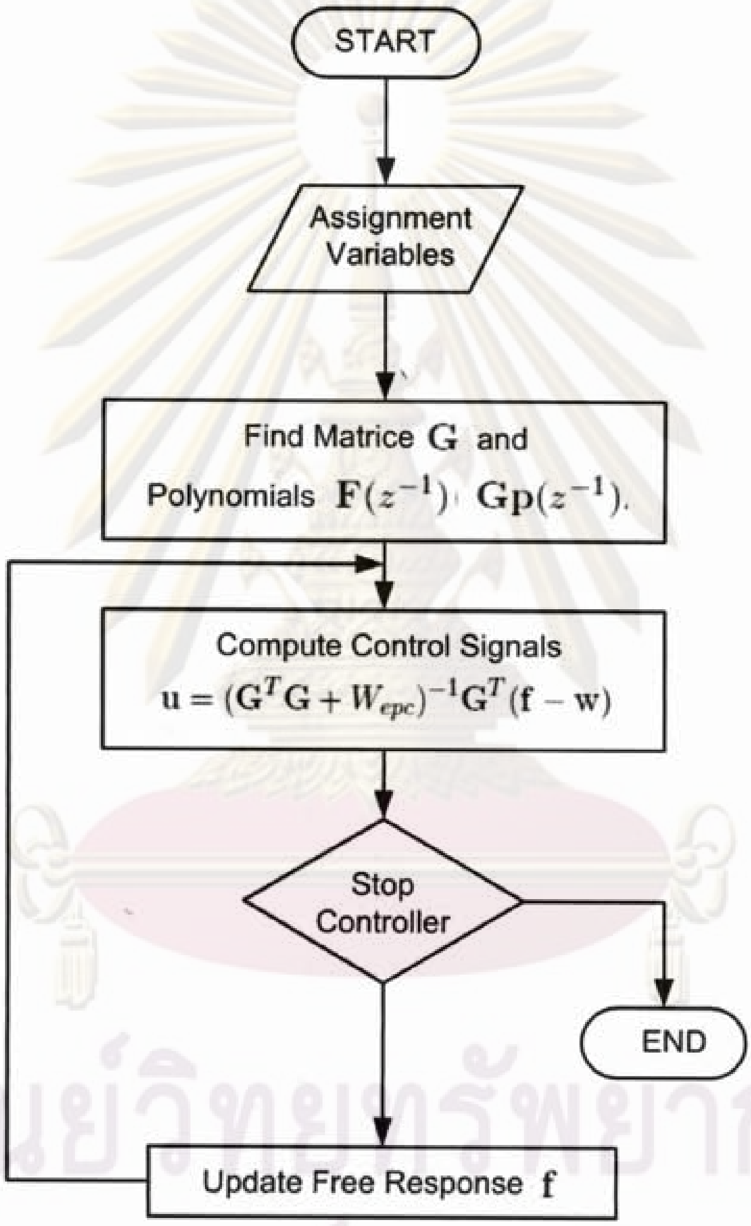

Figure 3.3: EPC Algorithm

CHAPTER IV

IMPLEMENTATION RESULTS

This chapter will show both the simulation and experiment result for water level control process. In Section 4.2 the simulation result using MATLAB will be given to illustrate how is the difference between the control performance of EPC and GPC. Then there will be a discussion on issues relating to implementation of level control via DCS in Section 4.3. This is following by a discussion of the tuning parameters. Finally the chapter is concluded by a discussion on the control performance.

4.1 Introduction

In this chapter the theoretical concept presented in previously are simulated. An experimental validation is also conducted of real liquid level control via CENTUM CS3000 R3, an acronym for Distributed Control System. A model chosen corresponded to a middle level control of YOKOGAWA FIELDBUS TRAINER plant is elaborated and utilized for simulation and implementation. The model was identified by Permpornsri [8] and has been used to try difference control strategies, as detail in Table (4.1).

As shown in Fig. (2.2), the process has two variables LV1 and LV2 that have to be controlled. The controlled variables LV1 and LV2 are the water level of each tank whose specifications are fixed by economic and operational goals and must be kept within a 2 percent of their set point at steady state. Furthermore, it must be controlled within limits fixed by operation constraint 45 cm to 65 cm. The manipulated variables are control valve CV1 and CV2 which are also constraint by 0 to 100. The feed water is pumped all liquid requirement from the reservoir unit for each tank.

Table 4.1: Validation model at the middle level of the tank

Model Parameter	$i=1,\quad j=1$	$i=1,\quad j=2$	$i=2,\quad j=1$	$i=2,\quad j=2$
Static gain: K	3.2	4.15	3.1	4.9
Time constant: τ	452	416	472	443
Time delay: θ	7	32	32	4

4.2 Simulation Results

In order to show the way GPC and EPC can be simulated, simple identified model is presented, where the tuning parameters are shown in the Table (4.2). The controller will be designed for first order system for the sake of clarity. The following transfer matrix is given

Table 4.2: Tuning parameter for GPC and EPC

Control Scheme	Parameters	
GPC	$\gamma_1 = 1$	$\gamma_2 = 1$
	$\lambda_1 = 12$	$\lambda_2 = 12$
EPC	$\sigma_1 = 6.27$	
	$\sigma_{2,1} = 18.80$	$\sigma_{2,2} = 31.35$
$N_u = 3$	$N_2 = 20$	$T_s = 14$

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{3.2\,e^{-7s}}{452\,s+1} & \dfrac{4.15\,e^{-32\,s}}{416\,s+1} \\ \dfrac{3.1\,e^{-32\,s}}{472\,s+1} & \dfrac{4.9\,e^{-4\,s}}{443\,s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \tag{4.1}$$

where the manipulated variables $u_1(s)$ and $u_2(s)$ are the control valve CV1 and CV2 respectively. The control variables $y_1(s)$ and $y_2(s)$ are water level at tank 1 and 2 respectively.

The discretized model for a sampling time of 14 seconds is

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \dfrac{0.0492 + 0.0484\,z^{-1}}{1 - 0.9695\,z^{-1}} & \dfrac{0.0986 + 0.0388\,z^{-1}}{1 - 0.9669\,z^{-1}} \\ \dfrac{0.0650 + 0.0256\,z^{-1}}{1 - 0.9708\,z^{-1}} & \dfrac{0.1094 + 0.0431\,z^{-1}}{1 - 0.9689\,z^{-1}} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{4.2}$$

A left matrix fraction description can be obtained by making matrix $A(z^{-1})$ equal to a diagonal matrix with diagonal elements equal to the least common multiple of the denominators of the corresponding row of the transfer function, resulting in

$$A(z^{-1}) = \begin{bmatrix} 1 - 1.9364z^{-1} + 0.9374z^{-2} & 0 \\ 0 & 1 - 1.9397z^{-1} + 0.9406z^{-2} \end{bmatrix} \tag{4.3}$$

$$B(z^{-1}) = \begin{bmatrix} 0.0492 + 0.0009z^{-1} - 0.0468z^{-2} & 0.0986 - 0.0568z^{-1} - 0.0376z^{-2} \\ 0.0650 - 0.0374z^{-1} - 0.0248z^{-2} & 0.1094 - 0.0631z^{-1} - 0.0418z^{-2} \end{bmatrix} \tag{4.4}$$

Since the plant is a 2×2 system and control horizon $N_u = 3$, the weighting matrix is now

$$W_1 = \begin{bmatrix} 0 & -75.24 & -12 \\ -75.24 & 0 & -225.72 \\ -12 & -225.72 & 0 \end{bmatrix} \tag{4.5}$$

$$W_2 = \begin{bmatrix} 0 & -75.24 & -12 \\ -75.24 & 0 & -376.20 \\ -12 & -376.20 & 0 \end{bmatrix} \tag{4.6}$$

The computer simulation of tuning strategy for BIBO level process, the weighting ratio $\sigma_{2,2}$ is tuned first then the parameter $\sigma_{2,1}$ is chosen based on the value of $\sigma_{2,2}$. The system matrix of the process with $N_u = 3$ and $\sigma_1 = 6.27$ is given on the Table (4.2). To tune $\sigma_{2,2}$ and $\sigma_{2,1}$, the submatrices A_1 and A_2 are used independently. Using the submatrix A_2 obtain $\sigma_{2,2}$ and the subsequent using the A_1 matrix, obtain $\sigma_{2,1}$. Note that in order to obtain $\sigma_{2,1}$, the tuned A_2 matrix is used first.

With the matrix A_1 and A_2 are formed as following

$$A_2 = G_{12}^T G_{12} + G_{22}^T G_{22} + W_2 \tag{4.7}$$

$$A_1 = \begin{bmatrix} G_{11}^T G_{11} + G_{21}^T G_{21} + W_1 & 0 \\ 0 & A_2 \end{bmatrix} \tag{4.8}$$

In numerical, the submatrix A_2 and A_1 are given as

$$A_2 = \begin{bmatrix} 75.24 & -4.84 & 53.44 \\ -4.84 & 66.03 & -314.68 \\ 53.44 & -314.68 & 57.48 \end{bmatrix} \tag{4.9}$$

$$A_1 = \left[\begin{array}{cccccc} 32.08 & -45.25 & 15.84 & & & \\ -45.25 & 28.10 & -199.57 & & \mathbf{0} & \\ 15.84 & -199.57 & 24.40 & & & \\ & & & 75.24 & -4.84 & 53.44 \\ & \mathbf{0} & & -4.84 & 66.03 & -314.68 \\ & & & 53.44 & -314.68 & 57.48 \end{array}\right] \tag{4.10}$$

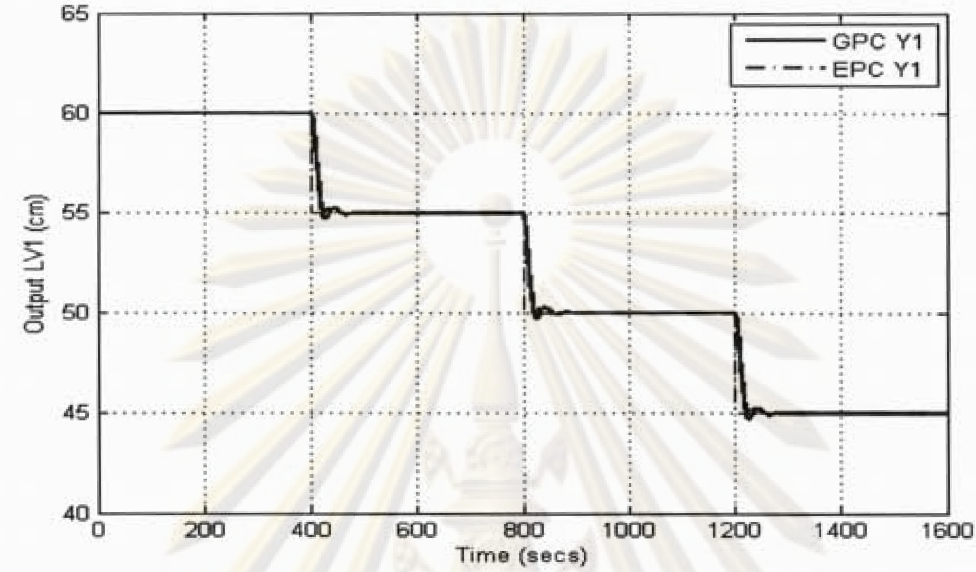

Figure 4.1: Performance comparison of LV1 for process tank using EPC and GPC controllers

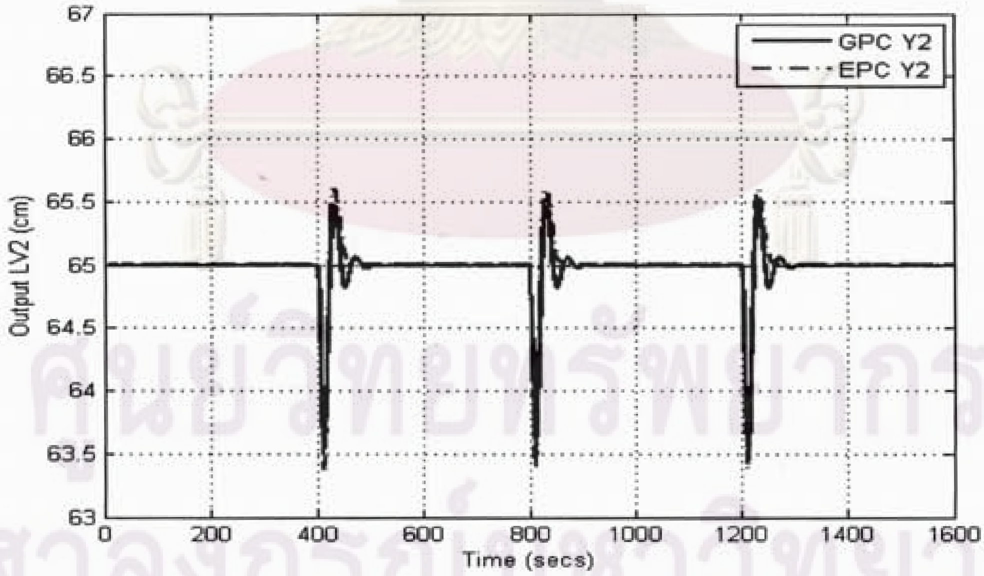

Figure 4.2: Response of LV2 for process tank using EPC and GPC for disturbance rejection capabilities at a set point 65 cm

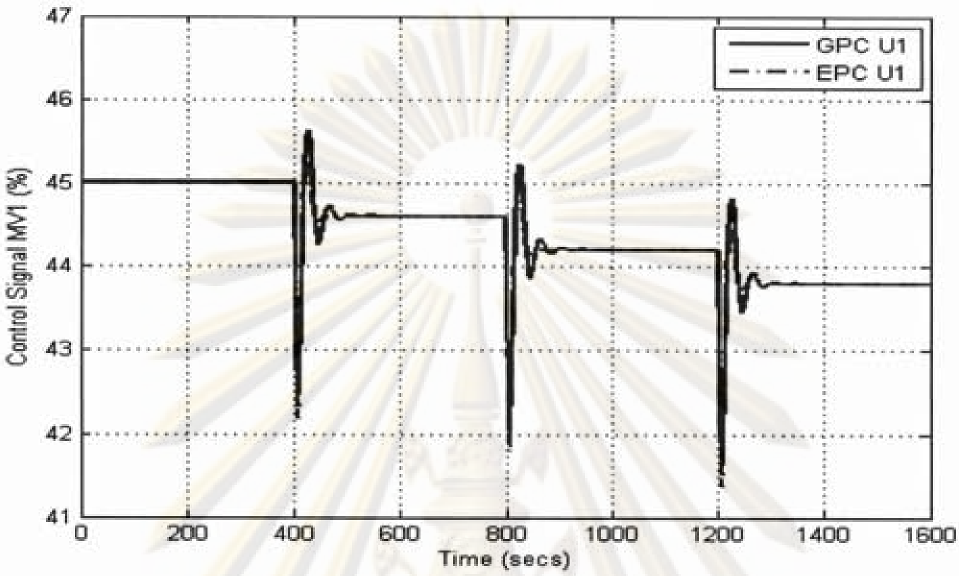

Figure 4.3: Manipulated Variable 1 (MV1) comparison of EPC and GPC controllers for TTLP

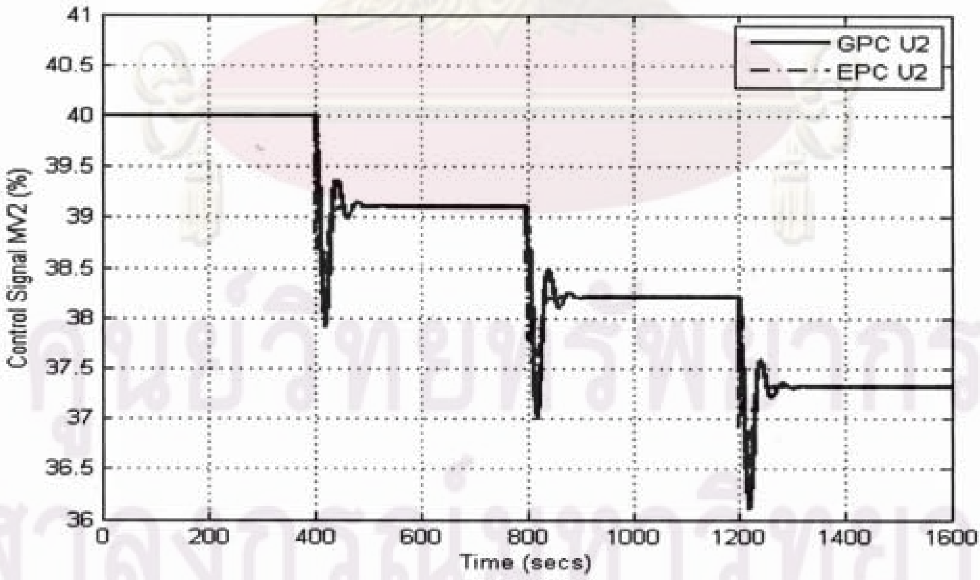

Figure 4.4: Manipulated Variable 2 (MV2) comparison of EPC and GPC controllers for TTLP

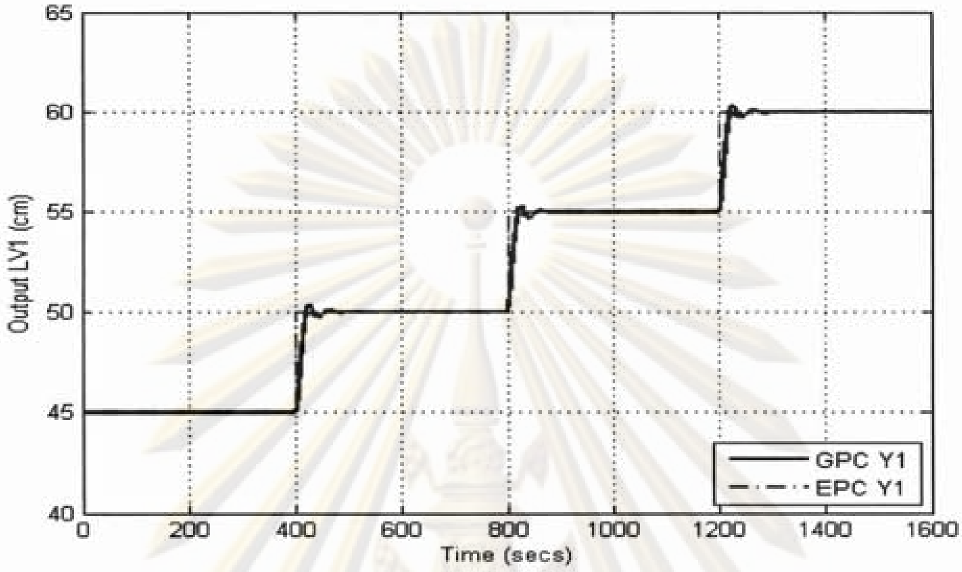

Figure 4.5: Response of LV1 for process tank using EPC and GPC controllers

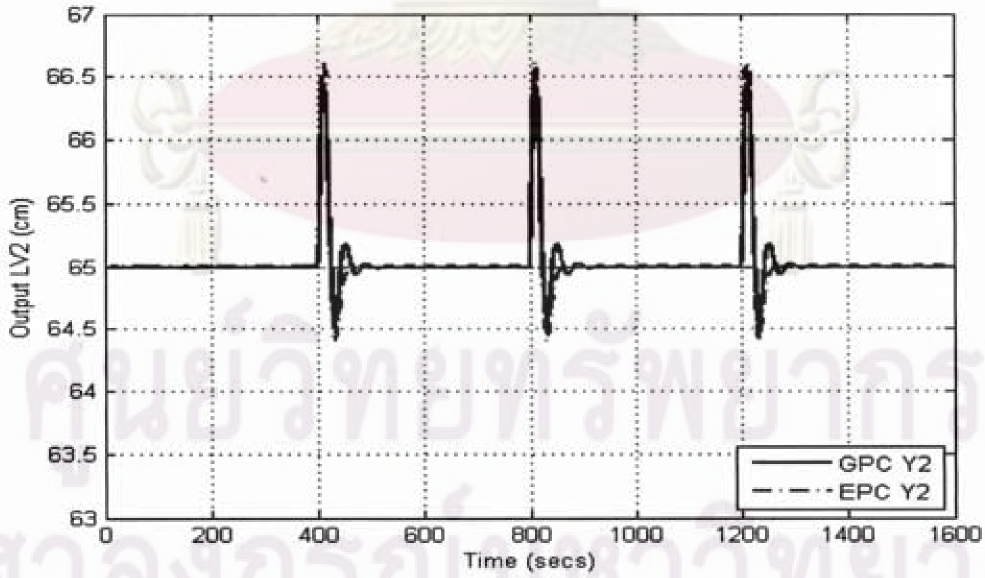

Figure 4.6: Response of LV2 for process tank using EPC and GPC for disturbance rejection capabilities at a set point 65 cm

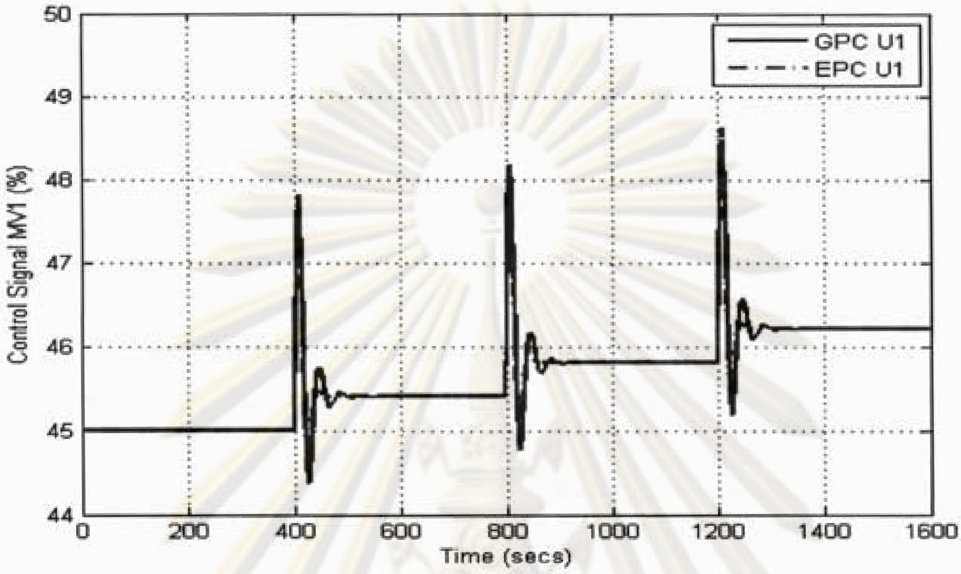

Figure 4.7: MV1 comparison of EPC and GPC controllers

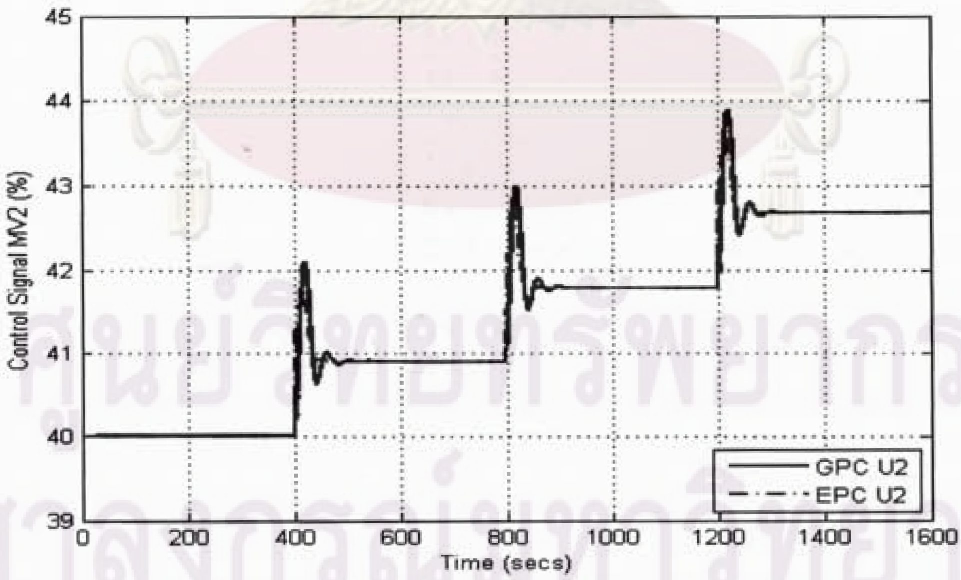

Figure 4.8: MV2 comparison of EPC and GPC controllers

In order to illustrate the performance of controlling TITO process tank with difference operation points for the output and the manipulated variables GPC and EPC controller are applied for TTLP and the simulation results are obtained as shown from Fig. 4.1 to Fig:4.8.

Fig. 4.1, 4.3, 4.5 and 4.6 show the behavior of the water level $LV1$ and $LV2$ in the tank in the present of change in the level reference 5 cm at each 400 sec for tank 1 and keep constant 65 cm for tank 2. As can be observed, the outputs follow the reference by mean of the two manipulated variables $MV1$ and $MV2$. It can also be seen that any change affects all the variables, such as $LV1$ and $LV2$, which are slightly move from the reference during the transient stage.

The simulation results of response and manipulated variables for GPC and EPC shown the closed-loop results due to the set point changes. The weighting matrix W_{epc} in Eq. (3.32) for $N_3 = 3$ translate the simple and general form, where the form of this matrix is applicable for any process industrial by using only the specification of σ variable. The effect on closed-loop response is clearly seen as σ varies. By considering performance such as percent overshoot and settling time, the EPC controller has ability to handle both wideranging set point change with small oscillation and responding quickly in order to keep good tracking of the set point changes in comparing with GPC as shown on the Table 4.5. A noticeable feature of EPC from the Table 4.5 is that it has a relatively low condition number and the highest determinant of the system matrix in comparison to GPC controller.

Table 4.3: A comparison of GPC and EPC controller for TTLP

Control Scheme	γ_{exact}	$\|A_{exact}\|$	Operation (cm)	Setting Time (sec)	Overshoot (%)
GPC	24.54	7.88×10^7	LV1: 60 − 55	52	6
			LV1: 55 − 50	52	6
			LV1: 50 − 45	52	6
			LV1: 45 − 50	52	6
			LV1: 50 − 55	52	6
			LV1: 55 − 60	52	6
EPC	57.10	3.37×10^{12}	LV1: 60 − 55	45	0.5
			LV1: 65 − 65	45	0.5
			LV1: 45 − 50	45	0.5
			LV1: 65 − 65	45	0.5
			LV1: 50 − 55	45	0.5
			LV1: 55 − 60	45	0.5

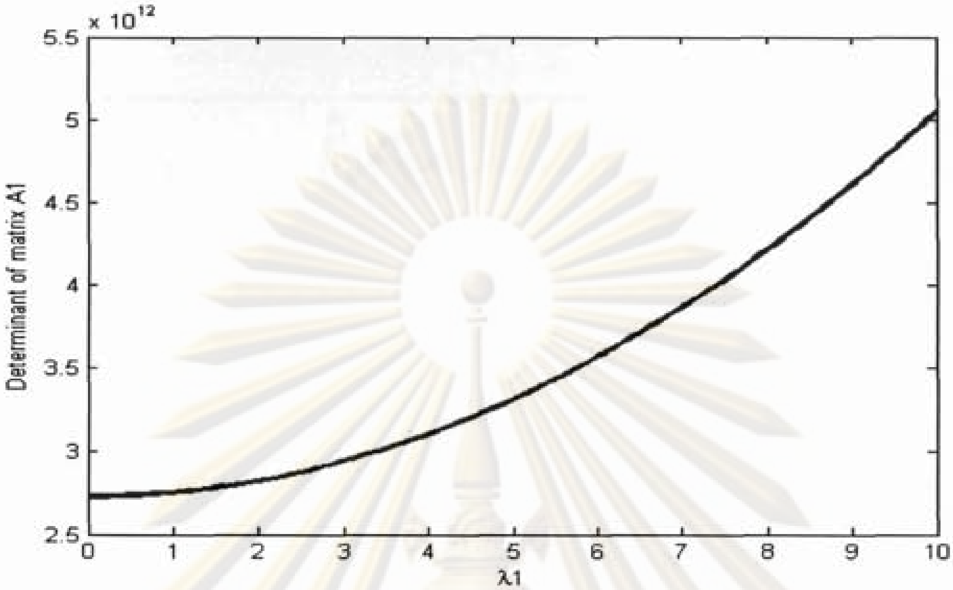

Figure 4.9: The effect of the weighting factor λ_1 on the determinant of matrix A_1 ($N_u = 3$)

Since λ_1 and λ_2 must be positive [3, 4], in order to have a weighting influence on the determinant of the weighted matrix C_{gpc} and C_{epc} the effect of λ_1 on $|C_{gpc}|$ and $|C_{epc}|$ are presented. From the Fig. 4.9 shows the effect of variable λ_1 on $|C_{gpc}|$, where the value of $\lambda_2 = 12$ is selected [3, 4].

To have the larger determinant of C_{gpc} for any value of $\lambda < \left(5 - \sqrt{21}\right)\frac{a}{2}$. It must be increasing. From this increasing value of λ_1 the poor performance in controlling for GPC controller is obtained as shown in the Fig. 4.10 and 4.11. It can be conclude that with the same weighting factors $\lambda_1 = \lambda_2 = 12$ the conditionality of C_{epc} is improved such $|C_{epc}| = 0.07 \times 10^9$, the value of λ for GPC must be large to increase its determinant as given on the Table 4.4.

Table 4.4: Tuning parameter for GPC and EPC

Control Scheme	Weighting Factor	γ_{exact}	$\lvert A_{exact}\rvert$	Setting Time (sec)	Overshoot (%)
GPC	$\lambda_1 = \lambda_2 = 12$	24.54	0.07×10^9	52	6
	$\lambda_1 = \lambda_2 = 15$	19.83	0.23×10^9	79	11
	$\lambda_1 = \lambda_2 = 20$	15.12	0.23×10^9	215	18

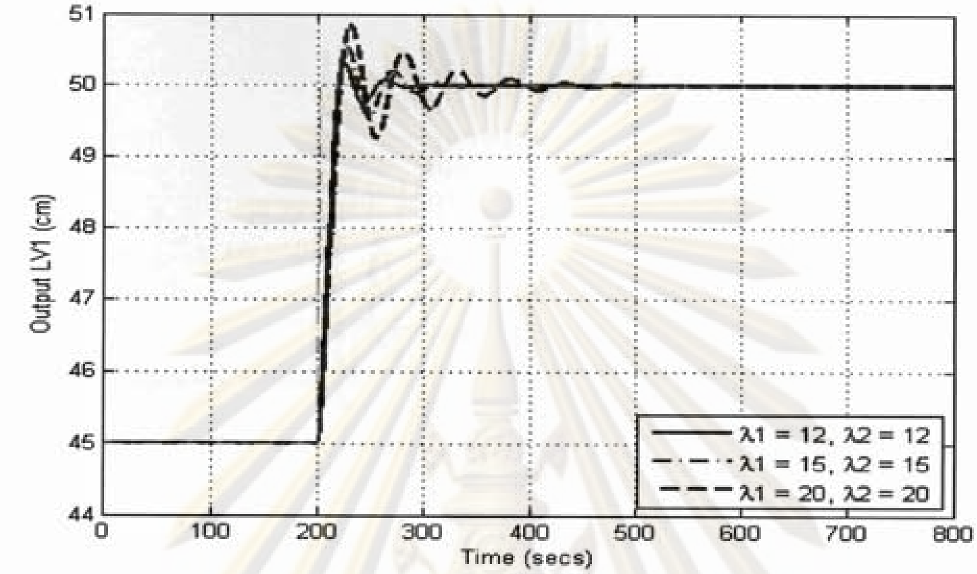

Figure 4.10: Output LV1 Comparison of GPC controller for difference λ

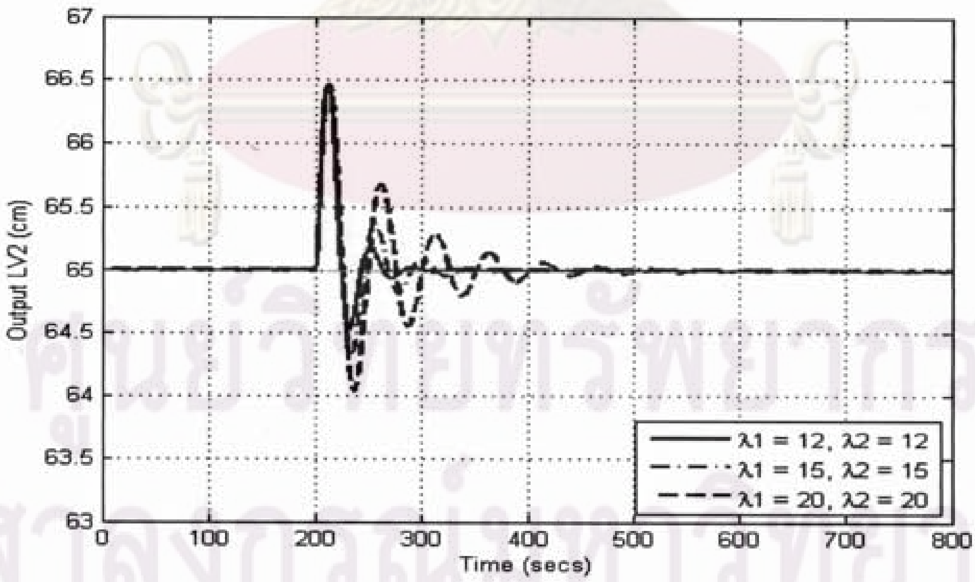

Figure 4.11: Response of LV2 using GPC controller for difference λ

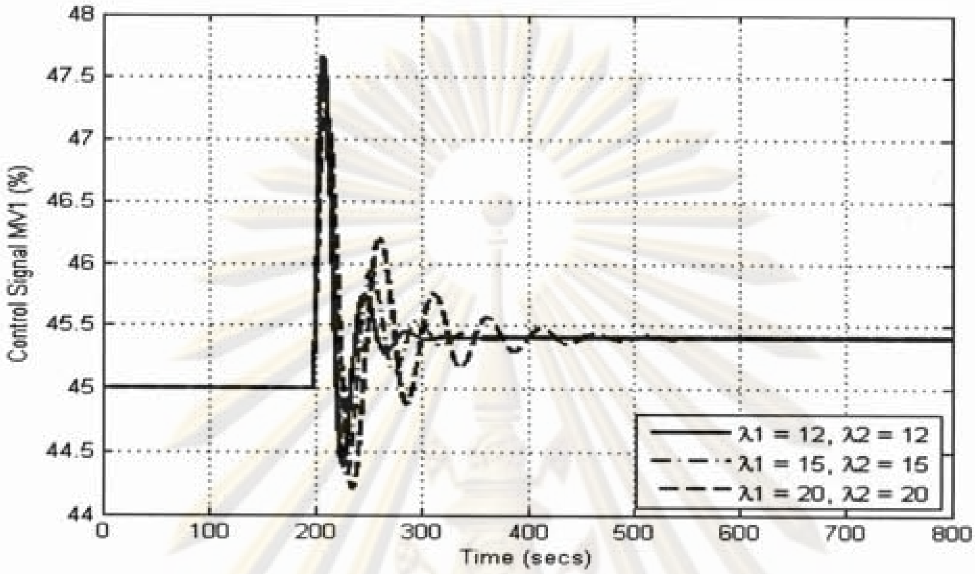

Figure 4.12: Manipulated variable comparison of GPC controller for difference λ

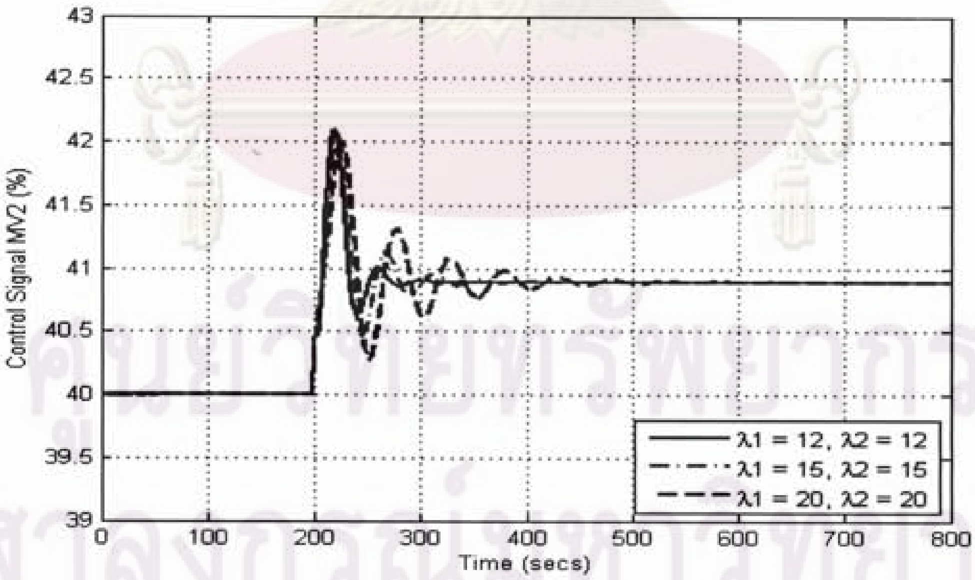

Figure 4.13: Manipulated variable comparison of GPC controller for difference λ

4.3 Experiment Results

In this section, experimental test are carried out on the process tank existing on Yokogawa Fieldbus Trainer (YFT) plant via DCS. The connection between plant and CENTUM CS3000 R3 is given in Fig. 4.14. The predictive control GPC and EPC tuning scheme are applied to the real liquid process, where its model is discretized by Tustin Approximation. This method has the advantage of being a stable continuous time system that is transformed into a stable sampled system. For CENTUM CS3000 R3 the controller is represented as the function block in which SEquence and Batch Oriented Language (SEBOL) programming language is used for the level process. In addition, SEBOL is a programming language that is suitable for process control. It has special features for the functions of a generic programming language. Programs written in SEBOL are executed as an action equivalent to one SFC step on the FCS.

The objective of the implementation is to maintain the water level in the tank 1 at 65 cm inspire of changing operating level in the tank 2. First the water in these two tanks are fixed at 55 cm, the control valve CV1 and CV2 open at 40 and 65 percent respectively. The level of the tank 1 will be changed from 55 cm to 45 cm while the level in the tank 2 change from 55 cm to 65 cm. For the second experiment is just keep the level of tank 1 be 55 cm.

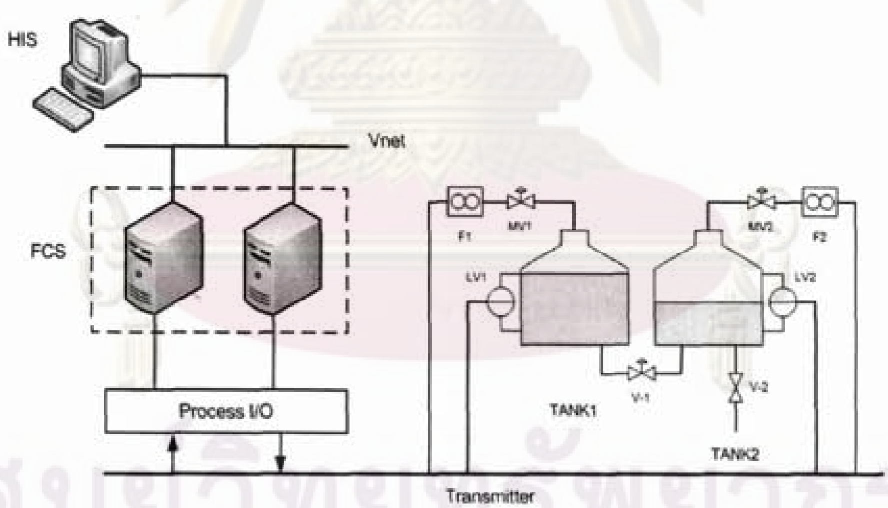

Figure 4.14: The connection of DCS with level process on YFT

In the case of practical implementation, the performance of EPC method was compared with GPC. Using GPC and EPC tuning procedure, closed-loop test was conducted on TTLP plant with the dynamic matrix **G** in Eq. (3.39) was obtained from the open-loop test of the two-tank process. From Fig. 4.15 to Fig. 4.22 shown the results of EPC at the specified $\sigma_1 = 6.27$, $\sigma_{2,1} = 18.8$, and

$\sigma_{2,2} = 31.35$ and GPC having tuning parameters and $\lambda_1 = \lambda_2 = 12$, both using $N_u = 3$

Fig. 4.16 and Fig. 4.21 are the experiment results of response variables $LV1$ and $LV2$ for the two difference cases. It is clear that using the EPC method drastically gives faster closed-loop responses and shorter settling times in comparing with GPC. An interesting observation is that even though γ_{exact} for GPC is lower than γ_{exact} for EPC for the control horizon $N_u = 3$, the closed-loop output level using EPC is better than GPC. This can be explained by the fact that the $|C_{exact}|$ for GPC as compared to EPC is smaller in value which makes the manipulated variable changes more aggressive resulting in higher oscillation. In general, EPC minimizes the errors rapidly between the reference trajectories and the predicted outputs much better than GPC.

Table 4.5: A comparison of GPC and EPC controller for TTLP

Control Scheme	γ_{exact}	$\lvert A_{exact}\rvert$	Operation (cm)	Setting Time (sec)	Overshoot (%)
GPC	24.54	7.88×10^7	LV1: 55 − 45	1440	−
			LV2: 55 − 65	1632	−
			LV1: 55 − 55	1450	−
			LV2: 55 − 65	1348	−
EPC	57.10	3.37×10^{12}	LV1: 55 − 45	740	15.62
			LV2: 55 − 65	745	15.70
			LV1: 55 − 55	1123	−
			LV2: 55 − 65	1200	−

4.4 Discussions

The objective of this section is to illustrate the difference results between simulation and implementation of GPC and EPC method for TTLP. The simulation results show that the methods are applicable for level process and the output responses LV1 and LV2 performance for single tuning parameter EPC compared with GPC is improved results. However, implementation results showed not much improvement of the proposed methods since the system behavior has effected by the uncertain parameter that can be encountered as below

- In fact, DCS is composed of central controller and the remote system containing a physical plant, sensors and actuators that always have the uncertain parameter or physical error. In addition, the controller and the plant are located at difference spatial locations and connected through network to form closed-loop control. Therefore, the identified two-tank model may not the best match to the real process.
- It is not easy to select the best initial parameter for both controller which must to follow the Algorithm 3.2 and 3.3

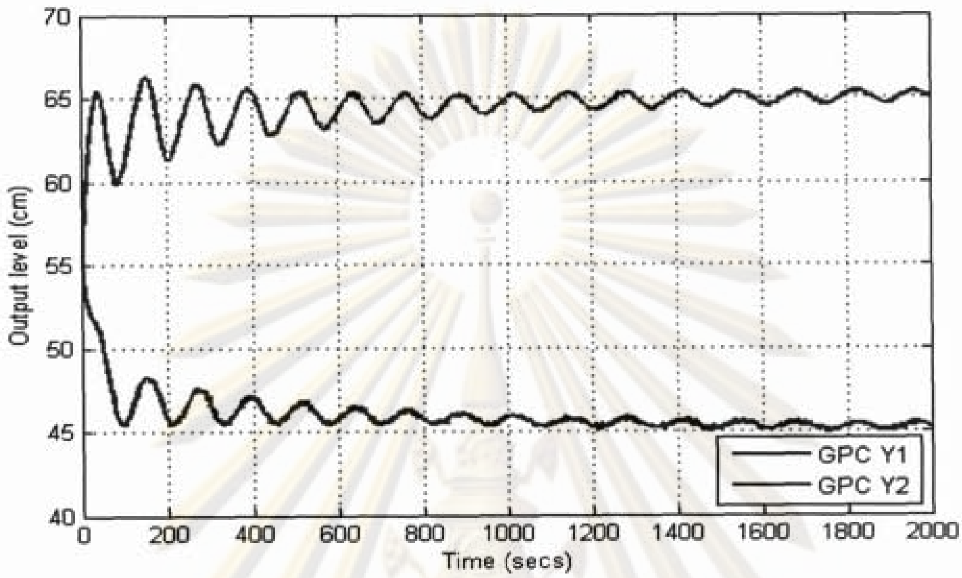

Figure 4.15: Response of the outputs LV1 and LV2 using GPC controller

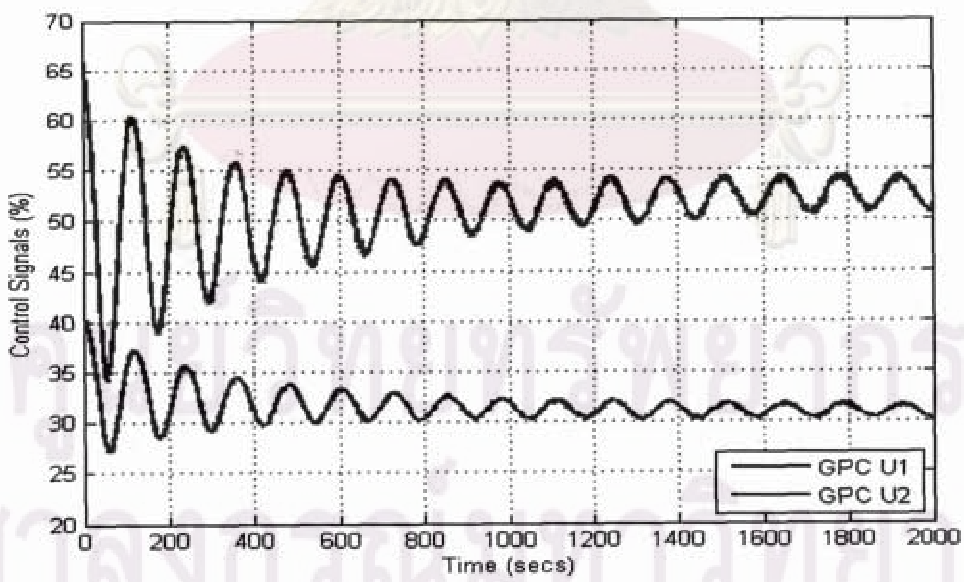

Figure 4.16: Manipulated Variables of GPC controller

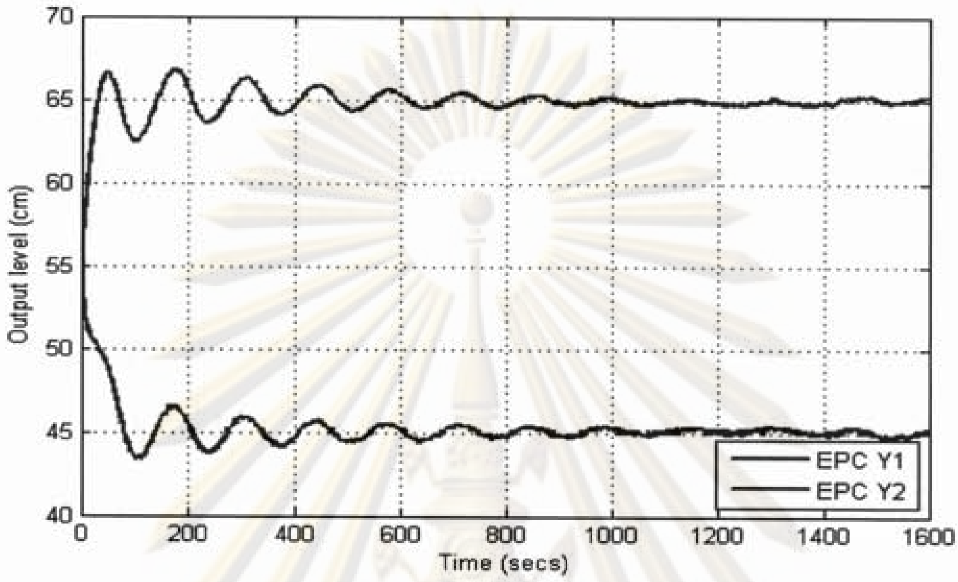

Figure 4.17: Response of the outputs LV1 and LV2 using EPC controller

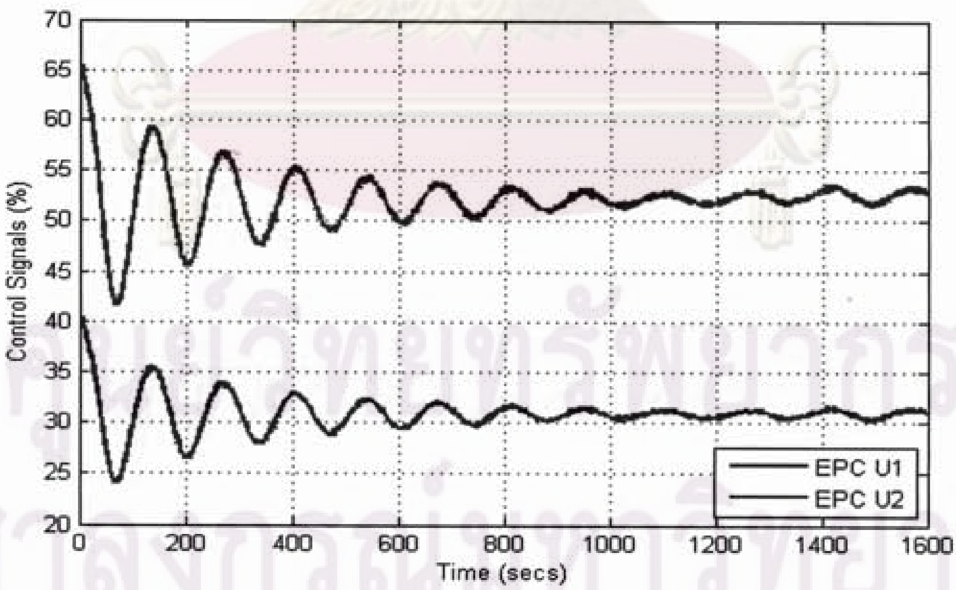

Figure 4.18: Manipulated Variables of EPC controller

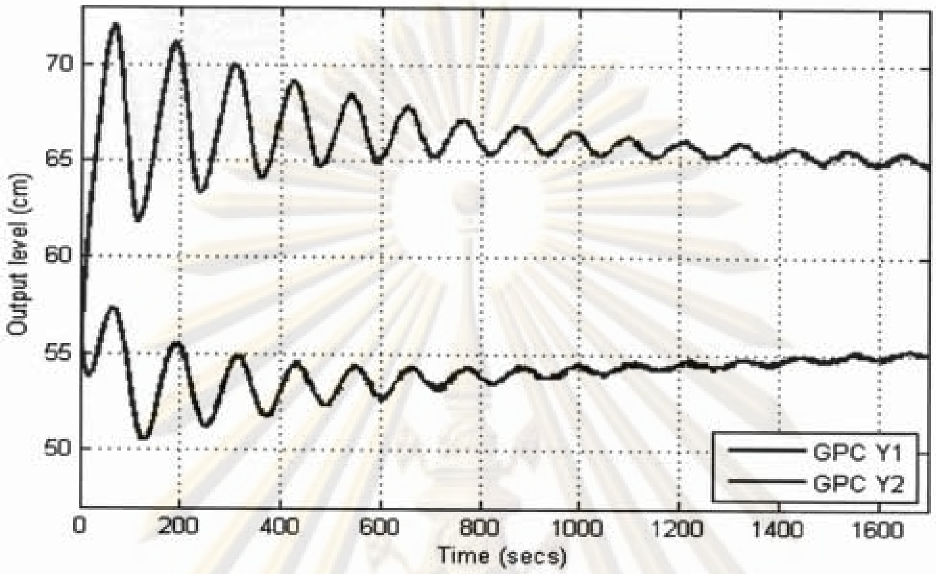

Figure 4.19: The output responses of TTLP using GPC controller

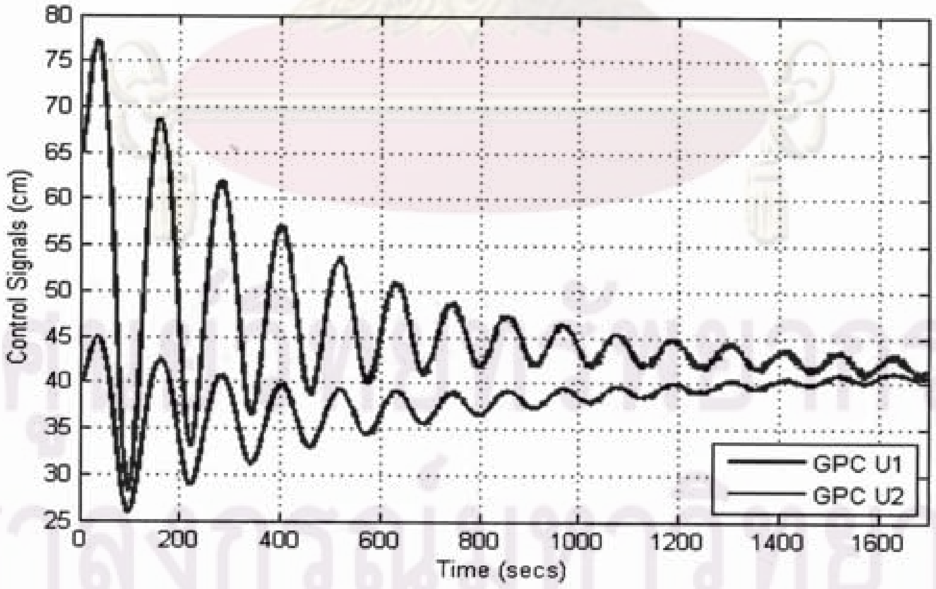

Figure 4.20: Manipulated Variables of GPC controller

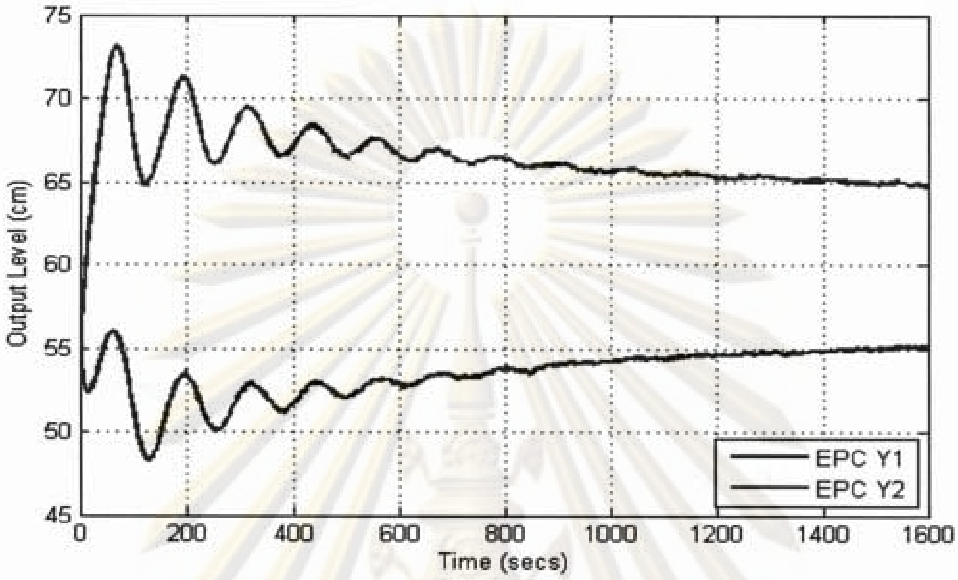

Figure 4.21: The output responses of TTLP using EPC controller

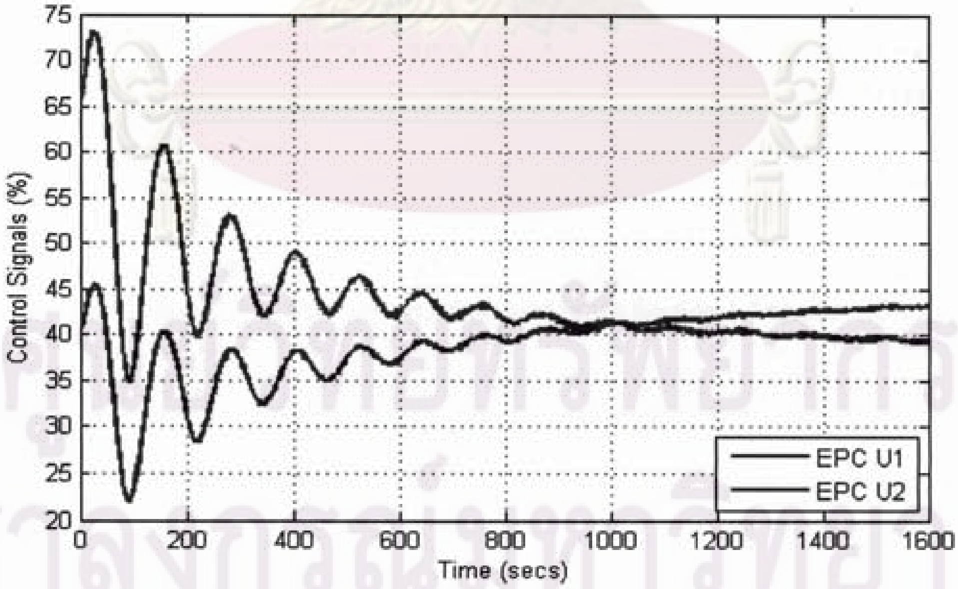

Figure 4.22: Manipulated Variables of EPC controller

CHAPTER V

CONCLUSIONS

5.1 Summary of Results

This thesis investigates two main research orientations that are simulation of the two-tank water level process and implementation of this plant via distributed control system. The two predictive controllers which are generalized predictive controller and extended predictive controller have been studied. Based on the previous work done by Clarke and Mohtadi [5], the GPC tuning strategy for the principle components that are minimum and maximum prediction horizon, control horizon, and weighting factor was given. However, the system matrix conditionality of the GPC control law based on quadratic function is not guaranteed well-conditioning, the EPC algorithm presented in this thesis employs the condition number of the system matrix to provide tuning procedure and improve the system matrix conditionality by increasing its determinant for strict control horizon. Simulation and implementation to two-tank process via DCS shows the improved control performance of ECP compared with GPC.

The specific conclusions associated with the difference chapters of this work are reported hereafter.

Chapter 2, the dynamic model of TTLP has been described. After introducing the essential features and model of TTLP, we provided brief introduction of distributed control system (DCS) comprised of the three main components, human interface station (HIS), field control station (FCS), and process input/output. It is important for control practitioner to understand how to implement the real system.

Chapter 3 covers the theoretical and computational basic for providing technique in controller design. As described in Section 3.1, the GPC control law can be obtained by solving the quadratic cost function over prediction horizon and control horizon. In addition, the choice of three specific tuning parameters was proposed. This approach was then used in the Section 3.2 to assess the influence of the design variables on the stability of the closed-loop control system. Moreover, the system matrix of GPC control law can be reformulated with the new structure by considering the well-conditioning matrix, as presented in the Section 3.2.

Finally, not only the simulation results, but also application results via DCS at the middle operating point were shown in Chapter **??**. In order to illustrated the performance of each controller, the process tank modeled based on step response analysis is used for tuning all control parameters. In comparison to GPC, it can be seen from the results that EPC controller is better than GPC controller with provided faster closed-loop responses with minimum overshoot. It is to be noted that both GPC and EPC is constrained to a control horizon N_u of 3.

5.2 Recommendation

The implementation in this thesis is done via distributed control system (YOKOGAWA CENTUM CS R3). To make TTLP has more general in practice, we need to improve some possible extensions that can be encountered as below

- In this thesis, the control horizon is strict for $N_u = 3$ to implement both GPC and EPC controller for control the water level. With this number we can control the plant. For more general, it is interesting to apply for any large number of N_u and derive a general expression of condition number and determinant in order to provide an effective tuning rule.
- Since the result shows that EPC controller has the performance better than GPC controller. It is strongly recommended for TTLP system as the process could be operated for not only middle tank level but also other set points.

REFERENCES

[1] K. Wanzeng, Z. Shan, and Y. Can. GPC Algorithm and Queuing-Selecting for Networked Level Control. *IEEE Int. Conf. on Control and Automation* 6, 4, (2007): 2626–2629.

[2] S. Permpornsri. Model Predictive Control with Multiple Model Adaptive Controller for Liquid Level via Distributed Control System. Master's thesis, Chulalongkorn University, Bangkok, 2007.

[3] C. E. Carcia and A. M. Morshedi. Quadratic Programming Solution of Dynamic Matrix Control (QDMC). *Chem. Eng. Commun* 73, 46, (1986): .

[4] D. M. Prett and M. Morari. *Shell Process Control Workstation.* London: Butterworths, 1987.

[5] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized Predictive Control-Part I. the Basic Algorithm, part II. Extension and Interpretation. *Automatica* 23, (1987): 137-160.

[6] J. S. Qin and T. A. Badgwell. An Overview of Industrial Model Predictive Control Technology. in *Process Control Conference., CPC V*, (Tahoe, CA).

[7] M. G. Grimble and W. O. Andrzej. Predictive Control for Industrial Applications. *Annual Reviews in Control* 25, (2001): 13-24.

[8] D. W. Clarke and P. J. Gawthrop. Self-Tuning Control. *IEE* 123, (1979): 633-640.

[9] D. W. Clarke and C. Mohtadi. Properties of Generalized Predictive Control. *Automatica* 25, (1989): 859-875.

[10] R. Shridhar and D. J. Cooper. A Tuning Strategy for Unconstrained SISO Model Predictive Control. *Ind. Eng. Chem.* 36, (1997): 729-746.

[11] R. Dubay, G. Kember, and B. Pramujati. Well-Conditioned Model Predictive Control. *ISA Transactions* 43, 1, (2004): 23-32.

[12] M. Abu-Ayyad and R. Dubay. SISO Extended Predictive Controllimplementation and the Basic Algorithm. *ISA Transactions* 45, 1, (2006): 9-20.

[13] M. Abu-Ayyad, R. Dubay, and G. Kember. SISO extended predictive control: Implementation and robust stability analysis. *ISA Transactions* 45, 3, (2006): 373-391.

[14] M. Abu-Ayyad and R. Dubay. MOMO Extended Predictive Control-Implementation and Robust Stability Analysis. *ISA Transactions* 45, 4, (2006): 545-561.

[15] E. F. Camacho and C. Bordons. *Model Predictive Control.* London: Springer-Verlag, 1999.

APPENDIX

Appendix

Two Tank Water Process Implemented in MATLAB and SEBOL

---

1 Program for Computing Dynamic Matrix

G_MATLAB.m

```
% =====================================================
%        COMPUTE DYNAMIC MATRIX BY MATLAB PROGRAM
%         TITO LEVEL CONTROL OF YOKOKAWA IN CSRL
%                 Author : SOPHEAK HEL
%                Date : 25 November, 2008
% =====================================================
%
clc
clear all; close all;
T = 1800;
n = 4;
P = 20;
M = 3;
Ts = 14;
la1 = 12;
la2 = 12;
% ================= DECLARE VARIABLE =================
%
ah11 = zeros(1,4);  ah22 = zeros(1,4);
b11 = zeros(1,3);   b12 = zeros(1,3);
b21 = zeros(1,3);   b22 = zeros(1,3);
%
dwf_g = zeros(2*P,1);
dwf_e = zeros(2*P,1);
f_g = zeros(2*P,1);
f_e = zeros(2*P,1);
%
w1 = zeros(P,1);      w2 = zeros(P,1);
yg1 = zeros(4,1);     yg2 = zeros(4,1);
ug1 = zeros(4,1);     ug2 = zeros(4,1);
del_ug1 = zeros(3,1);
del_ug2 = zeros(3,1);
%
ye1 = zeros(4,1);   ye2 = zeros(4,1);
ue1 = zeros(4,1);   ue2 = zeros(4,1);
del_ue1 = zeros(3,1);
del_ue2 = zeros(3,1);
%
n_model = zeros(2,4);
d_model = zeros(2,4);
Kp = zeros(2,2);
Tu = zeros(2,2);
Td = zeros(2,2);
```

```
delay = zeros(2,2);
epsilon = zeros(2,2);
%
% =================== INITIAL DATA ===================
%
F1 = zeros(P,3);
F2 = zeros(P,3);
%
Gp11 = zeros(P,2);  Gp12 = zeros(P,2);
Gp21 = zeros(P,2);  Gp22 = zeros(P,2);
%
G11_1 = zeros(P,1); G12_1 = zeros(P,1);
G21_1 = zeros(P,1); G22_1 = zeros(P,1);
%
G11 = zeros(P,M);   G12 = zeros(P,M);
G21 = zeros(P,M);   G22 = zeros(P,M);
%
% =================  MODEL PARAMETERS  ===============
%
Kp(1,1) = 3.2;  Kp(1,2) = 4.15;
Kp(2,1) = 3.1;  Kp(2,2) = 4.9;
%
Tu(1,1) = 452;  Tu(1,2) = 416;
Tu(2,1) = 472;  Tu(2,2) = 443;
%
Td(1,1) = 7;    Td(1,2) = 32;
Td(2,1) = 32;   Td(2,2) = 4;
%
% ===================================================
%               DIGITAL MODEL CONVERSION
% ===================================================
%
d1 = fix(min(Td(1,1)/Ts,Td(1,2)/Ts));
d2 = fix(min(Td(2,1)/Ts,Td(2,2)/Ts));
%
for i = 1:2
    for j = 1:2
        delay(i,j) = fix(Td(i,j)/Ts);
        epsilon(i,j) = Td(i,j)/Ts - delay(i,j);
    end
end
for i = 1:2
    for j = 1:2
        d_model(i,(2*j - 1)) = 1;
        x = - Ts/Tu(i,j);
        a = exp(x);
        d_model(i,2*j) = - a;
        y = - epsilon(i,j);
        alpha = a*(a^(y) - 1)/(1 - a);
        n_model(i,(2*j - 1)) = Kp(i,j)*(1 - a)*(1 - alpha);
        n_model(i,2*j) = Kp(i,j)*(1 - a)*alpha;
    end
end
%
ah11(1,1) = d_model(1,1)*d_model(1,3);
ah11(1,2) = d_model(1,1)*d_model(1,4) + d_model(1,2)*d_model(1,3) - d_model(1,1)*d_model(1,3);
ah11(1,3) = d_model(1,2)*d_model(1,4) - d_model(1,1)*d_model(1,4) - d_model(1,2)*d_model(1,3);
ah11(1,4) = - d_model(1,2)*d_model(1,4);
```

```
%
ah22(1,1) = d_model(2,1)*d_model(2,3);
ah22(1,2) = d_model(2,1)*d_model(2,4) + d_model(2,2)*d_model(2,3) - d_model(2,1)*d_model(2,3);
ah22(1,3) = d_model(2,2)*d_model(2,4) - d_model(2,1)*d_model(2,4) - d_model(2,2)*d_model(2,3);
ah22(1,4) = - d_model(2,2)*d_model(2,4);
%
b11(1,1) = n_model(1,1)*d_model(1,3);
b11(1,2) = n_model(1,1)*d_model(1,4) + n_model(1,2)*d_model(1,3);
b11(1,3) = n_model(1,2)*d_model(1,4);
%
b12(1,1) = d_model(1,1)*n_model(1,3);
b12(1,2) = d_model(1,1)*n_model(1,4) + d_model(1,2)*n_model(1,3);
b12(1,3) = d_model(1,2)*n_model(1,4);
%
b21(1,1) = n_model(2,1)*d_model(2,3);
b21(1,2) = n_model(2,1)*d_model(2,4) + n_model(2,2)*d_model(2,3);
b21(1,3) = n_model(2,2)*d_model(2,4);
%
b22(1,1) = d_model(2,1)*n_model(2,3);
b22(1,2) = d_model(2,1)*n_model(2,4) + d_model(2,2)*n_model(2,3);
b22(1,3) = d_model(2,2)*n_model(2,4);
%
% ======================================================
%      INITIAL VALUE FOR PREDICTIVE CONTROLLER
% ======================================================
%
G11_1(1 + d1,1) = b11(1,1);
G12_1(1 + d1,1) = b12(1,1);
G21_1(1 + d2,1) = b21(1,1);
G22_1(1 + d2,1) = b22(1,1);
for i = 1:3
    F1(1 + d1,i) = - ah11(1,i + 1);
    F2(1 + d2,i) = - ah22(1,i + 1);
end
for i = 1:2
    Gp11(1 + d1,i) = b11(1,i + 1);
    Gp12(1 + d1,i) = b12(1,i + 1);
    Gp21(1 + d2,i) = b21(1,i + 1);
    Gp22(1 + d2,i) = b22(1,i + 1);
end
%
for j = 2:P - d1
    for i = 1:3
        if i == 3
            F1(j + d1,i) = - F1(j + d1 - 1,1)*ah11(i + 1);
        else
        F1(j + d1,i) = F1(j + d1 - 1,i + 1) - F1(j + d1 - 1,1)*ah11(i + 1);
        end
    end
    for k = 1:2
        if k == 2
            Gp11(j + d1,k) = F1(j + d1 - 1,1)*Gp11(1 + d1,k);
            Gp12(j + d1,k) = F1(j + d1 - 1,1)*Gp12(1 + d1,k);
        else
        Gp11(j + d1,k) = Gp11(j + d1 - 1,k + 1) + F1(j + d1 - 1,1)*Gp11(1 + d1,k);
        Gp12(j + d1,k) = Gp12(j + d1 - 1,k + 1) + F1(j + d1 - 1,1)*Gp12(1 + d1,k);
        end
    end
```

```
    %
    G11_1(j + d1,1) = Gp11(j + d1 - 1,1) + F1(j + d1 - 1,1)*b11(1);
    G12_1(j + d1,1) = Gp12(j + d1 - 1,1) + F1(j + d1 - 1,1)*b12(1);
    %
end
for j = 2:P - d2
    for i = 1:3
        if i == 3
            F2(j + d2,i) = - F2(j + d2 - 1,1)*ah22(i + 1);
        else
        F2(j + d1,i) = F2(j + d1 - 1,i + 1) - F2(j + d1 - 1,1)*ah22(i + 1);
        end
    end
    for k = 1:2
        if k == 2
            Gp21(j + d2,k) = F2(j + d2 - 1,1)*Gp21(1 + d2,k);
            Gp22(j + d2,k) = F2(j + d2 - 1,1)*Gp22(1 + d2,k);
        else
        Gp21(j + d2,k) = Gp21(j + d2 - 1,k + 1) + F2(j + d2 - 1,1)*Gp21(1 + d2,k);
        Gp22(j + d2,k) = Gp22(j + d2 - 1,k + 1) + F2(j + d2 - 1,1)*Gp22(1 + d2,k);
        end
    end
    %
    G21_1(j + d2,1) = Gp21(j + d2 - 1,1) + F2(j + d2 - 1,1)*b21(1,1);
    G22_1(j + d2,1) = Gp22(j + d2 - 1,1) + F2(j + d2 - 1,1)*b22(1,1);
    %
end
%
g11_1 = zeros(P,1);
g12_1 = zeros(P,1);
g21_1 = zeros(P,1);
g22_1 = zeros(P,1);
for j = 1:M
    G11(:,j) = G11_1(:,1);
    G12(:,j) = G12_1(:,1);
    G21(:,j) = G21_1(:,1);
    G22(:,j) = G22_1(:,1);
    for i = 2:P
        g11_1(i,1) = G11_1(i - 1,1);
        g12_1(i,1) = G12_1(i - 1,1);
        g21_1(i,1) = G21_1(i - 1,1);
        g22_1(i,1) = G22_1(i - 1,1);
    end
    g11_1(1,1) = 0;
    g12_1(1,1) = 0;
    g21_1(1,1) = 0;
    g22_1(1,1) = 0;
    G11_1(:,1) = g11_1(:,1);
    G12_1(:,1) = g12_1(:,1);
    G21_1(:,1) = g21_1(:,1);
    G22_1(:,1) = g22_1(:,1);
end
G = [G11 G12; G21 G22];
Q = G'*G;
%
```

2 Program for Computing Static Gain Kg by GPC

Kg_MATLAB.m

```
%=====================================================
%           COMPUTE STATIC GAIN Kg BY GPC
%         TITO LEVEL CONTROL OF YOKOKAWA IN CSRL
%=====================================================
%
R = [la1*eye(M,M) zeros(M,M);zeros(M,M) la2*eye(M,M)];
Ag_condition = cond(Q + R);
Ag_determinant = det(Q + R);
Kg = inv(Q + R)*G';
%
lg1_w1 = Kg(1,1:P);
lg1_w2 = Kg(1,P + 1:2*P);
lg2_w1 = Kg(M + 1,1:P);
lg2_w2 = Kg(M + 1,P + 1:2*P);
Kg = [lg1_w1 lg1_w2; lg2_w1 lg2_w2];
```

---

3 Program for Computing Static Gain Ke by EPC

Ke_MATLAB.m

```
%=====================================================
%           COMPUTE STATIC GAIN Kg BY EPC
%         TITO LEVEL CONTROL OF YOKOKAWA IN CSRL
%=====================================================
%
a1 = Q(1,1);
a2 = Q(M + 1,M + 1);
%
R2_1 = 18.81;
R2_2 = 31.35;
R1 = Q(M + 1,M + 1)/le2;
%
W1 = [0 -R1*la1 -la1;-R1*la1 0 -R2_1*la1;-la1 -R2_1*la1 0];
W2 = [0 -R1*la2 -la2;-R1*la2 0 -R2_2*la2;-la2 -R2_2*la2 0];
W = [W1 zeros(M,M); zeros(M,M) W2];
%
A2 = Q(M + 1:2*M,M + 1:2*M) + W2;
A1 = [Q(1:M,1:M) + W1 zeros(M,M);zeros(M,M) A2];
Ae_condition = cond(Q + W);
Ae_determinant = det(Q + W);
Ke = inv(Q + W)*G';
%
le1_w1 = Ke(1,1:P);
le1_w2 = Ke(1,P + 1:2*P);
le2_w1 = Ke(M + 1,1:P);
le2_w2 = Ke(M + 1,P + 1:2*P);
Ke = [le1_w1 le1_w2; le2_w1 le2_w2];
```

---

4 Predictive Controller GPC and EPC in MATLAB program

GPC_MATLAB.m, EPC_MATLAB.m

```
% ==================================================
%       MAIN PROGRAM FOR GPC AND EPC CONTROLLER
% ==================================================
%
uoog1 = zeros(1,T); uoog2 = zeros(1,T);
yoog1 = zeros(1,T); yoog2 = zeros(1,T);
%*************************************
uooe1 = zeros(1,T); uooe2 = zeros(1,T);
yooe1 = zeros(1,T); yooe2 = zeros(1,T);
%
% ============ REFERENCE AND NOISE SIGNAL ===========
%
wr1 = [60*ones(1,400) 55*ones(1,400) 50*ones(1,400) 45*ones(1,T + P - 1200)];
wr2 = 65*ones(1,T);

Lg1.PV = 60;
Lg2.PV = 65;
Cg1.MV = 45;
Cg2.MV = 40;
%
Le1.PV = 60;
Le2.PV = 65;
Ce1.MV = 45;
Ce2.MV = 40;
%
uog1 = Cg1.MV;    uog2 = Cg2.MV;
yog1 = Lg1.PV;    yog2 = Lg2.PV;
%
for i = 1:n
    yg1(i,1) = yog1;
    yg2(i,1) = yog2;
    ug1(i,1) = uog1;
    ug2(i,1) = uog2;
end
%********************************
uoe1 = Ce1.MV;    uoe2 = Ce2.MV;
yoe1 = Le1.PV;    yoe2 = Le2.PV;
%
for i = 1:n
    ye1(i,1) = yoe1;
    ye2(i,1) = yoe2;
    ue1(i,1) = uoe1;
    ue2(i,1) = uoe2;
end
%
% ================= PLANT PROCESSING ================
%
for i = 2:T
    for p = 1:3
        del_ug1(p,1) = ug1(p + 1,1) - ug1(p,1);
        del_ug2(p,1) = ug2(p + 1,1) - ug2(p,1);
        %*************************************
        del_ue1(p,1) = ue1(p + 1,1) - ue1(p,1);
        del_ue2(p,1) = ue2(p + 1,1) - ue2(p,1);
    end
```

```
    yog1 = F1(1,1)*yg1(n) + F1(1,2)*yg1(n - 1) + F1(1,3)*yg1(n - 2) +...
           b11(1,1)*del_ug1(3) + b11(1,2)*del_ug1(2) + b11(1,3)*del_ug1(1) +...
           b12(1,1)*del_ug2(3) + b12(1,2)*del_ug2(2) + b12(1,3)*del_ug2(1);
    yoog1(i - 1) = yog1;
    %
    yog2 = F2(1,1)*yg2(n) + F2(1,2)*yg2(n - 1) + F2(1,3)*yg2(n - 2) +...
           b21(1,1)*del_ug1(3) + b21(1,2)*del_ug1(2) + b11(1,3)*del_ug1(1) +...
           b22(1,1)*del_ug2(3) + b22(1,2)*del_ug2(2) + b12(1,3)*del_ug2(1);
    yoog2(i - 1) = yog2;
    %***********************************************************************
    %
    yoe1 = F1(1,1)*ye1(n) + F1(1,2)*ye1(n - 1) + F1(1,3)*ye1(n - 2) +...
           b11(1,1)*del_ue1(3) + b11(1,2)*del_ue1(2) + b11(1,3)*del_ue1(1) +...
           b12(1,1)*del_ue2(3) + b12(1,2)*del_ue2(2) + b12(1,3)*del_ue2(1);
    yooe1(i - 1) = yoe1;
    %
    yoe2 = F2(1,1)*ye2(n) + F2(1,2)*ye2(n - 1) + F2(1,3)*ye2(n - 2) +...
           b21(1,1)*del_ue1(3) + b21(1,2)*del_ue1(2) + b11(1,3)*del_ue1(1) +...
           b22(1,1)*del_ue2(3) + b22(1,2)*del_ue2(2) + b12(1,3)*del_ue2(1);
    yooe2(i - 1) = yoe2;
%
% ======= SET REFERENCES BE CONSTANT IN RANGE 1:P ======
%
    for j = 1:P
        w1(j,1) = wr1(1,i);
        w2(j,1) = wr2(1,i);
    end
    for j = 1:n - 1
        yg1(j,1) = yg1(j + 1,1);
        yg2(j,1) = yg2(j + 1,1);
        %***********************
        ye1(j,1) = ye1(j + 1,1);
        ye2(j,1) = ye2(j + 1,1);
    end
    yg1(n,1) = yog1;
    yg2(n,1) = yog2;
    %**************
    ye1(n,1) = yoe1;
    ye2(n,1) = yoe2;
    %
    % =============== UPDATE FREE RESPONSE ================
    %
    for j = 1:P
        fyg1 = 0;
        fyg2 = 0;
        %%%%%%%%%
        fye1 = 0;
        fye2 = 0;
        for r = 1:3
            fyg1 = fyg1 + F1(j,r)*yg1(n + 1 - r,1);
            fyg2 = fyg2 + F2(j,r)*yg2(n + 1 - r,1);
            %***********************************
            fye1 = fye1 + F1(j,r)*ye1(n + 1 - r,1);
            fye2 = fye2 + F2(j,r)*ye2(n + 1 - r,1);
        end
        f_g(j,1) = Gp11(j,1)*del_ug1(n-1,1) + Gp11(j,2)*del_ug1(n-2,1) +...
                   Gp12(j,1)*del_ug2(n-1,1) + Gp12(j,2)*del_ug2(n-2,1) + fyg1;
        f_g(j + P,1) = Gp21(j,1)*del_ug1(n-1,1) + Gp21(j,2)*del_ug1(n-2,1) +...
```

```
                    Gp22(j,1)*del_ug2(n-1,1) + Gp22(j,2)*del_ug2(n-2,1) + fyg2;
    %**********************************************************************
    f_e(j,1) = Gp11(j,1)*del_ue1(n-1,1) + Gp11(j,2)*del_ue1(n-2,1) +...
               Gp12(j,1)*del_ue2(n-1,1) + Gp12(j,2)*del_ue2(n-2,1) + fye1;
    f_e(j + P,1) = Gp21(j,1)*del_ue1(n-1,1) + Gp21(j,2)*del_ue1(n-2,1) +...
                   Gp22(j,1)*del_ue2(n-1,1) + Gp22(j,2)*del_ue2(n-2,1) + fye2;
end
%
% ================ DIFFERENCE W AND F =================
%
for j = 1:P
    dwf_g(j,1) = w1(j,1) - f_g(j,1);
    dwf_g(j+P,1) = w2(j,1) - f_g(j+P,1);
    %**********************************
    dwf_e(j,1) = w1(j,1) - f_e(j,1);
    dwf_e(j+P,1) = w2(j,1) - f_e(j+P,1);
end
%
% ==================== CALCULATE U ====================
%
sg1 = 0;
sg2 = 0;
%%%%%%%%
se1 = 0;
se2 = 0;
for j = 1:2*P
    sg1 = sg1 + Kg(1,j)*dwf_g(j,1);
    sg2 = sg2 + Kg(2,j)*dwf_g(j,1);
    %*****************************
    se1 = se1 + Ke(1,j)*dwf_e(j,1);
    se2 = se2 + Ke(2,j)*dwf_e(j,1);
end
%
% ==================== CALCULATE U ====================
%
delta_ug1 = sg1;
delta_ug2 = sg2;
uog1 = uog1 + delta_ug1;
uog2 = uog2 + delta_ug2;
%**********************
delta_ue1 = se1;
delta_ue2 = se2;
uoe1 = uoe1 + delta_ue1;
uoe2 = uoe2 + delta_ue2;
%
uoog1(i - 1) = uog1;
uoog2(i - 1) = uog2;
%******************
uooe1(i - 1) = uoe1;
uooe2(i - 1) = uoe2;
%
if (uog1 < 0)
Cg1.MV = 0;
elseif (uog1 > 100)
Cg1.MV = 100;
else
Cg1.MV = uog1;
end
```

```
    %
    if (uog2 < 0)
    Cg2.MV = 0;
    elseif (uog2 > 100)
    Cg2.MV = 100;
    else
    Cg2.MV = uog2;
    end
    %
    uog1 = Cg1.MV;
    uog2 = Cg2.MV;
    %*************
    %
    if (uoe1 < 0)
    Ce1.MV = 0;
    elseif (uoe1 > 100)
    Ce1.MV = 100;
    else
    Ce1.MV = uoe1;
    end
    %
    if (uoe2 < 0)
    Ce2.MV = 0;
    elseif (uoe2 > 100)
    Ce2.MV = 100;
    else
    Ce2.MV = uoe2;
    end
    %
    uoe1 = Ce1.MV;
    uoe2 = Ce2.MV;
    %================== UPDATE U AND Y ==================
    %
    for j = 2:n
        ug1(j-1,1) = ug1(j,1);
        ug2(j-1,1) = ug2(j,1);
        %*********************
        ue1(j-1,1) = ue1(j,1);
        ue2(j-1,1) = ue2(j,1);
    end
    ug1(n,1) = uog1;
    ug2(n,1) = uog2;
    %%%%%%%%%%%%%%%%
    ue1(n,1) = uoe1;
    ue2(n,1) = uoe2;
end
%
% ===================================================
%    PLOT PROCESS TANK OUTPUTS AND CONTROL SIGNALS
% ===================================================
figure(1)
plot(yoog1,'b','LineWidth',2)
grid
axis([0 1600 40 65])
ylabel('Output LV1 (cm)')
xlabel('Time (secs)')
hold on
plot(yooe1,'-.r','LineWidth',2)
```

```
plot(wr1,'-.k','LineWidth',1)
legend('GPC Y1','EPC Y1',1)
hold off
%****************************
figure(2)
plot(yoog2,'b','LineWidth',2)
grid
axis([0 1600 63 67])
ylabel('Output LV2 (cm)')
xlabel('Time (secs)')
hold on
plot(yooe2,'-.r','LineWidth',2)
plot(wr2,'-.k','LineWidth',1)
legend('GPC Y2','EPC Y2',1)
hold off
%****************************
figure(3)
plot(uoog1,'b','LineWidth',2)
grid
axis([0 1600 41 47])
xlabel('Time (secs)')
ylabel('Control Signal MV1 (%)')
hold on
plot(uooe1,'-.r','LineWidth',2)
legend('GPC U1','EPC U1',1)
hold off
%****************************
figure(4)
plot(uoog2,'b','LineWidth',2)
grid
axis([0 1600 36 41])
xlabel('Time (secs)')
ylabel('Control Signal MV2 (%)')
hold on
plot(uooe2,'-.r','LineWidth',2)
legend('GPC U2','EPC U2',1)
hold off
```

5 Predictive Controller GPC and EPC in SEBOL program

GPC_SEBOL, EPC_SEBOL

```
! =================================================
!        DESIGN GPC AND EPC BY SEBOL PROGRAM
!                Author : SOPHEAK HEL
!               Date : 9 February 2009
! =================================================
! =================================================
!       TITO LEVEL CONTROL OF YOKOKAWA IN CSRL
! =================================================
!
#include <std.h>
global block PVI-DV L1
global block PVI-DV L2
global block MLD C1
```

```
global block MLD C2
global block _SFCSW GPC_MIMO
!
! ================= DECLARE VARIABLE =================
!
double ah11[1,4],ah22[1,4],b11[1,3],b12[1,3],b21[1,3],b22[1,3]
double y1[4,1],y2[4,1],u1[4,1],u2[4,1],del_u1[3,1],del_u2[3,1]
double uo1,uo2,yo1,yo2
double delta_u1,delta_u2
!
double a,alpha,x,y,Ts
double n_model[2,4],d_model[2,4]
double Kp[2,2],Tu[2,2],Td[2,2],epsilon[2,2]
!
! =============== INITIAL DATA FOR GPC ===============
!
!
double K[2,40],dwf[40,1],f[40,1]
double s1,s2,fy1,fy2,wr1,wr2
!
double F1[20,3],F2[20,3]
!
double Gp11[20,2],Gp12[20,2],Gp21[20,2],Gp22[20,2],w1[20,1],w2[20,1]
!
! =================        INDEX        =================
!
integer i,j,r
integer P,M,n,d1,d2
!
! =================  SET PARAMETERS  =================
!
n = 4
P = 20
M = 3
Ts = 14
d1 = 0
d2 = 0
epsilon[1,1] = 0.5000000
epsilon[1,2] = 0.2857143
epsilon[2,1] = 0.2857143
epsilon[2,2] = 0.2857143
!
Kp[1,1] = 3.2
Kp[1,2] = 4.15
Kp[2,1] = 3.1
Kp[2,2] = 4.9
!
Tu[1,1] = 452
Tu[1,2] = 416
Tu[2,1] = 472
Tu[2,2] = 443
!
Td[1,1] = 7
Td[1,2] = 32
Td[2,1] = 32
Td[2,2] = 4
!
! ==================================================
```

```
!               DIGITAL MODEL CONVERSION
! ====================================================
!
for i = 1 to 2
    for j = 1 to 2
        d_model[i,(2*j - 1)] = 1
        x = - Ts/Tu[i,j]
        a = exp(x)
        d_model[i,2*j] = - a
        y = - epsilon[i,j]
        alpha = a*(power(a,y) - 1)/(1 - a)
        n_model[i,(2*j - 1)] = Kp[i,j]*(1 - a)*(1 - alpha)
        n_model[i,2*j] = Kp[i,j]*(1 - a)*alpha
    next@
next@
!
ah11[1,1] = d_model[1,1]*d_model[1,3]
ah11[1,2] = d_model[1,1]*d_model[1,4] + d_model[1,2]*d_model[1,3] - d_model[1,1]*d_model[1,3]
ah11[1,3] = d_model[1,2]*d_model[1,4] - d_model[1,1]*d_model[1,4] - d_model[1,2]*d_model[1,3]
ah11[1,4] = - d_model[1,2]*d_model[1,4]
!
ah22[1,1] = d_model[2,1]*d_model[2,3]
ah22[1,2] = d_model[2,1]*d_model[2,4] + d_model[2,2]*d_model[2,3] - d_model[2,1]*d_model[2,3]
ah22[1,3] = d_model[2,2]*d_model[2,4] - d_model[2,1]*d_model[2,4] - d_model[2,2]*d_model[2,3]
ah22[1,4] = - d_model[2,2]*d_model[2,4]
!
b11[1,1] = n_model[1,1]*d_model[1,3]
b11[1,2] = n_model[1,1]*d_model[1,4] + n_model[1,2]*d_model[1,3]
b11[1,3] = n_model[1,2]*d_model[1,4]
!
b12[1,1] = d_model[1,1]*n_model[1,3]
b12[1,2] = d_model[1,1]*n_model[1,4] + d_model[1,2]*n_model[1,3]
b12[1,3] = d_model[1,2]*n_model[1,4]
!
b21[1,1] = n_model[2,1]*d_model[2,3]
b21[1,2] = n_model[2,1]*d_model[2,4] + n_model[2,2]*d_model[2,3]
b21[1,3] = n_model[2,2]*d_model[2,4]
!
b22[1,1] = d_model[2,1]*n_model[2,3]
b22[1,2] = d_model[2,1]*n_model[2,4] + d_model[2,2]*n_model[2,3]
b22[1,3] = d_model[2,2]*n_model[2,4]
!
! ====================================================
!               INITIAL VALUE FOR GPC
! ====================================================
!
for i = 1 to 3
    F1[1 + d1,i] = - ah11[1,i + 1]
    F2[1 + d2,i] = - ah22[1,i + 1]
next@
for i = 1 to 2
    Gp11[1 + d1,i] = b11[1,i + 1]
    Gp12[1 + d1,i] = b12[1,i + 1]
    Gp21[1 + d2,i] = b21[1,i + 1]
    Gp22[1 + d2,i] = b22[1,i + 1]
next@
!
! ====================================================
```

```
!               COMPUTE DYNAMIC MATRIX
! =====================================================
!
for j = 2 to P-d1
    for i = 1 to 3
        if (i == 3) then
           F1[j + d1,i] = - F1[j + d1 - 1,1]*ah11[1,i + 1]
        else
        F1[j + d1,i] = F1[j + d1 - 1,i + 1] - F1[j + d1 - 1,1]*ah11[1,i + 1]
        end if
    next@
    !
    for r = 1 to 2
        if (r == 2) then
           Gp11[j + d1,r] = F1[j + d1 - 1,1]*Gp11[1 + d1,r]
           Gp12[j + d1,r] = F1[j + d1 - 1,1]*Gp12[1 + d1,r]
        else
        Gp11[j + d1,r] = Gp11[j + d1 - 1,r + 1] + F1[j + d1 - 1,1]*Gp11[1 + d1,r]
        Gp12[j + d1,r] = Gp12[j + d1 - 1,r + 1] + F1[j + d1 - 1,1]*Gp12[1 + d1,r]
        end if
    next@
next@
!
for j = 2 to P-d2
    for i = 1 to 3
        if (i == 3) then
           F2[j + d2,i] = - F2[j + d2 - 1,1]*ah22[1,i + 1]
        else
        F2[j + d2,i] = F2[j + d2 - 1,i + 1] - F2[j + d2 - 1,1]*ah22[1,i + 1]
        end if
    next@
    !
    for r = 1 to 2
        if (r == 2) then
           Gp21[j + d2,r] = F2[j + d2 - 1,1]*Gp21[1 + d2,r]
           Gp22[j + d2,r] = F2[j + d2 - 1,1]*Gp22[1 + d2,r]
        else
        Gp21[j + d2,r] = Gp21[j + d2 - 1,r + 1] + F2[j + d2 - 1,1]*Gp21[1 + d2,r]
        Gp22[j + d2,r] = Gp22[j + d2 - 1,r + 1] + F2[j + d2 - 1,1]*Gp22[1 + d2,r]
        end if
    next@
next@
!
! =====================================================
!          ASSIGN STATIC GAIN K FROM MATLAB
!                   MAIN PROGRAM
! =====================================================
!
uo1 = C1.MV
uo2 = C2.MV
!
yo1 = L1.PV
yo2 = L1.PV
!
for i = 1 to n
    y1[i,1] = yo1
    y2[i,1] = yo2
    u1[i,1] = uo1
```

```
    u2[i,1] = uo2
next@
!
while(1<>0)
for i = 1 to n-1
    del_u1[i,1] = u1[i + 1,1] - u1[i,1]
    del_u2[i,1] = u2[i + 1,1] - u2[i,1]
next@
!
! ======= SET REFERENCES BE CONSTANT IN RANGE 1:P ======
!
wr1 = L1.SV
wr2 = L2.SV
for j = 1 to P
    w1[j,1] = wr1
    w2[j,1] = wr2
next@
for i = 1 to n-1
    y1[i,1] = y1[i+1,1]
    y2[i,1] = y2[i+1,1]
next@
y1[n,1] = L1.PV
y2[n,1] = L2.PV
!
! =============== UPDATE FREE RESPONSE ===============
!
for j = 1 to P
    fy1 = 0
    fy2 = 0
    for r = 1 to 3
        fy1 = fy1 + F1[j,r]*y1[n + 1 - r,1]
        fy2 = fy2 + F2[j,r]*y2[n + 1 - r,1]
    next@
    f[j,1] = Gp11[j,1]*del_u1[n-1,1] + Gp11[j,2]*del_u1[n-2,1] + \
             Gp12[j,1]*del_u2[n-1,1] + Gp12[j,2]*del_u2[n-2,1] + fy1
    f[j + P,1] = Gp21[j,1]*del_u1[n-1,1] + Gp21[j,2]*del_u1[n-2,1] + \
                 Gp22[j,1]*del_u2[n-1,1] + Gp22[j,2]*del_u2[n-2,1] + fy2
next@
!
! ================ DIFFERENCE W AND F ================
!
for j = 1 to P
    dwf[j,1] = w1[j,1] - f[j,1]
    dwf[j+P,1] = w2[j,1] - f[j+P,1]
next@
!
! =================== CALCULATE U ===================
!
s1 = 0
s2 = 0
for j = 1 to 2*P
    s1 = s1 + K[1,j]*dwf[j,1]
    s2 = s2 + K[2,j]*dwf[j,1]
next@
!
! =================== CALCULATE U ===================
!
delta_u1 = s1
```

```
delta_u2 = s2
uo1 = uo1 + delta_u1
uo2 = uo2 + delta_u2
!
if (uo1 < 0) then
C1.MV = 0
else if (uo1 > 100) then
C1.MV = 100
else
C1.MV = uo1
end if
!
if (uo2 < 0) then
C2.MV = 0
else if (uo2 > 100) then
C2.MV = 100
else
C2.MV = uo2
end if
!
uo1 = C1.MV
uo2 = C2.MV
!
!================== UPDATE U AND Y ===================
!
for i = 2 to n
    u1[i-1,1] = u1[i,1]
    u2[i-1,1] = u2[i,1]
next@
!
u1[n,1] = uo1
u2[n,1] = uo2
!
!y1[n,1] = L1.PV
!y2[n,1] = L2.PV

delay(1000)  ! 1 sec    (1000 = 1sec)
wend@
```

Biography

Sopheak Hel was born in Phnom Penh, Cambodia, in 1982. He received his Bachelor's degree in electrical engineering from Institute of Technology of Cambodia, Cambodia, in 2006. He has been granted a scholarship by the AUN/SEED-Net (www.seed-net.org) to pursue his Master's degree in electrical engineering at Chulalongkorn University, Thailand, since 2006. He conducted his graduate study with the Control Systems Research Laboratory, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University. His research interest includes the predictive control system, computer application and implementation in control engineering.