Analysis of Tuning Parameters of Model Predictive Controller (MPC)

by

Mohamad Hamizan bin Jamil

(SID: 13127)

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

JANUARY 2014

Chemical Engineering Department
Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original wok is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MOHAMAD HAMIZAN BIN JAMIL

ABSTRACT

Process optimization is very important in the engineering industries. As optimisation is achieved, less consumption of energy and utilities can be obtained for the process. In achieving optimisation, the response should be responded close to the reference values. The refineries nowadays consist mainly of multi variable unit process. Thus, to achieve optimisation using classical approach will be less reliable and time consuming. Hence, the introduction of Model Predictive Controller (MPC) to the process unit is more suitable compared to the classical approach. MPC is capable to solve high order problem and multivariable processes. The successful of MPC depends on the selection of tuning parameters. Therefore, by analysing the effect of each tuning parameters on the controller performance, promising performance of MPC can be produced. Firstly, the processes are selected from books as a case study to resemble the high order and multi variable problem processes. Then, the analysis will be done to study the effect of input weightage (u_{wt}), output weightage (y_{wt}), control horizon (M) and prediction horizon (P) on the controller performance. By changing one of the tuning parameter, the other tuning parameters have to be kept constant.

ACKNOWLEDGEMENT

Alhamdulillah, thanks to the Almighty, Allah s.w.t., this project is finally completed and all results were successfully obtained. All process flows were run smoothly according to the work timeline.

With the exposure to the reality of researching, it is very much understood that researching process is not as easy as it looks. Researching through this study has provided me tools and skills to experienced life as a researcher. Even though there are problems encountered along the way in completing this project, with the help of my supervisor, AP Dr M Ramasamy, such problems were managed to be handled wisely. With his assistance and guidance, much information and understanding of this project were successfully obtained and developed through simulation works and literature research. My utmost gratitude goes to him for spending lots of time monitoring all FYP students worked under his supervision.

Not to forget the coordinator, Dr Asna Md Zain for her arrangement and assistance for making all of this possible.

My sincere thank you also goes for all personnel who have directly or indirectly contributed to the completion of this project.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Background of Study

Currently, in industrial business, it is crucial to minimise cost in terms of operation or utilities and maintenance cost while maintaining the efficiency of the operation and mass production. However, there are obstacles in process industry that needs to be optimised to achieve the goals. Model Predictive Controller (MPC) has good track record in terms of controller design strategy as it provides good solving strategy towards difficult high-order problem and multivariable processes which resembles the industry nowadays especially oil and gas industry. Hence, by using MPC, the existence of problems commonly in the process industry can be overcome. MPC tends to minimise the cost, but at the same time maintaining the quality of product and produce mass quantity of products to meet the market demands.

The plant controller system layout consists of measured output (real process) and model. The model in MPC will then decide sequence of control moves by manipulating the changes in input so that the measured output moves in the trajectory line to achieve optimisation. Besides, calculations are being made on the predicted values of the output to compute a series of strategies to optimise the output behaviour of a plant.

Before designing the MPC, tuning parameters must be specified. Some key design issues and recommended value for the tuning parameters can be used to tune MPC controller. The tuning parameters that involved in tuning MPC controllers are Sampling Period (Δt) and Model Horizon (N), Control Horizon (M) and Prediction Horizon (P), and Reference Trajectory (α_i). In terms of Sampling Period (Δt) and Model Horizon (N) selection, the $N\Delta t$have to be equal to t_s, in which t_s refers to settling time for the open-loop response. The purpose of the selection is to ensure the model reflects the full effect of a change in an input variable over the time required to achieve steady state.

For Control Horizon (M) and Prediction Horizon (P), it can be seen that as control horizon (M) increases, the MPC controller will respond more aggressive and required computational effort increases too. By introducing input blocking, the computational effort can be reduced. Some typical requirements are $5 \leq M \leq 20$ and $N/3 < M < N/2$. In order to ensure full effect of the last input move is taken into account, the Prediction Horizon P is often chosen as $P = N + M$. The controller tends to become more aggressive when the value of Prediction Horizon P is decreasing.

By tuning Reference Trajectory (α_i), the impact can be seen on the desired speed of response for each output. The performance ratio concept can be used as an alternative in order to specify the Reference trajectory α_i.

A major difference between MPC and PID can be seen where performance of MPC can be measured by looking onto the simulation done using the model. However, appropriate model have to be selected to achieve the optimum performance. In addition, MPC itself is more complex than PID as the calculation has to be made at each sampling time.

1.2. Problem Statement

The successful of MPC depends on the selection of tuning parameters. Therefore, in order to produce promising performance of MPC, these tuning parameters need to be analysed and selected to achieve good performance of MPC.

1.3. Objective

The objective of the research is to analyse the tuning parameters that have impact to the performance of MPC. The purpose of analysing the tuning parameters is to decide the tuning parameters to enhance performance in MPC.

1.4. Scope of Study

In the research, one particular process has to be chosen. The process will has its own model to indicate the process. In designing MPC, a number of design parameters must be specified. The tuning parameters will then be analysed to observe its effect towards the performance of MPC.

CHAPTER 2

LITERATURE REVIEW

2.1. History of MPC

In the early 1960s, Kalman introduced Linear Quadratic Regulator (LQR). It applies infinite horizon which gives the ability of LQR to have powerful stability properties. The development of this technology do not contribute much to the control world, as the application of LQR itself that do not take into account on the constraints in its formulation, the nonlinearities of real system and the main factor is the lack of exposure towards optimal control concepts in the instrument technicians and engineers at that time.

Later, Model Predictive Heuristic Model being introduced by Richaletet.al (1978) and Dynamic Matrix Control (DMC) by Cutler and Ramaker (1980). Both of the controller have same properties by using dynamic model of process in which the past history response and step response introduced later is taken into account in order to predict the effect of future control actions. The control action being introduced is by minimising the predicted error but not exceeding the operational limit in system. The earlier versions of MPC were not automatically stabilizing. However, by manipulating the weights of the cost function, choosing a stable plant and keeping the horizon larger compared to settling time of plant, the stability then can be achieved. The next improvement of MPC being developed is the Quadratic Dynamic Matrix Control (QDMC; Garcia, Morshedi, 1986). Basically, the system is assumed to be linear and it used quadratic programming to solve constrained open-loop optimal control problem. In addition, the control and state constraints and quadratic cost are defined by linear inequalities. In 1970s, Aston et. al. had developed Minimum Variance Control. The main objective to achieve in the controller is to minimise the quadratic function of the error between most recent output and the prediction horizon. In addition, Generalized Minimum Variance Control (GMVC) was introduced to handle non-minimum phase plants by assigning penalized input to the objective function.

Peterka (1984) then developed Predictor-Based Self-Tuning control that overcame the horizon limitation. Alternatively, by not taking Diophantine equation as a based, Extended Proposal Self-Adaptive Control (EPSAC) was introduced by De Keyser et. al. (1985) that was using constant control signal starting from the present while using a sub-optimal predictor. In order to assure zero-steady state error, the input was replaced by the increment in control signal.

2.2. Stability Factor

One of the important issues being debated by the researches in the last decades is the stability of predictive control. It is very essential because the properties of finite horizon itself that is not assured to be stable and is achieved by tuning the weights and horizons. Besides, by using state-space relationship and analysing the influence of filter polynomials on robustness improvement, Mohtadi had explained the specific stability theorems of Generalized Predictive Control (GPC). Although it was well explained, the general stability property with finite horizons of predictive controller was still inadequate.

This has contributed to new predictive control method studies to ensure stability can be achieved starting from the year 1990. Some modifications had been made including the use of terminal constraints (Kwon et. al., 1983; Meadows et. al., 1995), the introduction of dual-mode designs (Mayne and Michalska, 1993) and the use of infinite prediction horizons (Rawling and Muske, 1993). By imposing end-point equality constraints on the output after a finite horizon, Clarke and Scattolini (1991) and Moscaet. al. (1990) had introduced stable predictive controllers. Meanwhile, the minimisation of the objective function as well as stabilizing the process had led to a stable formulation for GPC that was proposed by Kouvaritakiset. al. (1992). Most of the stated techniques achieve stability by introducing additional constraints and changing the design structure. However, these approaches should be avoided. Instead of modifying structure design, it is preferable to gain stability through tuning the parameters in the predictive controller.

2.3. Process Models in MPC

2.3.1. MPC Problem Formulation

$$min_{\Delta U(k)} J = \sum_{i=1}^{N}(\hat{y} - y_{ref})^2 Q + \sum_{i=1}^{M}(\Delta U)^2 R \quad (1)$$

N	=	Model Horizon
M	=	Control Horizon
$\hat{y}$	=	Measured Output
y_{ref}	=	Reference Output
Q	=	Output weighting matrix
R	=	Input weighting matrix

The basic formulation of MPC is well explained by using the above equation. There are numbers of tuning parameters that need to be analysed and selected so that optimization of process can be achieved.

The tuning parameters are:

1. Sampling Period and Model Horizon (N)
2. Control Horizon (M) and Prediction Horizon (P)
3. Reference Trajectory (α_i)
4. Weighting Matrix (Q and R) – the preferred variables that need to be controlled or manipulated depends on the process

Each of the tuning parameters is related to each other. Consequently, if one of the tuning parameters is changed, it will give an impact to the MPC formula thoroughly. Therefore, to achieve optimization, the best value of these tuning parameters must be selected as the success of MPC depends on the selection of tuning parameters.

The reference trajectory formula is given by the equation of:

$$y_{i,r}(k+j) = (\alpha_i)^j y_i(k) + [1 - (\alpha_i)^j] y_{i,sp}(k) \quad (2)$$

for i = 1,2,...,m and j= 1,2,...,P

$y_{i,r}$	=	ith element of reference trajectory
α_i	=	Filter constant ; $0 \leq \alpha_i \leq 1$
$y_{i,sp}$	=	Set point value

2.3.2. Model Predictive Control Law

The objective in MPC is to bring the measured output moves in the line of reference trajectory by using prediction. Therefore, the control calculation will be based on minimizing the prediction deviations from the reference trajectory. This can be explained by using this equation.

$$\hat{E}(k+1) \triangleq Y_r(k+1) - \tilde{Y}(k+1) \quad (3)$$

$\hat{E}(k+1)$	=	Predicted error vector
$Y_r(k+1)$	=	Reference trajectory
$\tilde{Y}(k+1)$	=	mP-dimensional vector of corrected predictions over the prediction horizon P

Whereas, $\hat{E}^0(k+1)$ is defined as the predicted unforced error which is the past impact of step change being introduced to the system. It also represents the predicted deviations from the reference trajectory when no further control action is taken.

$$\hat{E}^0(k+1) \triangleq Y_r(k+1) - \tilde{Y}^0(k+1) \quad (4)$$

$\tilde{Y}^0(k+1)$	=	mP-dimensional vector of corrected predictions for the unforced case.

$$\tilde{Y}^0(k+1) \triangleq \underbrace{\hat{Y}^0(k+1)}_{\text{Past control action}} + \underbrace{I\,[y(k-\hat{y}(k)]}_{\text{Future control action}} \quad (5)$$

By determining the changes in the manipulated input at sampling instant, we can strategies the control moves for the next M intervals.

$$\Delta U(k) \triangleq col[\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+m-1) \quad (6)$$

In objective function of unconstrained MPC, we have to minimise some (or all) these three types of error (Qin and Badgwell, 2003):

a. the predicted error over the predicted horizon, $\tilde{E}(k+1)$
b. the next M control moves, $\Delta U(k)$
c. the deviation of $u(k+i)$ from the desired steady state value (set point) over the control horizon

The main advantage in using the receding horizon approach is the new measured output will be used in instant for next move calculation. Therefore, it can minimise the error in calculation due to the presence of disturbance.

2.3.3. Single Input Single Output (SISO) Model

There are several models that are related to MPC. Generally, in industrial applications, the system is assumed to be linear and the empirical model is in the form of step-response model. By using step response model as a based, they can exhibit stable processes in the unusual dynamic behaviour which cannot be defined by simple transfer function model. However, the detriment of this model is the existence of large number of model parameters. The equation below is the SISO model, which assumed to be stable in the step-response model. The equation will explain on the prediction of future process behaviour.

$$y(k+1) = y_0 + \sum_{i=1}^{P} S_i \, \Delta u(k-i+1) + S_N \, u(k-N+1) \qquad (7)$$

$y(k+1)$	=	output variable at the sampling instant of k+1
$\Delta u(k-i+1)$	=	changes in the manipulated input from one sampling instant to the next
S_i	=	Step response coefficient at *i*
S_N	=	Step response coefficient at *N*
y_0	=	Initial value. For simplicity assume it to be zero.

The key in MPC is to predict the future outputs over prediction horizon, P. Therefore, predicted variable is included in the equation.

$$\hat{y}(k+1) = \sum_{i=1}^{P} S_i \, \Delta u(k-i+1) + S_N \, u(k-N+1) \qquad (8)$$

$\hat{y}(k+1)$ = predicted output variable at the sampling instant of k+1

However, when we introduce step change in the process, the past step change will have an impact to current sampling instant. Thus, the effect of past control actions cannot be neglected as they still produce impact to the response of model in current sampling instant. The equation can be expanded as,

$$\hat{y}(k+1) = \underbrace{S_i \Delta u\,(k)}_{\text{Effect of current action}} + \underbrace{\sum_{i=2}^{P} S_i \, \Delta u(k-i+1) + S_N \, u(k-N+1)}_{\text{Effect of past control action}} \qquad (9)$$

In general, for a j-step ahead prediction, it is well explained with the equation of,

$$\hat{y}(k+j) = \underbrace{\sum_{i=1}^{j} S_i \, \Delta u(k+j-i)}_{\text{Effect of current and future control action}} + \underbrace{\sum_{i=j+1}^{P} S_i \Delta u(k+j-i) \, S_N \, u(k+j-1)}_{\text{Effect of past control action}} \qquad (10)$$

For past control actions, it is called as predicted unforced response and denoted by the symbol of $\hat{y}_0(k+j)$; which cause the equation to be,

$$\hat{y}(k+j) = \sum_{i=1}^{j} S_i \, \Delta u(k+j-i) + \hat{y}_0(k+j) \qquad (11)$$

The above equations explained on the simple predictive controller that takes a basis on single prediction for J step ahead. However, in typical situation, MPC calculation is based on multiple predictions instead of single prediction. Thus, the vector matrix notation is introduced. For the next P sample instant,

$$\hat{Y}(k+1) \triangleq col \, [\hat{y}(k+1), \hat{y}(k+2), \ldots, \hat{y}(k+P)] \qquad (12)$$

where *col* defines a column vector. For predicted unforced responses, it is the same, in which the equation can be written as,

$$\hat{Y}^0(k+1) \triangleq col\ [\hat{y}^0(k+1), \hat{y}^0(k+2), \dots, \hat{y}^0(k+P)] \qquad (13)$$

whereas, $\Delta U(k)$, a vector of control actions of the input responses for the next M sampling instants is defined as,

$$\Delta U(k) \triangleq col\ [\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+M-1)] \qquad (14)$$

By calculating $\Delta U(k)$, to move the predicted output to the new set point in optimum manner, we can conclude the equation to be,

$$\hat{Y}(k+1) = S\Delta U(k) + \hat{Y}^0(k+1) \qquad (15)$$

2.3.4. Extension of Basic MPC Model (Integrating Processes)

The integrating process is being introduced to the formulation of MPC is because of bounded output rate of change. By a simple modification, the equation will be,

$$\Delta\hat{y}(k+1) = \sum_{i=1}^{P} S_i\, \Delta u(k-i+1) + S_N\, u(k-N+1) \qquad (16)$$

From the above equation, $\Delta\hat{y}(k+1)$ is being introduced as it provides an appropriate step response model for the integrating processes (Hokanson and Gerstle, 1992).

2.3.5. Extension of Basic MPC Model (Known Disturbance)

As been explained earlier, the key in MPC is to predict future output so that measured output moves towards reference trajectory to achieve optimisation. When measured output much deviated from the trajectory line, there are some disturbances present that cause the measured output to move away from the reference trajectory line. Thus, if the disturbance variable is known or can be measured, it should be included in the model.

$$\hat{y}(k+1) = \sum_{i=1}^{P} S_i \, \Delta u(k-i+1) + S_N u(k-N+1) + \sum_{i=1}^{P_d} S_i^d \, \Delta d(k-i+1) + S_N^d d(k-N_d+1)$$

N_d = Number of step-response coefficient for disturbance variable ($N_d \neq N$).

In case of multiple predictions, we have to predict on the future disturbances. Usually, the disturbances are assumed to be the same as current disturbance if there are no information on the next disturbances.

CHAPTER 3

METHODOLOGY

3.1. Research Methodology

The methodology used in analyzing the tuning parameters of MPC is by conducting simulation in MATLAB software. A process model is selected from the book to resemble the high order and multivariable problem. It will be used as a tool to analyse the tuning parameters in MPC. The analysis of the tuning parameters will lead to the best tuning parameters condition to achieve optimisation in MPC. High performance of MPC can be achieved, by determining the best tuning parameters condition in MPC.

The process model of project 1 has two inputs (u_1 and u_2) and two disturbance variables (y_1 and y_2). The transfer function of the process is given by $\frac{5.72e^{-14}}{60s+1}$ for the output while $\frac{1.52e^{-15}}{25s+1}$ for disturbance variables.

In process 2, The Wood-Berry Model is used to indicate the distillation column process which is given by the equation of:

$$\begin{bmatrix} X_D \\ X_S \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} R(s) \\ S(s) \end{bmatrix} + \begin{bmatrix} \frac{3.8e^{-8.1s}}{14.9s+1} \\ \frac{4.9e^{-3.4s}}{13.2s+1} \end{bmatrix} F(s)$$

where,

Controlled Variable $= X_D$ and X_S (distillate and bottom compositions)

Manipulated Variable = R and S (reflux flow rate and steam flow rate)

Disturbance Variable = F (feed flow rate)

3.2. Project Activities

3.2.1. Flowchart

The project starts with applying MPC to the model. There are four tuning parameters that need to be tested. ISE for each graph have to calculated and analysed to see the effect of the tuning parameters towards MPC performance. The flow of the project activities as per Figure 1:

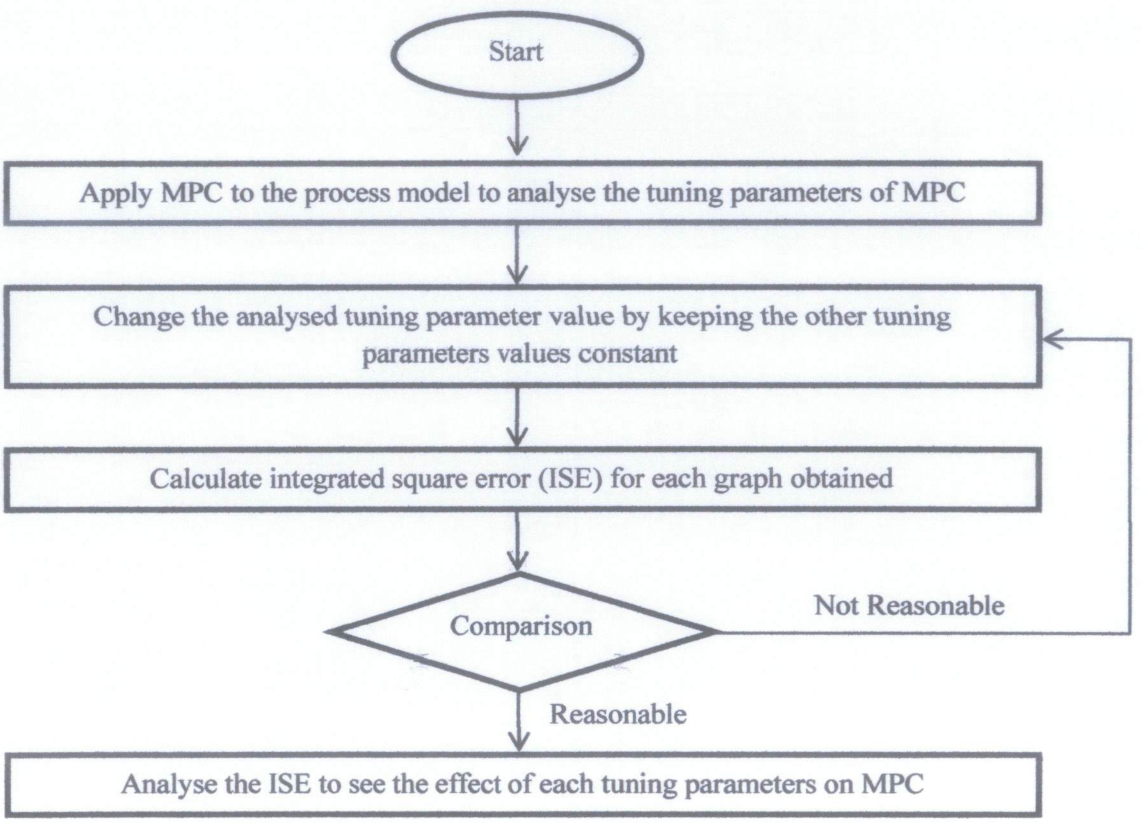

Figure 1: Flowchart for Project Simulation

3.2.2. Project 1

3.2.2.1. Apply MPC to the model

MPC is applied to the process model to analyse the tuning parameters of MPC. The process is tested for as long as 245 seconds, the sampling time is set to 7 and *nout* for both transfer functions is set to 1 as both of outputs are stable. The MATLAB coding is shown in APPENDIX.

3.2.2.2. Change the analysed tuning parameters value

In this project, there are four tuning parameters that need to be tuned. There are the output weightage (y_{wt}), input weightage (u_{wt}), control horizon (M) and prediction horizon (P). The first tuning parameters being tuned is the input weightage (u_{wt}) from 0 to 10 with the increment of 0.5. The other tuning parameters are kept constant with control horizon (M) is set to 5, prediction horizon (P) is set to 20 and output weightage (y_{wt}) is set to 1. The MATLAB coding is shown in APPENDIX. Graph will be generated in every simulation to study the effect of tuning parameters on performance of MPC graphically. Sample of graph is as shown in figure below.

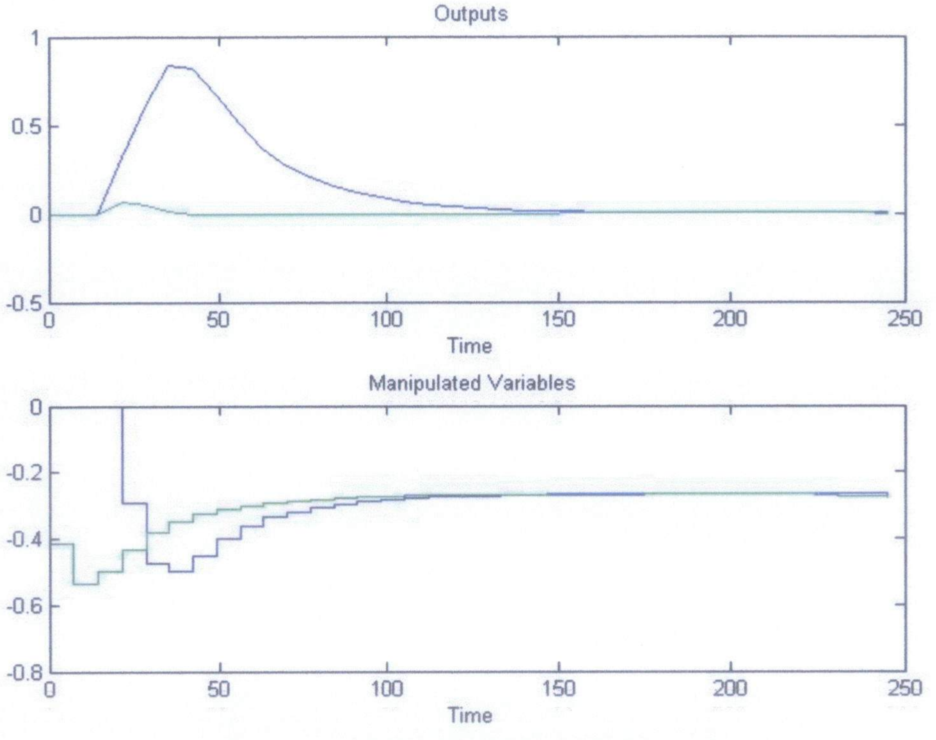

Figure 2: Sample of graph for Project 1

M=5 P = 20 ywt = 1 uwt= 1.0

3.2.2.3. Calculate ISE for each graph

ISE has to be calculated for each graph obtained to analyse the effect of each tuning parameters on performance of MPC.

3.2.3. Project 2

3.2.3.1. Apply MPC to the model

MPC is applied to the process model to analyse the tuning parameters of MPC. The process is tested for as long as 30 seconds, the sampling time is set to 2 and ny is set 2 because there are two measured outputs for the process. The input model and disturbance model is added up for the process. The MATLAB coding is shown in APPENDIX.

3.2.3.2. Change the analysed tuning parameters value

In this project, there are four tuning parameters that need to be tuned. There are the output weightage (y_{wt}), input weightage (u_{wt}), control horizon (M) and prediction horizon (P). The first tuning parameters being tuned is the output weightage (y_{wt}) by keeping the other tuning parameters constant. The output weightage (y_{wt}) is increase by 1.0 from 1 to 8. The other tuning parameters are kept constant with input weightage (uwt) is set to 1, control horizon (M) is set to 5and prediction horizon (P) is set to 10. The MATLAB coding is shown in APPENDIX. Graph will be generated in every simulation to study the effect of tuning parameters on performance of MPC graphically. Sample of graph is as shown in figure below.

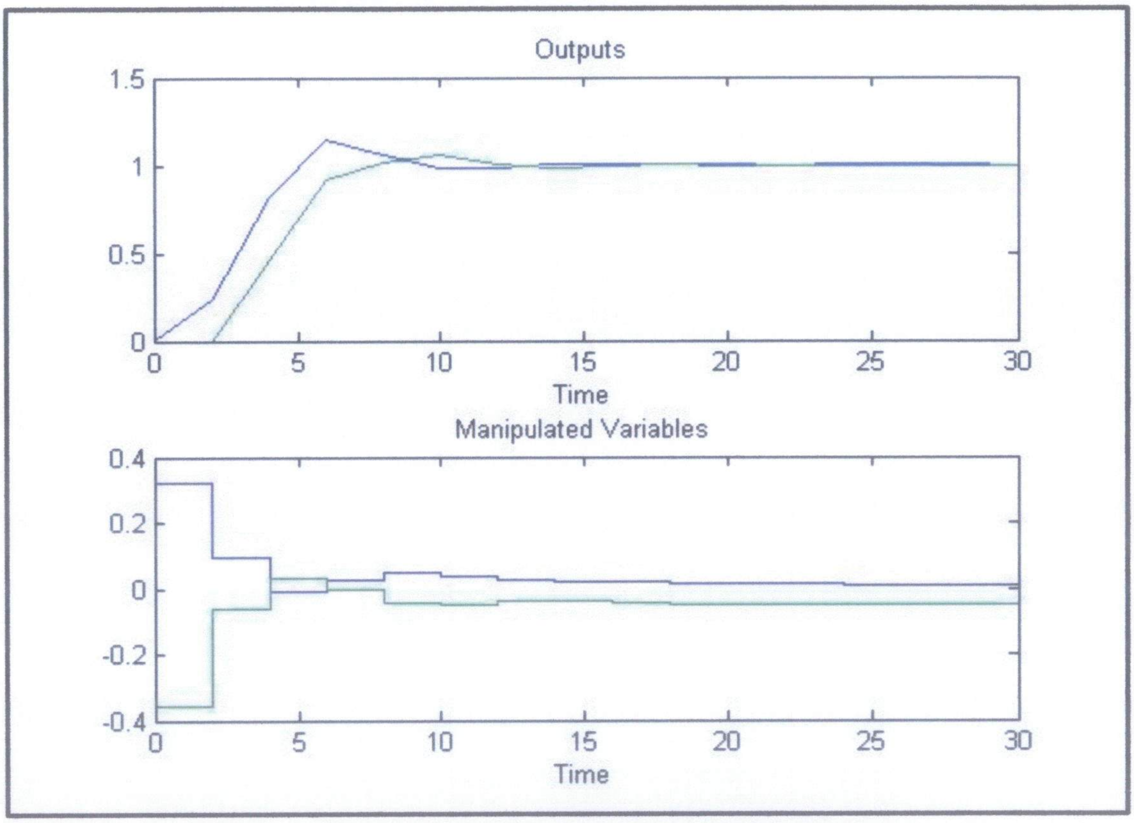

Figure 3: Sample of graph for Project 2

M= 5	P = 10	ywt1 = 1	ywt2 = 1	uwt1= 1	uwt2 = 1

3.2.3.3. Calculate ISE for each graph

ISE has to be calculated for each graph obtained to analyse the effect of each tuning parameters on performance of MPC.

3.3. Key Milestone

Table 1: Gantt chart

No	Project Activities	1	2	3	4	5	6	7		8	9	10	11	12	13	14	15
1	Selection of Process Model	■	■	■	■				Mid-semester break								
2	Apply MPC to Process Model					■	■										
3	Analysis of Tuning Parameters of MPC							■		■	■	■	■	■			
4	Submission of Progress Report									●							
5	Pre-EDX												●				
6	Submission of Draft Report													●			
7	Submission of Dissertation (Soft Bound)														●		
8	Submission of Technical Paper														●		
9	Oral Presentation															●	
10	Submission of Project Dissertation (Hard Bound)																●

■ Process ● Suggested Milestone

3.4. Tools

The tools required to develop this project:

Table 2: Tools required

Software	Purpose
Microsoft Excel 2010	Critical Analysis and Data Tabulation
MATLAB 7.12	Create Modelling of MPC, Analysis of Tuning Parameters of MPC and Simulation

CHAPTER 4

RESULTS AND DISCUSSION

4.1. Project 1

ISE for each graph with four variables, y_1 and y_2 (outputs) and u_1 and u_2 (disturbance variable) is calculated and tabulated in the Excel as per below.

Table 3: Effect of manipulating u_{wt} on controller performance

Changes in u_{wt} with increment of 0.5								
NO	M	P	y_{wt}	u_{wt}	ISE			
					y_1	y_2	u_1	u_2
1	5	20	1.00	0.00	2.2624	0.0000	3.2298	3.4588
2	5	20	1.00	0.50	2.8593	0.0064	3.1262	3.4404
3	5	20	1.00	1.00	3.4896	0.0238	3.0441	3.4257
4	5	20	1.00	1.50	3.9807	0.0503	2.9873	3.4055
5	5	20	1.00	2.00	4.4002	0.0897	2.9442	3.3790
6	5	20	1.00	2.50	4.8005	0.1452	2.9062	3.3479
7	5	20	1.00	3.00	5.2019	0.2179	2.8694	3.3137
8	5	20	1.00	3.50	5.6080	0.3071	2.8325	3.2778
9	5	20	1.00	4.00	6.0159	0.4107	2.7954	3.2412
10	5	20	1.00	4.50	6.4209	0.5257	2.7584	3.2048
11	5	20	1.00	5.00	6.8185	0.6491	2.7220	3.1692
12	5	20	1.00	5.50	7.2049	0.7780	2.6866	3.1349
13	5	20	1.00	6.00	7.5774	0.9100	2.6525	3.1022
14	5	20	1.00	6.50	7.9341	1.0431	2.6200	3.0712
15	5	20	1.00	7.00	8.2743	1.1756	2.5891	3.0419
16	5	20	1.00	7.50	8.5976	1.3065	2.5600	3.0144

17	5	20	1.00	8.00	8.9043	1.4349	2.5325	2.9886
18	5	20	1.00	8.50	9.1948	1.5603	2.5067	2.9644
19	5	20	1.00	9.00	9.4700	1.6825	2.4824	2.9418
20	5	20	1.00	9.50	9.7309	1.8013	2.4595	2.9205
21	5	20	1.00	10.00	9.9786	1.9169	2.4380	2.9005

Table 4: Effect of manipulating y_{wt} on controller performance

Changes in y_{wt} with increment of 0.5								
NO	M	P	y_{wt}	u_{wt}	ISE			
					y_1	y_2	u_1	u_2
1	5	20	1.00	1.00	3.4896	0.0238	3.0441	3.4257
2	5	20	1.50	1.00	3.0858	0.0113	3.0955	3.4357
3	5	20	2.00	1.00	2.8593	0.0064	3.1262	3.4404
4	5	20	2.50	1.00	2.7179	0.0039	3.1465	3.4437
5	5	20	3.00	1.00	2.6224	0.0025	3.1610	3.4464
6	5	20	3.50	1.00	2.5542	0.0016	3.1719	3.4485
7	5	20	4.00	1.00	2.5035	0.0011	3.1804	3.4503
8	5	20	4.50	1.00	2.4647	0.0008	3.1872	3.4517
9	5	20	5.00	1.00	2.4344	0.0006	3.1928	3.4528

Table 5: Effect of manipulating M on controller performance

Changes in M with increment of 1								
NO	M	P	y_{wt}	u_{wt}	ISE			
					y_1	y_2	u_1	u_2
1	1	20	1.00	1.00	8.3733	0.7561	2.3384	2.9377
2	2	20	1.00	1.00	3.2546	0.0143	3.0710	3.5181
3	3	20	1.00	1.00	3.2511	0.0144	3.0716	3.5189
4	4	20	1.00	1.00	3.4253	0.0132	3.0483	3.4619
5	5	20	1.00	1.00	3.4896	0.0238	3.0441	3.4257
6	6	20	1.00	1.00	3.4951	0.0314	3.0461	3.4113
7	7	20	1.00	1.00	3.4918	0.0341	3.0474	3.4072
8	8	20	1.00	1.00	3.4907	0.0346	3.0477	3.4064

9	9	20	1.00	1.00	3.4908	0.0345	3.0477	3.4064
10	10	20	1.00	1.00	3.4909	0.0344	3.0476	3.4066
11	11	20	1.00	1.00	3.4909	0.0344	3.0476	3.4066
12	12	20	1.00	1.00	3.4910	0.0344	3.0476	3.4067
13	13	20	1.00	1.00	3.4910	0.0344	3.0476	3.4067
14	14	20	1.00	1.00	3.4910	0.0344	3.0476	3.4067
15	15	20	1.00	1.00	3.4910	0.0344	3.0476	3.4067

Table 6: Effect of manipulating P on controller performance

Changes in P with increment of 1								
NO	M	P	y_{wt}	u_{wt}	ISE			
					y_1	y_2	u_1	u_2
1	5	10	1.00	1.00	3.4839	0.0322	3.0479	3.4099
2	5	11	1.00	1.00	3.4838	0.0309	3.0474	3.4119
3	5	12	1.00	1.00	3.4842	0.0298	3.0469	3.4138
4	5	13	1.00	1.00	3.4849	0.0288	3.0465	3.4156
5	5	14	1.00	1.00	3.4857	0.0278	3.0460	3.4173
6	5	15	1.00	1.00	3.4865	0.0270	3.0457	3.4190
7	5	16	1.00	1.00	3.4872	0.0262	3.0453	3.4205
8	5	17	1.00	1.00	3.4879	0.0255	3.0450	3.4220
9	5	18	1.00	1.00	3.4885	0.0249	3.0447	3.4233
10	5	19	1.00	1.00	3.4891	0.0244	3.0444	3.4246
11	5	20	1.00	1.00	3.4896	0.0238	3.0441	3.4257
12	5	21	1.00	1.00	3.4901	0.0234	3.0438	3.4268
13	5	22	1.00	1.00	3.4906	0.0230	3.0435	3.4278
14	5	23	1.00	1.00	3.4910	0.0226	3.0432	3.4287
15	5	24	1.00	1.00	3.4914	0.0222	3.0429	3.4295
16	5	25	1.00	1.00	3.4918	0.0219	3.0426	3.4302

As per tabulated in Table 3, by changing the input weightage (u_{wt}) from 0 to 10 with the increment of 0.5; keeping the other tuning parameters constant with control horizon (M) = 5, prediction horizon (P) = 20 and output weightage (y_{wt}) = 1, the ISE for y_1, y_2 and $u_1$increased. However, the ISE for u_2 decreased.

Then, in Table 4, changing the output weightage (y_{wt}) from 1 to 5 with the increment of 0.5; keeping the other tuning parameters constant with control horizon (M) = 5, prediction horizon (P) = 20 and input weightage (u_{wt}) = 1, the ISE for u_1 and u_2 increased. However, the ISE for y_1 and y_2 decreased.

After that, in Table 5, changing the control horizon (M) from 1 to 15 with the increment of 1.0; keeping the other tuning parameters constant with output weightage (y_{wt}) = 1, prediction horizon (P) = 20 and input weightage (u_{wt}) = 1, the ISE for y_1 and y_2 increased up to certain point before become constant until the end of simulation. However, the ISE for u_1 and u_2 decreased until certain point before become constant until the end of simulation.

Finally, in Table 6, changing the prediction horizon (P) from 10 to 25 with the increment of 1.0; keeping the other tuning parameters constant with output weightage (y_{wt}) = 1, control horizon (M) = 5 and input weightage (u_{wt}) = 1, the ISE for y_1 and u_2 increased. However, the ISE for u_1 and $y_2$decreased.

4.2. Project 2

ISE for each graph with eight variables, y_{11}, y_{12}, y_{21} and y_{22} (outputs) and u_{11}, u_{12}, u_{21} and u_{22} (manipulated variable) is calculated and tabulated in the Excel as per below.

Table 7: Effect of manipulating u_{wt1} on controller performance

Changes in u_{wt1} with increment of 1														
NO	M	P	y_{wt1}	y_{wt2}	u_{wt1}	u_{wt2}	ISE							
							y_{11}	y_{12}	y_{21}	y_{22}	u_{11}	u_{12}	u_{21}	u_{22}
1	5	10	1.00	1.00	1.00	1.00	14.0946	13.5144	13.5144	13.1755	0.1172	-0.1302	-0.1302	0.1579
2	5	10	1.00	1.00	2.00	1.00	13.8206	13.4966	13.4966	13.2316	0.0527	-0.0812	-0.0812	0.1740
3	5	10	1.00	1.00	3.00	1.00	13.6133	13.4669	13.4669	13.3333	0.0358	-0.0605	-0.0605	0.1857
4	5	10	1.00	1.00	4.00	1.00	13.4298	13.4337	13.4337	13.4455	0.0279	-0.0490	-0.0490	0.1940
5	5	10	1.00	1.00	5.00	1.00	13.2623	13.4002	13.4002	13.5572	0.0231	-0.0418	-0.0418	0.2000
6	5	10	1.00	1.00	6.00	1.00	13.1108	13.3682	13.3682	13.6624	0.0198	-0.0369	-0.0369	0.2045
7	5	10	1.00	1.00	7.00	1.00	12.9759	13.3389	13.3389	13.7584	0.0174	-0.0334	-0.0334	0.2080
8	5	10	1.00	1.00	8.00	1.00	12.8569	13.3126	13.3126	13.8447	0.0157	-0.0307	-0.0307	0.2108

Table 8: Effect of manipulating u_{wt2} on controller performance

Changes in u_{wt2} with increment of 1														
NO	M	P	y_{wt1}	y_{wt2}	u_{wt1}	u_{wt2}	ISE							
							y_{11}	y_{12}	y_{21}	y_{22}	u_{11}	u_{12}	u_{21}	u_{22}
1	5	10	1.00	1.00	1.00	1.00	14.0946	13.5144	13.5144	13.1755	0.1172	-0.1302	-0.1302	0.1579
2	5	10	1.00	1.00	1.00	2.00	13.9431	13.1645	13.1645	12.8299	0.1424	-0.1054	-0.1054	0.0939
3	5	10	1.00	1.00	1.00	3.00	13.8446	12.9029	12.9029	12.5994	0.1651	-0.0893	-0.0893	0.0701
4	5	10	1.00	1.00	1.00	4.00	13.7888	12.6978	12.6978	12.4258	0.1862	-0.0778	-0.0778	0.0583
5	5	10	1.00	1.00	1.00	5.00	13.7659	12.5333	12.5333	12.2862	0.2056	-0.0690	-0.0690	0.0515
6	5	10	1.00	1.00	1.00	6.00	13.7668	12.3978	12.3978	12.1664	0.2232	-0.0621	-0.0621	0.0472
7	5	10	1.00	1.00	1.00	7.00	13.7844	12.2832	12.2832	12.0576	0.2389	-0.0566	-0.0566	0.0443
8	5	10	1.00	1.00	1.00	8.00	13.8133	12.1843	12.1843	11.9549	0.2528	-0.0520	-0.0520	0.0423

Table 9: Effect of manipulating ywt1 on controller performance

Changes in y_{wt1} with increment of 1														
NO	M	P	y_{wt1}	y_{wt2}	u_{wt1}	u_{wt2}	ISE							
							y_{11}	y_{12}	y_{21}	y_{22}	u_{11}	u_{12}	u_{21}	u_{22}
1	5	10	1.00	1.00	1.00	1.00	14.0946	13.5144	13.5144	13.1755	0.1172	-0.1302	-0.1302	0.1579
2	5	10	2.00	1.00	1.00	1.00	14.3704	13.3167	13.3167	12.9620	0.3324	-0.1792	-0.1792	0.1194
3	5	10	3.00	1.00	1.00	1.00	14.5179	13.1282	13.1282	12.8518	0.5757	-0.1653	-0.1653	0.0872
4	5	10	4.00	1.00	1.00	1.00	14.6153	12.9841	12.9841	12.8044	0.8013	-0.1161	-0.1161	0.0746
5	5	10	5.00	1.00	1.00	1.00	14.6846	12.8775	12.8775	12.7882	0.9979	-0.0517	-0.0517	0.0795
6	5	10	6.00	1.00	1.00	1.00	14.7355	12.7987	12.7987	12.7866	1.1662	0.0171	0.0171	0.0967
7	5	10	7.00	1.00	1.00	1.00	14.7736	12.7401	12.7401	12.7913	1.3095	0.0845	0.0845	0.1213
8	5	10	8.00	1.00	1.00	1.00	14.8026	12.6960	12.6960	12.7840	1.4317	0.1477	0.1477	0.1494

Table 10: Effect of manipulating y_{wt2} on controller performance

Changes in y_{wt2} with increment of 1														
NO	M	P	y_{wt1}	y_{wt2}	u_{wt1}	u_{wt2}	ISE							
							y_{11}	y_{12}	y_{21}	y_{22}	u_{11}	u_{12}	u_{21}	u_{22}
1	5	10	1.00	1.00	1.00	1.00	14.0946	13.5144	13.5144	13.1755	0.1172	-0.1302	-0.1302	0.1579
2	5	10	1.00	2.00	1.00	1.00	14.2127	13.8259	13.8259	13.5560	0.0940	-0.1468	-0.1468	0.2662
3	5	10	1.00	3.00	1.00	1.00	14.2987	13.9509	13.9509	13.6986	0.0928	-0.1630	-0.1630	0.3550
4	5	10	1.00	4.00	1.00	1.00	14.3539	14.0189	14.0189	13.7736	0.0960	-0.1806	-0.1806	0.4366
5	5	10	1.00	5.00	1.00	1.00	14.3890	14.0601	14.0601	13.8190	0.1000	-0.1979	-0.1979	0.5132
6	5	10	1.00	6.00	1.00	1.00	14.4121	14.0870	14.0870	13.8488	0.1038	-0.2142	-0.2142	0.5850
7	5	10	1.00	7.00	1.00	1.00	14.4282	14.1057	14.1057	13.8698	0.1074	-0.2293	-0.2293	0.6520
8	5	10	1.00	8.00	1.00	1.00	14.4400	14.1193	14.1193	13.8852	0.1107	-0.2434	-0.2434	0.7140

Table 11: Effect of manipulating M on controller performance

Changes in M with increment of 1														
NO	M	P	y_{wt1}	y_{wt2}	u_{wt1}	u_{wt2}	ISE							
							y_{11}	y_{12}	y_{21}	y_{22}	u_{11}	u_{12}	u_{21}	u_{22}
1	1	10	1.00	1.00	1.00	1.00	8.5007	8.5826	8.5826	8.7003	0.0186	-0.0273	-0.0273	0.0446
2	2	10	1.00	1.00	1.00	1.00	13.8744	13.2442	13.2442	12.8170	0.1060	-0.1212	-0.1212	0.1537
3	3	10	1.00	1.00	1.00	1.00	14.0441	13.4916	13.4916	13.1467	0.1044	-0.1221	-0.1221	0.1566
4	4	10	1.00	1.00	1.00	1.00	14.0923	13.5129	13.5129	13.1718	0.1166	-0.1305	-0.1305	0.1594
5	5	10	1.00	1.00	1.00	1.00	14.0946	13.5144	13.5144	13.1755	0.1172	-0.1302	-0.1302	0.1579
6	6	10	1.00	1.00	1.00	1.00	14.0999	13.5179	13.5179	13.1787	0.1177	-0.1306	-0.1306	0.1582
7	7	10	1.00	1.00	1.00	1.00	14.0990	13.5170	13.5170	13.1780	0.1179	-0.1307	-0.1307	0.1581
8	8	10	1.00	1.00	1.00	1.00	14.0984	13.5167	13.5167	13.1780	0.1179	-0.1307	-0.1307	0.1581

Table 12: Effect of manipulating P on controller performance

Changes in P with increment of 1														
NO	M	P	y_{wt1}	y_{wt2}	u_{wt1}	u_{wt2}	ISE							
							y_{11}	y_{12}	y_{21}	y_{22}	u_{11}	u_{12}	u_{21}	u_{22}
1	5	7	1.00	1.00	1.00	1.00	14.0946	13.5135	13.5135	13.1753	0.1178	-0.1306	-0.1306	0.1580
2	5	8	1.00	1.00	1.00	1.00	14.0920	13.5128	13.5128	13.1756	0.1175	-0.1304	-0.1304	0.1579
3	5	9	1.00	1.00	1.00	1.00	14.0926	13.5135	13.5135	13.1758	0.1173	-0.1303	-0.1303	0.1579
4	5	10	1.00	1.00	1.00	1.00	14.0946	13.5144	13.5144	13.1755	0.1172	-0.1302	-0.1302	0.1579
5	5	11	1.00	1.00	1.00	1.00	14.0972	13.5153	13.5153	13.1747	0.1171	-0.1302	-0.1302	0.1578
6	5	12	1.00	1.00	1.00	1.00	14.0999	13.5161	13.5161	13.1738	0.1172	-0.1301	-0.1301	0.1578
7	5	13	1.00	1.00	1.00	1.00	14.1026	13.5167	13.5167	13.1727	0.1172	-0.1302	-0.1302	0.1578
8	5	14	1.00	1.00	1.00	1.00	14.1051	13.5173	13.5173	13.1717	0.1173	-0.1302	-0.1302	0.1578

As per tabulated in Table 7, by changing the input weightage 1 (u_{wt1}) from 1 to 8with the increment of 1.0; keeping the other tuning parameters constant with control horizon (M) = 5, prediction horizon (P) = 10, input weightage 2 (u_{wt2}) = 1, output weightage 1 (y_{wt1}) = 1 and output weightage 2 (y_{wt2}) = 1, the ISE for y_{22}, u_{12}, $u_{21}$and u_{22}increased. However, the ISE for y_{11}, y_{12}, y_{21} and u_{11} decreased.

Then, in Table 8, changing the input weightage 2 (u_{wt2}) from 1 to 8with the increment of 1.0; keeping the other tuning parameters constant with control horizon (M) = 5, prediction horizon (P) = 10, input weightage 1 (u_{wt1}) = 1, output weightage 1 (y_{wt1}) = 1 and output weightage 2 (y_{wt2}) = 1, the ISE for u_{11}, u_{12} and u_{21}increased. However, the ISE for y_{12}, y_{21}, y_{22} and u_{22} decreased.

After that, in Table 9, changing the output weightage1 (y_{wt1}) from 1 to 8with the increment of 1.0; keeping the other tuning parameters constant with control horizon (M) = 5, prediction horizon (P) = 10, input weightage 1 (uwt1) = 1, input weightage 2 (y_{wt2}) = 1 and output weightage 2 (y_{wt2}) = 1, the ISE for y_{11}, u_{12} and u_{21}increased. However, the ISE for y_{12}, y_{21}, y_{22} and u_{22} decreased.

Next, in Table 10, changing the output weightage 2 (y_{wt2}) from 1 to 8with the increment of 1.0; keeping the other tuning parameters constant with control horizon (M) = 5, prediction horizon (P) = 10, input weightage 1 (u_{wt1}) = 1, input weightage 2 (y_{wt2}) = 1 and output weightage 1 (y_{wt1}) = 1, the ISE for y_{11}, y_{12}, y_{21}, y_{22}, u_{11} and u_{22}increased. However, the ISE for u_{12} and u_{21} decreased.

Then, in Table 11, changing the control horizon (M) from 1 to 8with the increment of 1.0; keeping the other tuning parameters constant with prediction horizon (P) = 10, input weightage 1 (u_{wt1}) = 1, input weightage 2 (u_{wt2}) = 1, output weightage 1 (y_{wt1}) = 1 and output weightage 2 (y_{wt2}) = 1, the ISE for y_{11}, y_{12}, y_{21}, y_{22}, u_{11} and u_{22}increased. However, the ISE for u_{12}, u_{21} and y_{22} decreased.

Finally, in Table 12, changing the prediction horizon (P) from 7 to 14with the increment of 1.0; keeping the other tuning parameters constant with control horizon (M) = 5, input weightage 1 (u_{wt1}) = 1, input weightage 2 (u_{wt2}) = 1, output weightage 1 (y_{wt1}) = 1 and output weightage 2 (y_{wt2}) = 1, the ISE for y_{11}, y_{12}, y_{21}, u_{12} and u_{21}increased. However, the ISE for u_{11}, u_{22} and y_{22} decreased.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions and Recommendations

Optimisation provides the best solution towards the problem or design of process mainly in industrial decision making because of the most cost-effective solution feature. It is essential for the process operation to generate maximum production to gain maximum profit without eliminating the efficient operating of the process, so that the consumption of energy is least. MPC has a good track record in solving high order problem and multi-variable processes.

MPC is the best approach towards solving the problem to achieve optimisation as the feature of MPC itself in focusing the responses towards the optimum value. The study on the effect of tuning parameters which are input weightage (u_{wt}), output weightage (y_{wt}), control horizon (M) and prediction horizon (P) on the controller performance have been successfully studied. The successful implementation of the tuning parameters in the MPC will lead to more profitable market industries.

The analyses are just done from problems taken from books based on several justifications to decide the tuning parameters. Then, every tuning parameter has to be changed by keeping other tuning parameters constant to study the effect on MPC performance. The study will be more accurate if it is done on the real industrial problem which has much order and multi variables. Besides, the industry will benefit more and further research can be made from the industrial problem to assist the market industries in achieving optimization.

REFERENCES

Seborg D.E., Edgar, T.F., Mellichamp D.A. (2004). Process Dynamics and Control (Second Edition). John Wiley & Sons.

Nunes G.C. (2001). Design and Analysis of Multivariable Predictive Control Applied to an Oil-Water-Gas Separator: A Polynomial Approach. University of Florida.

Qin S.J., Badgewell T.A (2002). A Survey of Industrial Model Predictive Control Technology. Pergamon.

Cutler, C.R., Ramaker, B.C. (1980). Dynamic Matrix Control - A Computer Control Algorithm, Automatic Control Conference, San Francisco, CA.

Kalman, R. E. (1960). Contributions to the theory of optimal control.

Bulletin de la Societe Mathematique de Mexicana, 5, 102–119.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Transactions of ASME, Journal of Basic Engineering, 87, 35–45.

Richalet, J., Rault, A., Testud, J.L., Papon, J. (1978).Model Predictive Heuristic Control: Application to Industrial Processes, Automatica, 14, 2, 413-428.

Garcia, C. E., & Morshedi, A. M. (1986). Quadratic programming solution of dynamic matrix control (QDMC). Chemical Engineering Communications, 46, 73–87.

Peterka, V. (1984).Predictor-based Self-tuning Control," Automatica, 20, 1, 39-50.

De Keyser, R.M.C., Van Cuawenberghe, A.R., (1985).Extended Prediction Self-adaptive Control, IFAC: Symposium on Identification and System Parameter Estimation, York, UK, 1317-1322.

Kwon, W.H., Pearson, A.E. (1977).A Modified Quadratic Cost Problem and Feedback Stabilization of a Linear System, IEEE Transactions on Automatic Control, AC-22, 5,838-842.

Rawlings, J.B., Muske, K.R. (1993). The Stability of Constrained Receding Horizon Control, IEEE Transactions on Automatic Control, 38, 10.

Muske, K.R., Rawlings, J. (1993).Model Predictive Control with Linear Models, AIChE Journal, 39, 262-287.

Clarke, D.W., Scattolini, R. (1991). Constrained Receding-horizon Predictive Control, Proceedings IEE, 138, 4, 347-354.

Mosca, E., Lemos, J.M., Zhang, J. (1990).Stabilizing I/O Receding Horizon Control, IEEE Conference on Decision and Control.

Kouvaritakis, B., Rossiter, J.A., Chang, A.O.T. (1992). Stable Generalized Predictive Control:An Algorithm with Guaranteed Stability, Proceedings IEE, Part D, 139, 4, 349-362.

APPENDIX

MATLAB 7.12.0 (R2011a)

File Edit Debug Parallel Desktop Window Help

Current Folder: C:\Program Files\MATLAB\R2011a\bin

Shortcuts How to Add What's New

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> delt1=0;
delay1=14;
num1=5.72;
den1=[60 1];
g=poly2tfd(num1,den1,delt1,delay1);
tfinal=245;
delt2=7;
nout1=1;
plant=tfd2step(tfinal,delt2,nout1,g);
delay2=15;
num2=1.52;
den2=[25 1];
gd=poly2tfd(num2,den2,delt1,delay2);
delt2=7;
nout2=1;
dplant=tfd2step(tfinal,delt2,nout2,gd);
model=plant;
```

Figure 4: MATLAB coding to apply MPC to Project 1

```
gd=poly2tfd(num2,den2,delt1,delay2);
delt2=7;
nout2=1;
dplant=tfd2step(tfinal,delt2,nout2,gd);
model=plant;
ywt=0.5; uwt=0;
M=5; P=20;
Kmpc1=mpccon(model,ywt,uwt,M,P);
tend=245;
r=[ ]; usat=[ ]; tfilter=[ ];
dmodel=[ ];
dstep=1;
[y1,u1]=mpcsim(plant,model,Kmpc1,tend,r,usat,tfilter,...
dplant,dmodel,dstep);
dmodel=dplant; % measured disturbance
[y2,u2]=mpcsim(plant,model,Kmpc1,tend,r,usat,tfilter,...
dplant,dmodel,dstep);
plotall([y1,y2],[u1,u2],delt2);
pause;
 Percent error in the last step response coefficient
 of output yi for input uj is :
    2.1%
 Percent error in the last step response coefficient
 of output yi for input uj is :
   0.01%
 Time remaining 245/245
 Time remaining 175/245
 Time remaining 105/245
 Time remaining 35/245
Simulation time is 0.002 seconds.
 Time remaining 245/245
 Time remaining 175/245
 Time remaining 105/245
 Time remaining 35/245
Simulation time is 0.002 seconds.
fx >> model=plant;
```

Start

Figure 5: MATLAB coding in tuning one of the tuning parameters for Project 1

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> delt = 2;
ny = 2;
g11 = poly2tfd(12.8,[16.7 1],0,1);
g21 = poly2tfd(6.6,[10.9 1],0,7);
g12 = poly2tfd(-18.9,[21.0 1],0,3);
g22 = poly2tfd(-19.4,[14.4 1],0,3);
umod = tfd2mod(delt,ny,g11,g21,g12,g22);
% Defines the effect of u inputs
g13 = poly2tfd(3.8,[14.9 1],0,8);
g23 = poly2tfd(4.9,[13.2 1],0,3);
dmod = tfd2mod(delt,ny,g13,g23);
% Defines the effect of w input
pmod = addumd(umod,dmod); % Combines the two models.
imod = pmod;% assume perfect modeling
```

Figure 6: MATLAB coding in defining the process for Project 2

```
ywt = [1 1] ; % weights on both outputs
uwt = [1 1]; % weights on both inputs
P = 10; % prediction horizon
M = 5; % control horizon
Ks = smpccon(imod,ywt,uwt,M,P);
tend=30; % time period for simulation.
r = [1 0]; % setpoints for the two outputs.
[y1,u1] = smpcsim(pmod,imod,Ks,tend,r);
plotall(y1,u1,delt)
```

Figure 7: MATLAB coding to apply MPC to Project 2

APPENDIX (Continued)

```
>> imod = pmod;% assume perfect modeling
ywt = [1 2] ; % weights on both outputs
uwt = [1 1]; % weights on both inputs
P = 10; % prediction horizon
M = 5; % control horizon
Ks = smpccon(imod,ywt,uwt,M,P);
tend=30; % time period for simulation.
r = [1 0]; % setpoints for the two outputs.
[y2,u2] = smpcsim(pmod,imod,Ks,tend,r);
plotall(y2,u2,delt)
>>  imod = pmod;% assume perfect modeling
ywt = [1 3] ; % weights on both outputs
uwt = [1 1]; % weights on both inputs
P = 10; % prediction horizon
M = 5; % control horizon
Ks = smpccon(imod,ywt,uwt,M,P);
tend=30; % time period for simulation.
r = [1 0]; % setpoints for the two outputs.
[y3,u3] = smpcsim(pmod,imod,Ks,tend,r);
plotall(y3,u3,delt)
```

Figure 8: MATLAB coding in tuning one of the tuning parameters for Project 2