

**Control of Inverse Response Process using Model Predictive Controller  
(Simulation)**

by

Fawwaz bin Fuat

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

SEPTEMBER 2012

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 work 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.

---

FAWWAZ BIN FUAT

## **ABSTRACT**

Model predictive control is an important model-based control strategy devised for large multiple-input, multiple-output control problems with inequality constraints on the input and outputs. Applications typically involve two types of calculations: (1) a steady-state optimization to determine the optimum set points for the control calculations, and (2) control calculations to determine the input changes that will drive the process to the set points. The success of model-based control strategies such as MPC depends strongly on the availability of a reasonably accurate process model. Consequently, model development is the most critical step in applying MPC.

As Rawlings (2000) has noted, “feedback can overcome some effects of poor model, but starting with a poor process model is a kind to driving a car at night without headlight.” Finally the MPC design should be chosen carefully.

Model predictive control has had a major impact on industrial practice, with over 4500 applications worldwide. MPC has become the method of choice for difficult control problems in the oil refining and petrochemical industries. However, it is not a panacea for all difficult control problem(Shinkey, 1994; Hugo, 2000). Furthermore, MPC has had much less impact in the order process industries. Performance monitoring of MPC systems is an important topic of current research interest.

## **ACKNOWLEDGEMENT**

Grateful.....that's the best word that I can say to appear my grateful after finished completed final year project report at Universiti Teknologi PETRONAS. Firstly, I would like to show my appreciation to my supervisor, Dr Lemma Dendena Tufa for supporting me throughout this two semester and constantly provides me with assistance and help. He is not only sacrificed their working time to supervise me to complete this project with success but also taught me some moral virtues and how to behave a good employee in an organization.

Secondly, I would like to thank the Chemical Engineering Department in University of Technology PETRONAS (UTP) for giving me the opportunities to prepare this technical report as my final year project. It's has been help me a lot in refreshing what I have learnt throughout five years I am here.

Furthermore, I would also like to express my thankfulness to my beloved parents for giving me mentally supports to carry on this task.

Other than that, a special thanks to the reviewers of this report who contributed significantly to project and to all of my friends, who offered valuable suggestion to finished this report.

Finally, I would like to appear my hope that I can apply what I learnt when finished my study soon.

## TABLE OF CONTENTS

CERTIFICATION . . . . .	I
ABSTRACT . . . . .	II
ACKNOWLEDGEMENT . . . . .	III
CHAPTER 1: INTRODUCTION . . . . .	1
1.1. Project Background . . . . .	1
1.2. Problem Statements . . . . .	1
1.3. Objectives and Scope of Study. . . . .	2
CHAPTER 2: LITERATURE REVIEW . . . . .	3
CHAPTER 3: METHODOLOGY . . . . .	16
3.1. Project Work . . . . .	16
3.2. Research Methodology . . . . .	18
3.3. Activities/Gantt Chart and Milestone . . . . .	19
CHAPTER 4: RESULT AND DISCUSSION . . . . .	20
4.1. Data gathering and Analysis	
4.2. Result and Discussion	
CHAPTER 5: CONCLUSION AND RECOMMENDATION	45
5.1. Conclusion	
5.2. Recommendation	
REFERENCES . . . . .	46
APPENDICES	

## LIST OF FIGURES

Figure 1: Basic Structure of Model Predictive Controller

Figure 2: Model Predictive Controller block diagram

Figure 3: Project activities flow

Figure 4: MATLAB Simulation

Figure 5: Effect of  $T_i$

Figure 6: Block diagram

Figure 7: Effect of  $T_i$  with Model Predictive Controller

Figure 8: Graph with Input rate weight Tuning

Figure 9: Graph with Output rate weight Tuning

Figure 10: Graph with Prediction Horizon Tuning

Figure 11: Graph with Control Horizon Tuning

Figure 12: Feedback Control Loop

Figure 13: Servo Control

Figure 14: Graphical Interpretation of IAE

Figure 15: Input (bottom) and output (top) data used for identification

Figure 16: The Bode plot of the actual model

Figure 17: Bode plot of the identified model and the actual model

## LIST OF TABLE

Table 1: Gantt Chart

Table 2: Error with different  $T_i$

Table 3: Error with default MPC setting with  $T_i$  Tuning

Table 4: Error with Input rate weight Tuning

Table 5: Error with Output rate weight Tuning

Table 6: Error with Prediction Horizon Tuning

Table 7: Error with Control Horizon Tuning

# CHAPTER 1

## INTRODUCTION

### 1.1 Project Background

Systems which have inverse responses are difficult to control. Many types of process show inverse response over a part of the normal range operation among these is the recycling system so common in engineering practice. Control can be greatly improved by introducing a compensating feedback element into the control loop. In this project, we will use Model Predictive Controller to control the inverse response processes.

### 1.2 Problem Statement

Some effect to the behavior of the process is affected by the tuning of the Model Predictive Controller. The time taken to reach the set point and the error may be affected by the parameter tuning for inverse response process. Based on the system, we have to measure the error of the controller and make a comparison of efficiency between response with and without using controller.

### 1.3 Objectives and Scope of Study

- I. Analyze the effect of  $T_i$  response without using controller for a typical inverse response process by MATLAB software.
- II. Analyze the effect of  $T_i$  response by using Model Predictive Controller for a typical inverse response process by MATLAB software.
- III. Analyze the effect of input rate weight response by using Model Predictive Controller for a typical inverse response process by MATLAB software.



- IV. Analyze the effect of output rate weight response by using Model Predictive Controller for a typical inverse response process by MATLAB software.
- V. Analyze the effect of prediction horizon response by using Model Predictive Controller for a typical inverse response process by MATLAB software.
- VI. Analyze the effect of control horizon response by using Model Predictive Controller for a typical inverse response process by MATLAB software.

In order to study process control of inverse responses, the transfer functions should first be developed. The transfer function which is the ratio of the output variable  $X$  to the input signal  $Y$  in the Laplace Transformations is defines the performance of the process. These equations represent a variety of typical combined processes or their approximations. For this project, we will use second order equation of transfer function.

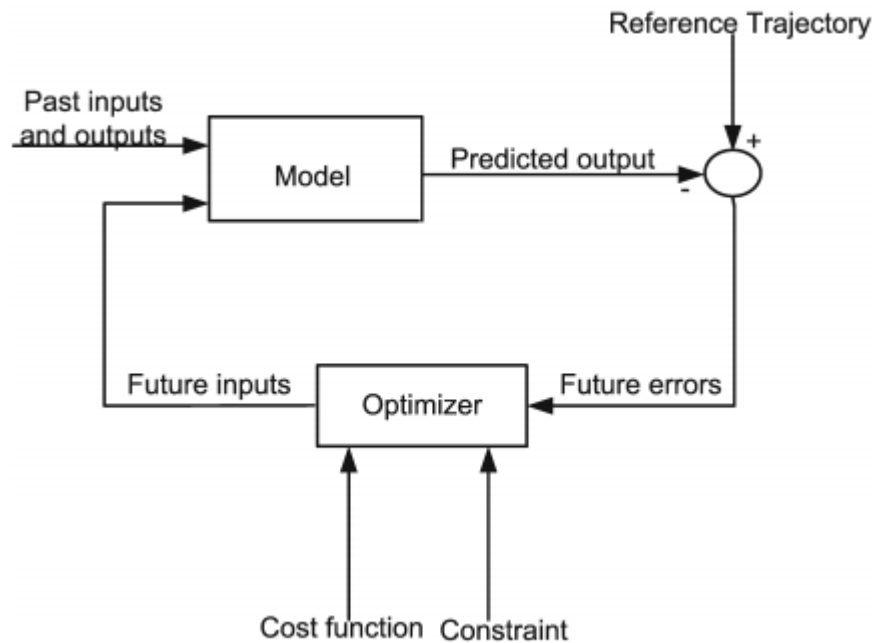
## **CHAPTER 2**

### **LITERATURE REVIEW**

The analysis, design, control, operation, and optimization of distillation columns was studied and researched for long time ago [1,2]. As time goes by, engineers try to come out with techniques and approaches to stabilize the control process due to the present of nonlinearities and constraints. The most famous and popular method of model predictive control is using the Dynamic Matrix Control (DMC) [3,4]. It is based on a discrete time step response model that calculates a desired value of manipulated value that remains unchanged during the next time step.

One of the applications of DMC is in multiple input and multiple output systems (MIMO). In simulation experiment, MIMO is always compared in term of performance with Single Input and Single Output (SISO) process system and the standard of model predictive control. By this performance measurement, we can know which controller is the most suitable for certain control process.

Besides, DMC introduced by Culter and Ramaker is available in most commercial industrial distributed control systems and process simulation software packages [5]. The three integral parts of any model predictive control algorithm which are process model, the cost function and the optimization technique. There are various forms of MPC such as Robust MPC that provides guaranteed feasibility and stability of the process, feedback MPC that mitigates shrinkage of feasible region, pre-computed MPC that using linear or quadratic programming, and decentralized MPC that can give fast response.



Basic structure of MPC

The major contributions of MPC in control system industry are it handles structural changes, easy to tune method, allow operation closer to constraint that can increase the profit, can take account of actuator limitations, has plenty of time for on-line computations, can handle non-minimal phase and unstable processes, and can handle multivariable control problems naturally. The applications of MPC are always used in distillation column, hydrocracker, pulp and paper plant, servo mechanism and robot arm.

The OBF models can be considered as a generalization of Finite Impulse Response (FIR) models in which the filter are replaced with more complex orthonormal basis filters [6,7]. There are many types of OBF such as Laguerre filter which has one real pole for damped processes, Kautz filter which allows the incorporation of a pair conjugate complex pole for modeling weakly damped processes, Generalized orthonormal basis filter and Markov OBF which for a system involves time delay and time delay estimation.

In many control application the desired performance cannot be expressed caused by the process constraint [8,9]. Example for physical constraint is the actuator limits,

safety constraint such as temperature and pressure limits, performance constraint such as overshoot and others constraints like manipulated variable constraint which hard limits on input, manipulated variable rate constraint which are hard limit on the size of the manipulated variable, and output variable constraint which hard or soft limits on the output on the system are imposed to.

Based on my research, the advantages of using MPC are it is a straight forward formulation based on well understood concepts, it can handle constraints, the development time much shorter than for competing advanced control methods, easier to maintain the changing model or the specs does not require complete redesign [10]. By running different scenarios in linear and nonlinear simulations, you can evaluate controller performance. You can adjust controller performance as it runs by tuning weights and varying constraints [11].

In addition to developing more flexible control technology, new process identification technology was developed to allow quick estimation of empirical dynamic models from test data, substantially reducing the cost of model development [12]. More importantly, MPC theory can lead to discoveries by which MPC can be improved through new formulation [13]. This is important to make sure the efficiency of the control process in minimizing the constraint and improve the control system in industry or any manufacturing.

## I. Model Predictive Control

### Introduction to MPC

A general development is presented that gives great insight into the roles of both the control algorithm and the process in the behavior of feedback systems. It also provides a method for tailoring the feedback control algorithm to each specific application. Since the model of the process is an integral part of the control algorithm, the controller equation structure depends on the process model. Although the control algorithm is different, the feedback concept is unchanged, and the selection criteria for manipulated

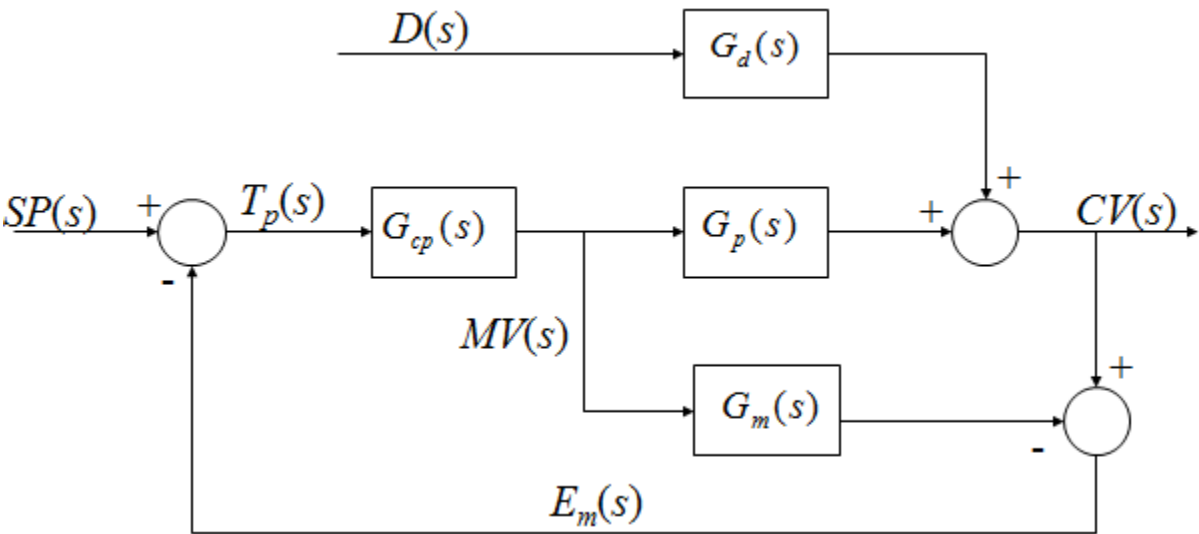
and controlled variables are the same. The algorithms could be used as replacements for the PID controller in nearly all applications. PID controller algorithm is considered the standard algorithm. Alternative algorithm is selected only when it provides better performance.

MPC Structure

Consider the typical thought process used by a human operator implementing a feedback control manually. The approach used by the operator has three important characteristics:

- I. It uses a model of the process to determine the proper adjustment to the manipulated variable, because the future behavior of the controlled variable can be *predicted* from the values of the manipulated variable.
- II. The important feedback information is the difference between the predicted model response and the actual process response. If this difference is zero, the control would be perfect, and no further correction would be needed
- III. This feedback approach can result in the controlled variable approaching its set point after several iterations, even with modest model errors

A continuous version of the approach can be automated with the general predictive control structure.



The variable  $E_m$  is equal to the effect of the disturbance,  $G_d(s)D(s)$ , if the model is perfect ( $G_m(s)=G_p(s)$ ). The structure highlights the disturbance for feedback correction.

However, the model is essentially never exact. The feedback signal includes the effect of the disturbance and the model error, or mismatch. The feedback signal can be considered as a model correction. It is used to correct the set point so as to provide a better target value,  $T_p(s)$ , to the predictive control algorithm. The controller calculates the value of the manipulated variable based on the corrected target.

The closed-loop transfer functions for the setpoint and disturbances are:

$$\begin{aligned} \frac{CV(s)}{SP(s)} &= \frac{G_{cp}(s)G_v(s)G'_p(s)}{1 + G_{cp}(s)[G_v(s)G'_p(s)G_s(s) - G_m(s)]} \\ &= \frac{G_{cp}(s)G_p(s)}{1 + G_{cp}(s)[G_p(s) - G_m(s)]} \end{aligned}$$

$$\begin{aligned} \frac{CV(s)}{D(s)} &= \frac{[1 - G_{cp}(s)G_m(s)]G_d(s)}{1 + G_{cp}(s)[G_v(s)G'_p(s)G_s(s) - G_m(s)]} \\ &= \frac{[1 - G_{cp}(s)G_m(s)]G_d(s)}{1 + G_{cp}(s)[G_p(s) - G_m(s)]} \end{aligned}$$

The controller algorithm,  $G_{cp}(s)$ , for the predictive structure is to be determined to give good dynamic performance. Let us determine a few properties of the predictive structure that establish important general features of its performance and give guidance for designing the controller. A very important control performance objective is to ensure that the controlled variable returns to its set point in steady state. This objective can be evaluated from the closed-loop transfer functions by applying the final value theorem

and determining whether the final value of the controlled variable, expressed as a deviation variable from the initial set point, reaches the set point.

The next control performance objective is perfect control. Perfect control means the controlled variable never deviates from the set point.

$CV(s)/D(s) = 0$  and  $CV(s)/SP(s) = 1$  provide the basis for the following condition:

$$G_{cp}(s) = G_m^{-1}(s)$$

Perfect control can be achieved if the controller could be set equal to the inverse of the process dynamic model. Block diagram algebra can be applied to derive the following condition for the behavior of the manipulated variable.

$$\begin{aligned} \frac{MV(s)}{D(s)} &= \frac{-G_d(s)G_{cp}(s)}{1 + G_{cp}(s)[G_p(s) - G_m(s)]} \\ &= \frac{-G_d(s)G_{cp}(s)}{1 + G_{cp}(s)G_p(s) - 1} = \frac{-G_d(s)}{G_p(s)} \end{aligned}$$

The perfect control system must invert the true process in some manner.

## II. Inverse Response

Inverse response behavior appears when the initial response of the output variable is in the opposite direction to the steady state value. In chemical process industry this phenomenon occurs in several systems such as distillation column and drum boilers [14]. The reason for the inverse response is that the process transfer function has an odd number of zeros in the open right half plane. This non-minimum phase characteristics of the process affects the achievable closed-loop performance because the controller operates on wrong sign information in the initial time of the transient [15]. This fact introduces essential limitations in terms of achievable output performance.

Let assume a process resulting from two parallel first-order stable processes having opposite gain

$$P(s) = P_1(s) - P_2(s) = \frac{K_1}{\tau_1 s + 1} - \frac{K_2}{\tau_2 s + 1}$$

Where  $K_1$ ,  $K_2$ ,  $\tau_1$ , and  $\tau_2$  are positive constants. The overall transfer function of  $P(s)$  in equation above can be posed as

$$P(s) = K_p \frac{-\alpha s + 1}{(\tau_1 s + 2)(\tau_2 s + 2)}$$

Where

$$K_p = K_1 - K_2$$



And

$$\alpha = \frac{(K_2\tau_1 - K_1\tau_2)}{(K_1 - K_2)}$$

In this context, inverse response appears due to competing effects of slow and fast dynamics. In concrete terms, it appears when the slower process has higher gain.

Therefore, the condition for inverse response reads as:

$$1 < \frac{K_1}{K_2} < \frac{\tau_1}{\tau_2} \quad \text{or} \quad 1 < \frac{K_2}{K_1} < \frac{\tau_2}{\tau_1}$$

Study of Other Controller

Dynamic Matrix Control (DMC)

DMC is one of the most popular methods of model predictive control. A way to have students explore the nature of DMC control is to use it on a simulated process [16].

DMC control is based on a discrete time step response model that calculates a desired value of the manipulated value that remains unchanged during the next time step. The new value of the manipulated variable the value that gives the smallest sum of squares error between the set point and the predicted value predicted values of the controlled variable. The dynamic model used to predict the future values of the controlled variable is represented by a vector A, whose elements are defined as

$$a_i = \frac{\Delta y(t_i)}{\Delta u(t_0)}$$

Where  $\Delta y(t_i) = y(t_i) - y(t_0)$

$y(t)$  = the value of the controlled variable at time  $t$

$\Delta y(t_0)$  = the change in the manipulated variable at  $t_0$

Thus, the response of a process to a step change  $\Delta u$ , in the manipulated variable at  $t_0$   $\Delta u(t_0)$  is given by

$$\begin{bmatrix} \Delta y(t_1) \\ \Delta y(t_2) \\ \Delta y(t_3) \\ \vdots \\ \Delta y(t_n) \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \Delta u(t_0)$$

#### Orthonormal Basis Control (OBF)

General orthonormal basis filter (OBF) models are generalization of FIR models, presented a least square identification method to estimate a finite number of model parameters based on Pulse, Laguerre and Kautz basis functions. They showed that the form of the chosen basis functions and the accuracy of the poles used in them can improve the identification results.

A test was simulated on the transfer function given and an output data was generated for a given input data sequence with additive white noise of power 2.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{34 e^{-8s}}{25.5s + 1}$$

An OBF model was developed using the input-output data with an initial time constant  $\tau=3$ , and time delay=0. It is observed that the resulting OBF model with only 5 OBF terms has a very good accuracy.

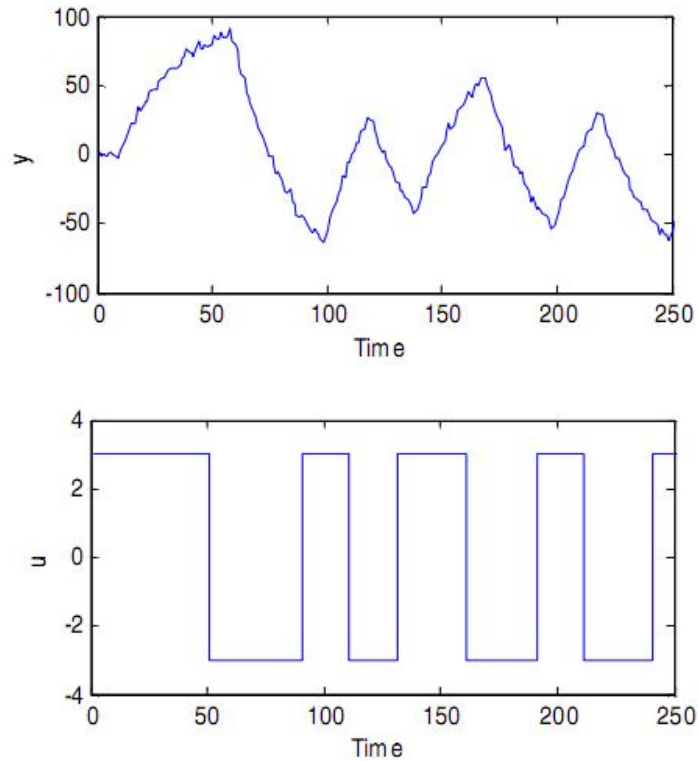


Figure 15: Input (bottom) and output (top) data used for identification

The Bode plot of the actual model and that which is identified by the discussed technique is presented below. As it can be seen, through there is small difference in the time constants, and gain however, the Bode plots are nearly indistinguishable, hence good enough for controller design purpose.

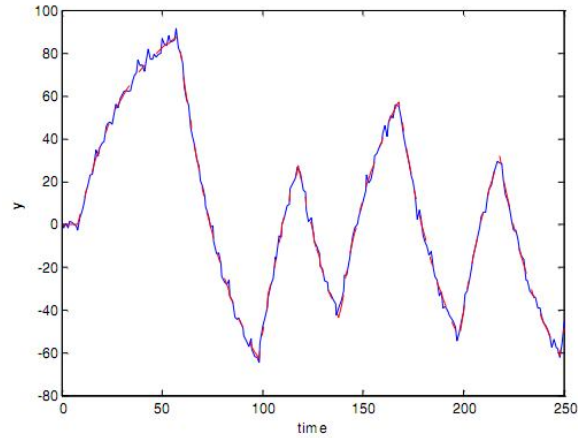


Figure 16: The Bode plot of the actual model

The response of the OBF model and the noisy output data plotted together for the same input

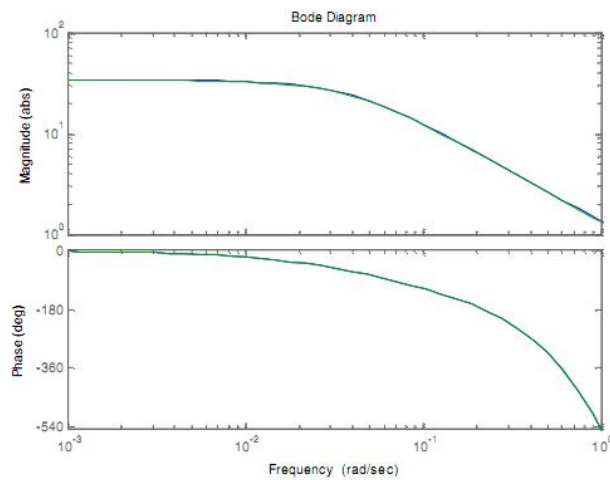


Figure 17: The Bode plot of the identified model and the actual model

### Internal Method Controller (IMC)

IMC controller is based on Brosilow (1979) and Garcia and Morari (1983). Since an exact inverse is not possible, the IMC approach segregates and eliminates the aspects of the model transfer function that make the calculation of realizable inverse impossible.

Thus, the first step to start the IMC approach is to factor the model into the product of two factors.

$$G_m(s) = G_m^+(s)G_m^-(s)$$

- $G_m^+(s)$  The noninvertible part has an inverse that is not causal or is unstable
- The steady state gain of this term must be 1.0
- $G_m^-(s)$  The invertible part has an inverse that is causal and stable, leading to realizable, stable controller

- The IMC Controller (idealized)

$$\tilde{G}_{cp}(s) = [G_m^-(s)]^{-1}$$

- This design ensures the controller is realizable and the system is internally stable

Example of IMC Controller

$$G_m(s) = \frac{0.039}{(5s+1)^3}$$

$$G_m^-(s) = \frac{0.039}{(5s+1)^3} \quad G_m^+(s) = 1.0$$

$$G_{cp}(s) = [G_m^-(s)]^{-1} = \frac{(5s+1)^3}{0.039}$$

- To make the controller proper or semiproper, add a filter to make the controller proper

$$G_{cp}(s) = \tilde{G}_{cp}(s) = [G_m^-(s)]^{-1} G_f(s)$$

- For tracking setpoint changes,

$$G_f(s) = \frac{1}{(\lambda s + 1)^n}$$

- Adjust the filter-tuning parameter to vary speed of the response of the closed-loop system

- $\lambda$  : small --- response is fast  
           large --- the closed loop response is more robust (insensitive to model error)

Example:

$$G_f(s)G_{cp}(s) = \frac{1}{0.039} \frac{10.5s + 1}{\lambda s + 1}$$

$$G_m(s) = \frac{0.039e^{-5.5s}}{(10.5s + 1)} \quad G_m^-(s) = \frac{0.039}{(10.5s + 1)} \quad G_m^+(s) = e^{-5.5s}$$

$$\tilde{G}_{cp}(s) = [G_m^-(s)]^{-1} = \frac{(10.5s + 1)}{0.039}$$

The controller is proportional-derivative, which still might be too aggressive but can be modified to give acceptable performance. Then, to make the controller semiproper,

$$G_{cp}(s) = \frac{(10.5s + 1)}{0.039(\lambda s + 1)}$$

# CHAPTER 3

## METHODOLOGY/PROJECT WORK

### 3.1 Project work

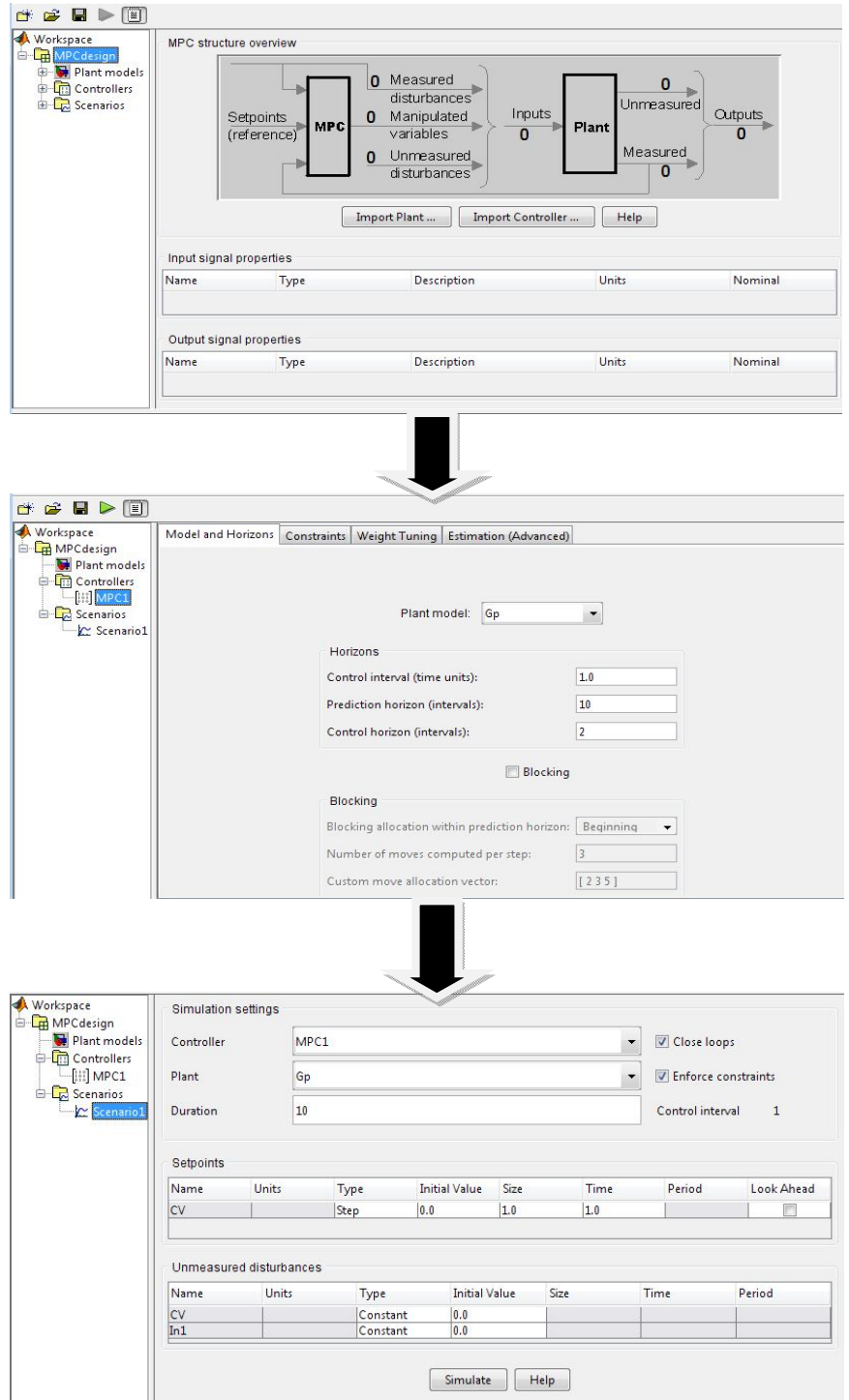
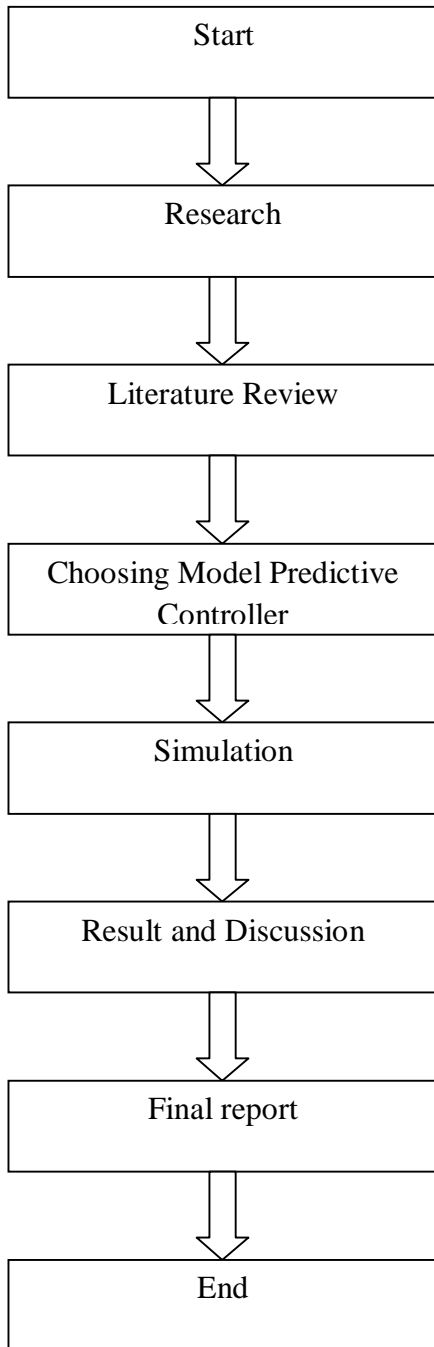


Figure 3: Project Activities Flow

Figure 4: MATLAB Simulation

The project is a simulation project. Specifically, it is a simulation of Model Predictive Controller on how the performance to system. First and for most, the project will begin with the research on several issues which had been mention in the research methodology above.

With the collective information, the project will proceed with the literature review that will mention about what people had done before for Model Predictive Controller strategies and the performance based on certain process. Besides, the author will discuss a basic knowledge of on how the Model Predictive Controller functioning and what are the benefits when using Model Predictive Controller in a control system or process.

After completing the literature review, the further studies will move on to tune the Model Predictive parameter of Controller approaches to make a performance comparison between them due to a process stated in the objective of the proposal. Besides, the author needs to identify the parameters that involved in measuring the performance of each Model Predictive Controller due to system tested such as the value  $T_i$ , input rate weight, output rate weight, prediction horizon and control horizon.

Then, the simulation of Model Predictive Controller will be done by using MATLAB. After completing the simulation, the result and discussion will be done to know the performance of each Model Predictive Controller based on the stated systems.

Lastly, all the studies and discussion will be compiled in the final report. Apart from that, the justification of performance of the Model Predictive Controller also will be mentioned in the discussion.



### 3.2 Research Methodology

Research is a method taken in order to gain information regarding the major scope of the project. The sources of the research cover the handbook of condensate stabilization unit, e-journal, e-thesis and several trusted link.

The steps of research:

1. Gain information of the Model Predictive Controller strategies and its performance.
2. List down the parameter used to be tested for inverse response process.
3. Make a comparison between the parameter tuned based on a control process decided.

#### Project Simulation

The project simulation will be done by using MATLAB. Then, we will make a comparison between parameter tuned based on a system that we already choose early in the topic.

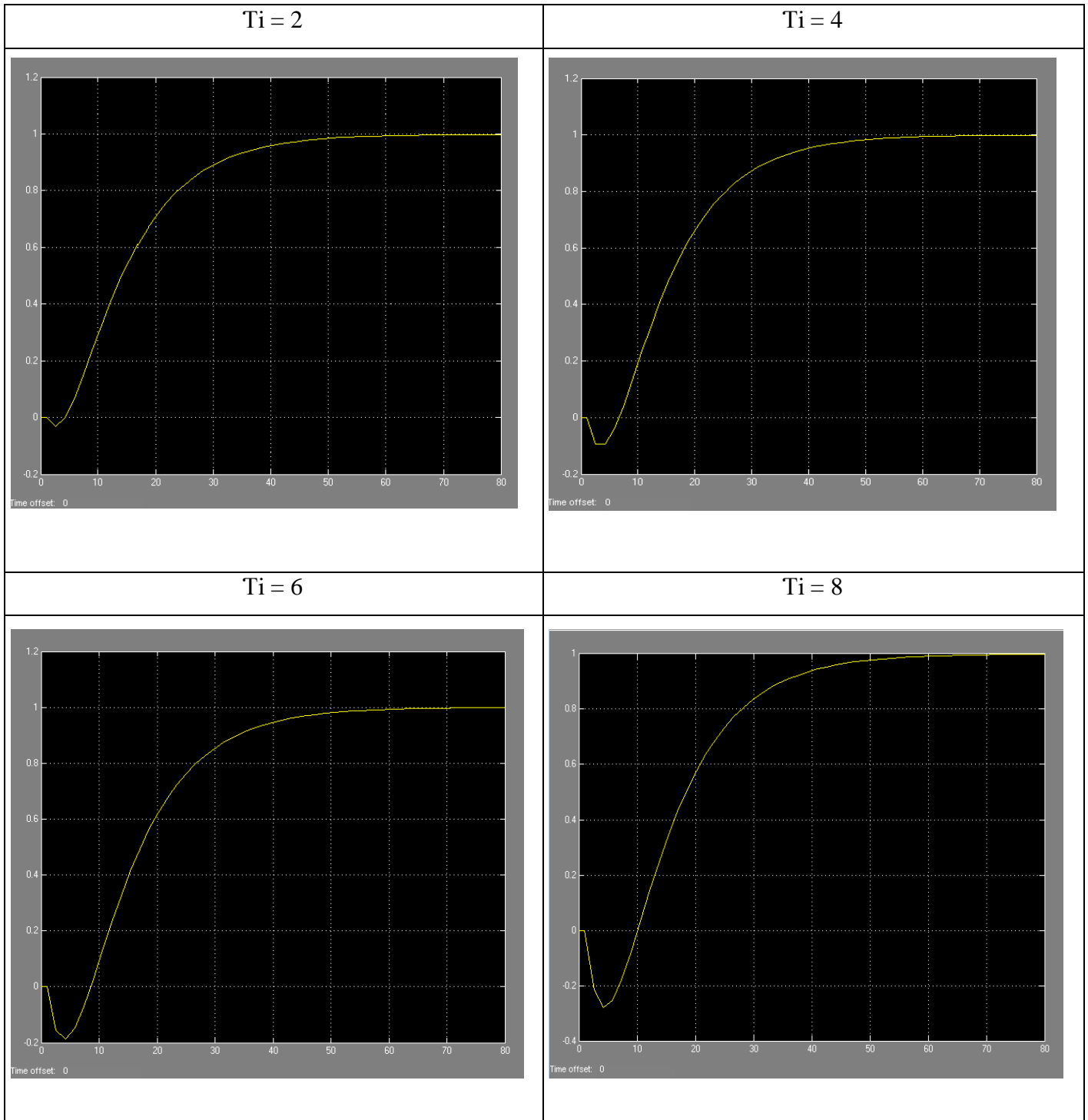
### 3.3 Activities/Gantt Chart and Milestone

No	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Project Work Continues								●							
2	Submission of Progress Report															
3	Project Work Continues															
4	Pre- EDX											●				
5	Submission of Draft Report												●			
6	Submission of Dissertation (soft bound)													●		
7	Submission of Technical Paper													●		
8	Oral Presentation														●	
9	Submission of Dissertation (hard bound)															●

# CHAPTER 4

## RESULT AND DISCUSSION

### THE EFFECT OF $T_i$ WITHOUT CONTROLLER



### Comparison between $T_i = 2, 4, 6,$ and $8$

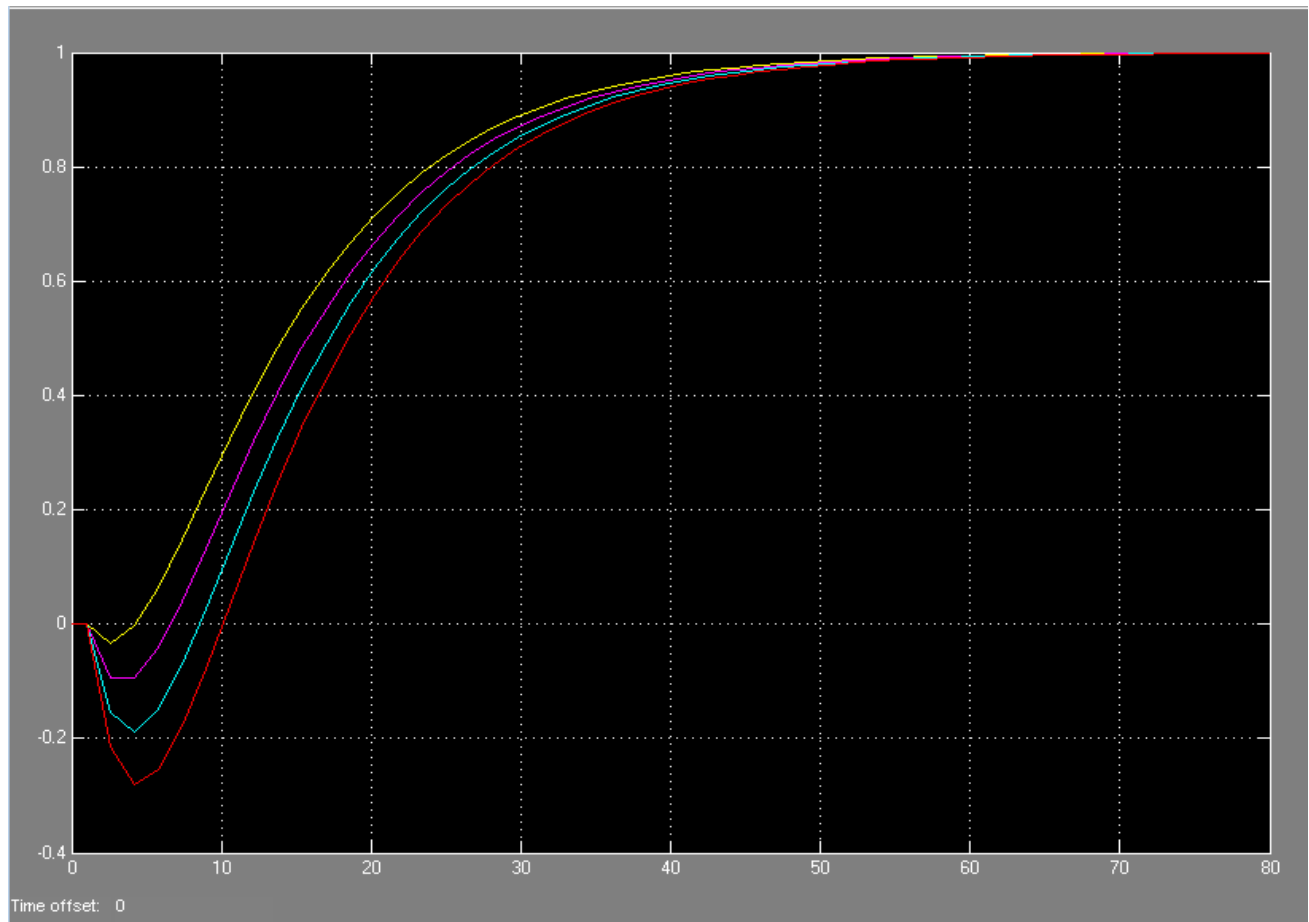


Figure 5: Effect of  $T_i$

Overall process for  $T_i = 2, 4, 6, \text{ and } 8$

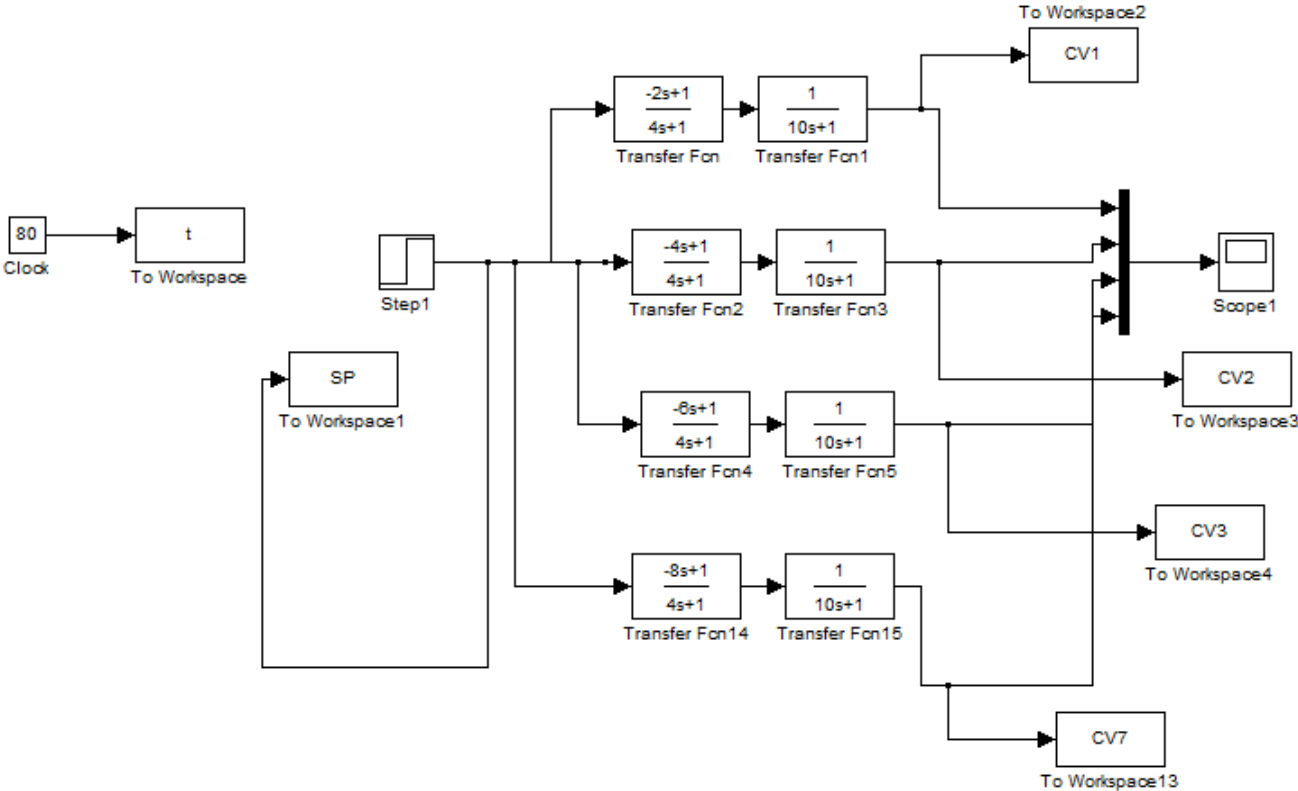


Figure 6: Block Diagram

Error for  $T_i = 2, 4, 6 \text{ and } 8$

CV	$T_i$	Input variation (Error)	Output Variation (Error)
CV1	2	1.0672	15.982
CV2	4	1.1896	17.9701
CV3	6	1.374	19.9583
CV4	8	1.5585	21.9464

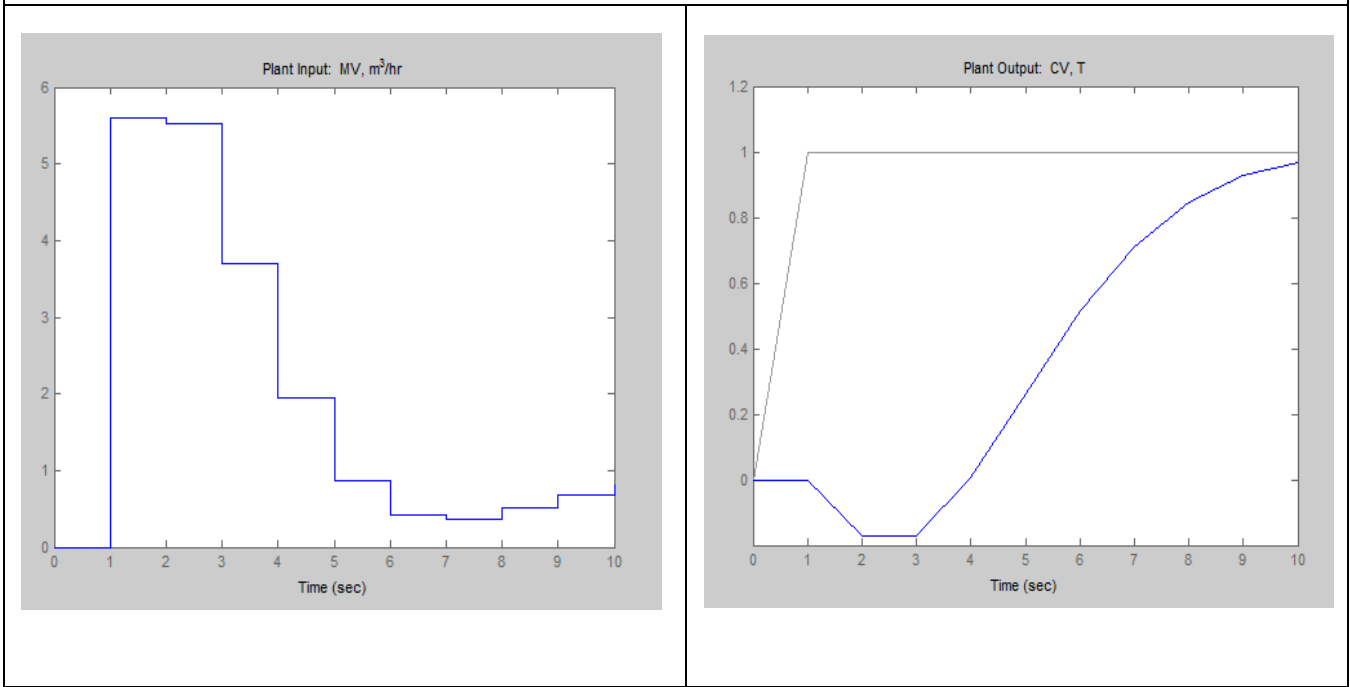
Table 2: Error of different  $T_i$

## DISCUSSION FOR THE EFFECT OF $T_i$ WITHOUT CONTROLLER

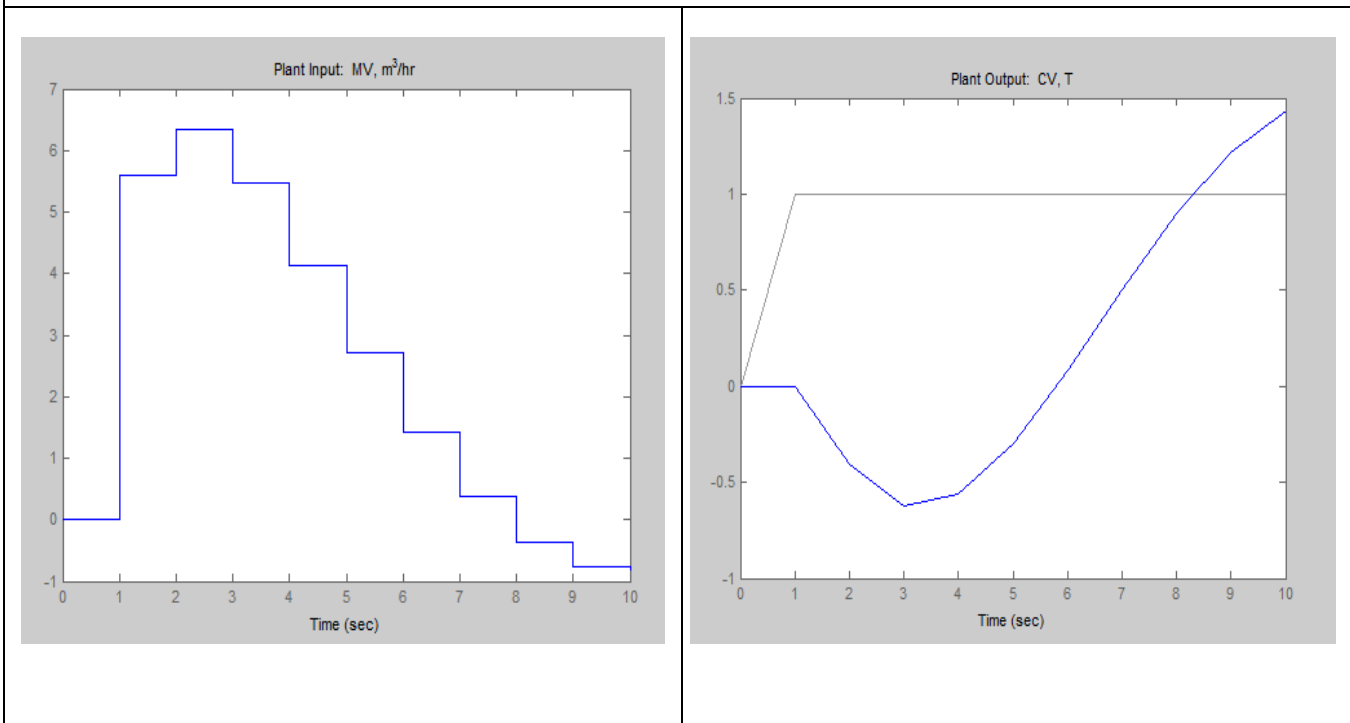
For the inverse response of second order process with different value of  $T_i$ , we can see the behavior of each graph changed with step value of 1. When we increase the value of  $T_i$ , the inverse response become lower compared to smaller value of  $T_i$ . By using MATLAB software, we calculate the error without using any controller in the process and make comparison between them. Based on the graph and data calculated, we can say that lower value of  $T_i$  is better and more efficiency to the process because it gives lower value of error for the system although without using any model controller.

# THE EFFECT OF $T_i$ WITH DEFAULT MPC SETTING

$T_i = 2$



$T_i = 4$



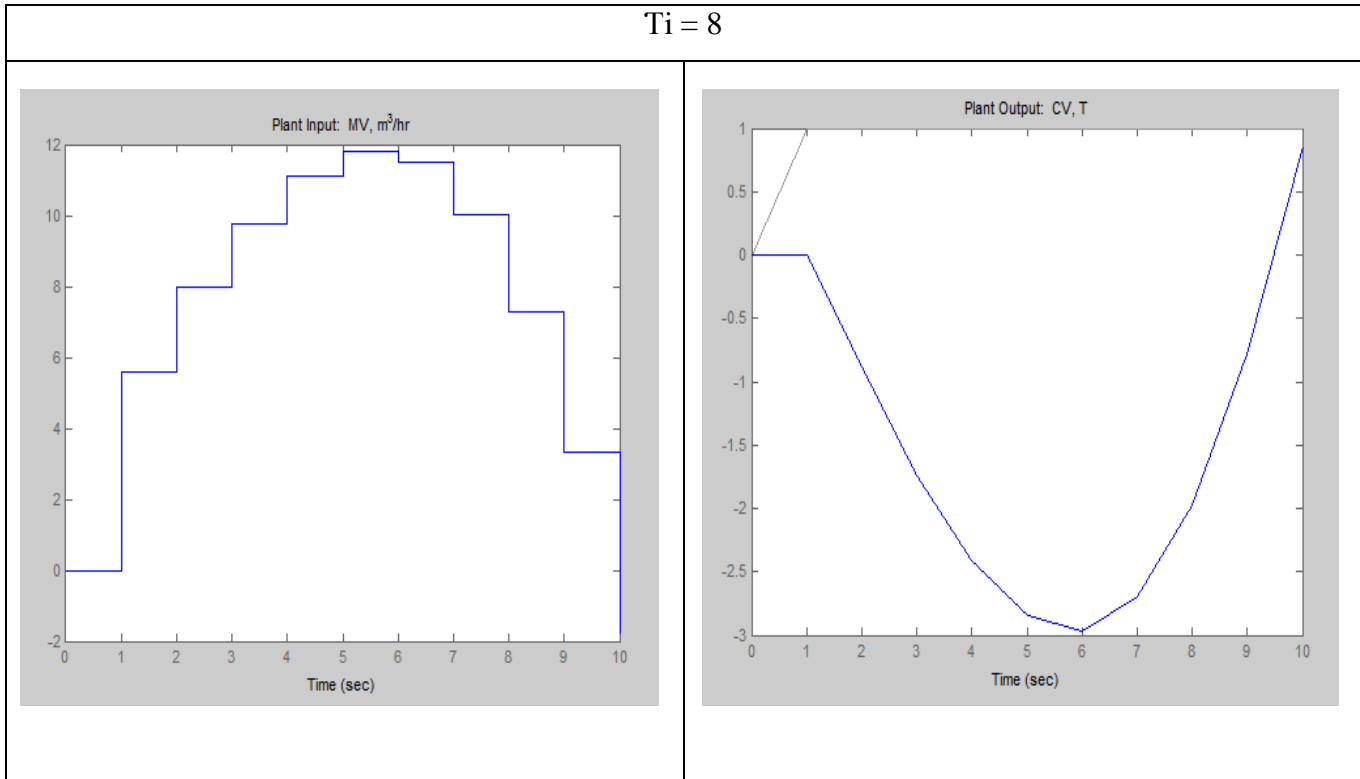
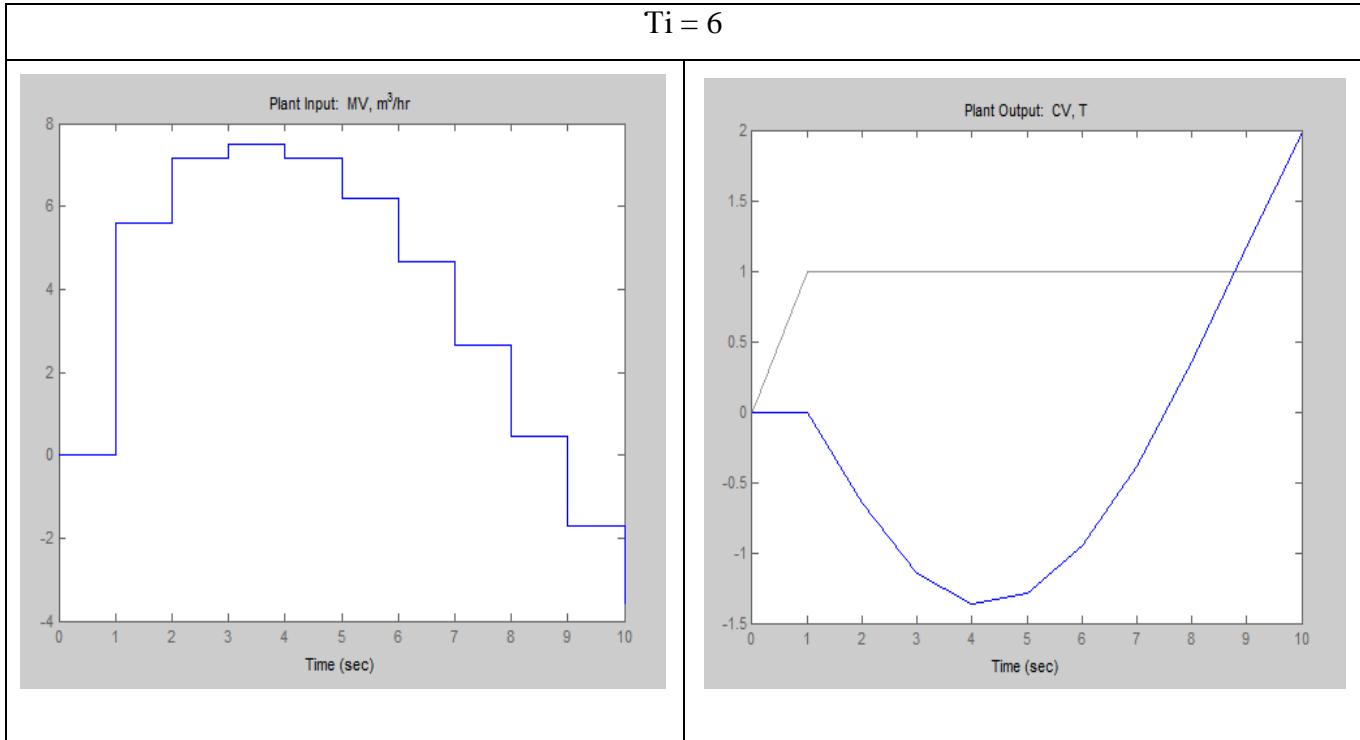


Figure 7: Effect of  $T_i$  with MPC



DATA AND ERROR VALUE FOR THE EFFECT OF  $T_i$  WITH DEFAULT MPC  
SETTING

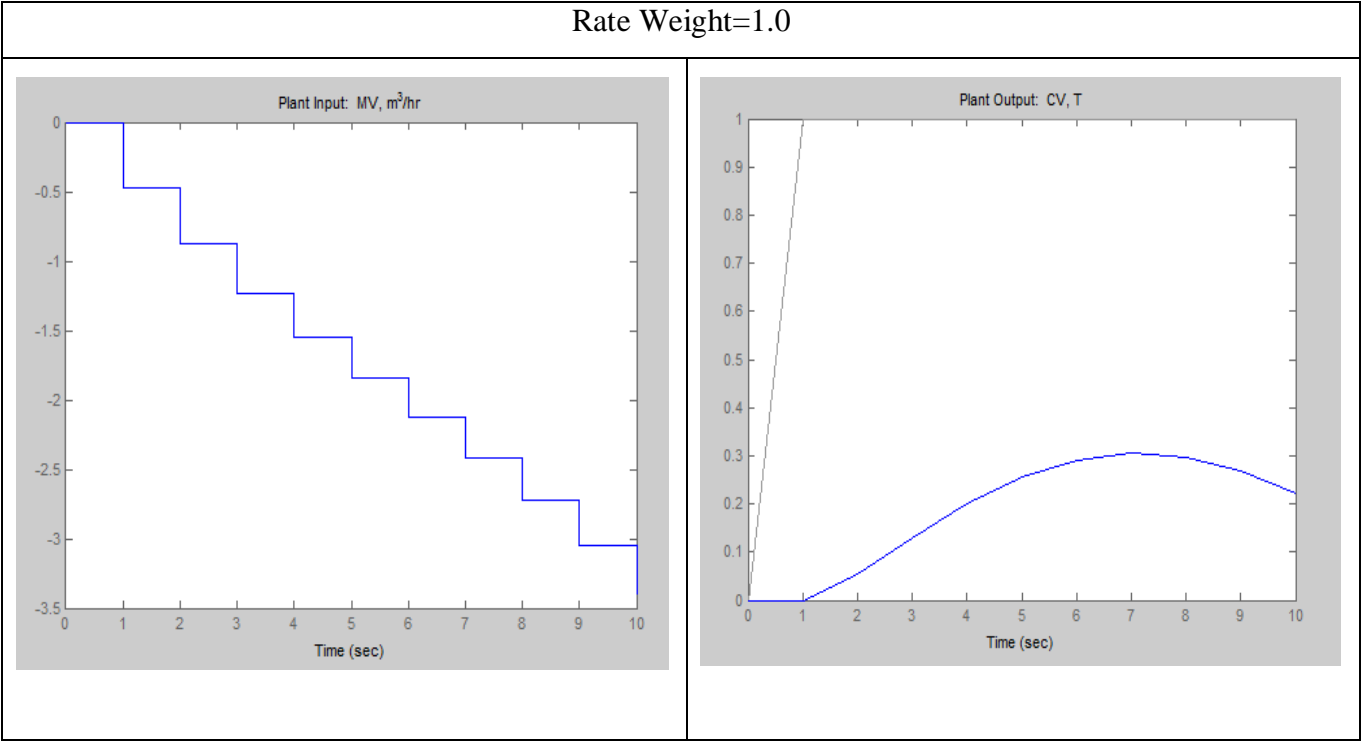
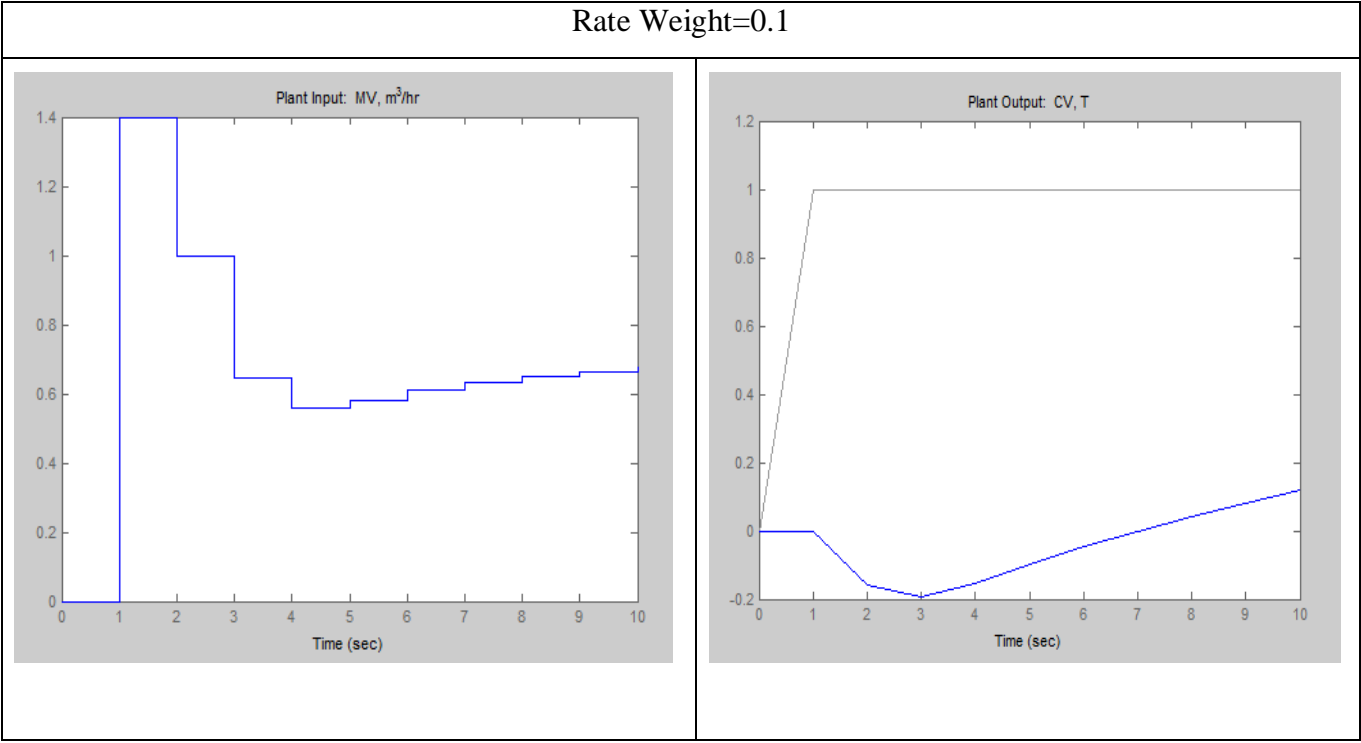
$t_i=2$			$t_i=4$			$t_i=6$			$t_i=8$	
5.6			0			0			0	
5.52	0.08		5.6	5.6		1.4	1.4		1.4	1.4
3.69	1.83		6.34	0.74		0.998	0.402		1.05	0.35
1.95	1.74		5.47	0.87		0.646	0.352		0.722	0.328
0.875	1.075		4.14	1.33		0.558	0.088		0.628	0.094
0.422	0.453		2.72	1.42		0.579	0.021		0.634	0.006
0.371	0.051		1.43	1.29		0.612	0.033		0.657	0.023
0.505	0.134		0.371	1.059		0.634	0.022		0.676	0.019
0.679	0.174		-0.366	0.737		0.651	0.017		0.706	0.03
Total	<b>5.537</b>		-0.762	0.396		0.665	0.014		0.692	0.014
			Total	<b>13.442</b>		Total	<b>2.349</b>		Total	<b>2.264</b>
<b>IAE</b>	<b>6.5922</b>		<b>IAE</b>	<b>9.3456</b>		<b>IAE</b>	<b>10.4738</b>		<b>IAE</b>	<b>11.1559</b>
ISE	6.125		ISE	11.4942		ISE	11.0621		ISE	12.5704
ITAE	21.5067		ITAE	35.3649		ITAE	50.6487		ITAE	54.489

Table 3: Error with default MPC setting with  $T_i$  tuning

## DISCUSSION FOR THE EFFECT OF $T_i$ WITH DEFAULT MPC SETTING

For this controller tuning, we set all parameter in default setting such as input rate weight=0.1, output rate weight=1.0, prediction horizon=10 and control horizon=2. We used  $T_i=2, 4, 6$  and 8 for this tuning to see the behavior of the graph after it been simulated and make comparison between the process without using any controller. After the simulation was run, the error value of  $T_i=2$  using MPC controller is lower that is 6.5922 compare to the process without controller which is 15.982. The different between these two errors is 9.3898. it is same goes to  $T_i=4, 6,$  and 8. The error becomes lower because the MPC can predict the past input or output to become a predicted output and come out with future error. Then, the future error undergoes optimization process to reduce the cost function and constraint and finally come out with future input which will send back to the controller. Then, the MPC will send the corrective signal to the set point to minimize the error. Thus, for second order process it is better to use small number of  $T_i$  in the transfer function.

EFFECT OF MPC INPUT RATE WEIGHT= 0.1, 1, 10 TUNING FOR  $T_i = 6$ , OTHERS ARE DEFAULT.



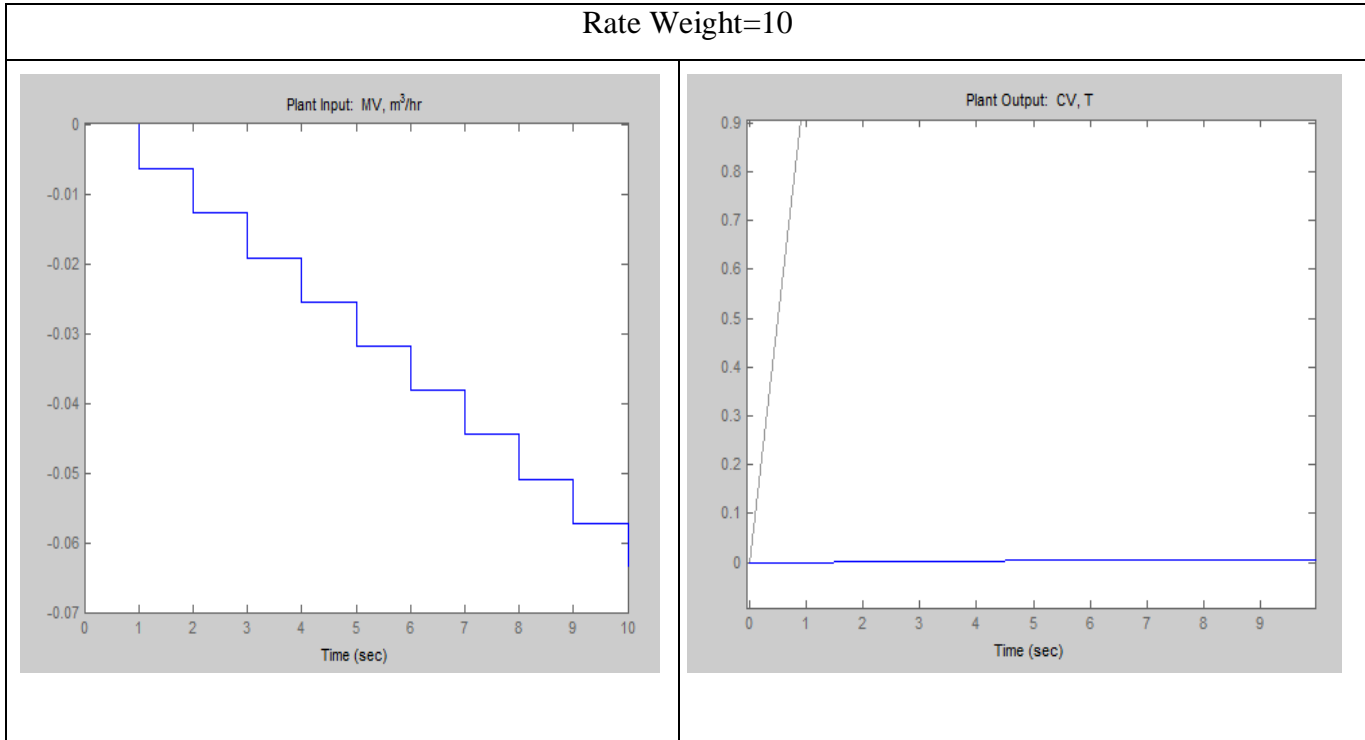


Figure 8: Graph with Input rate Weight Tuning

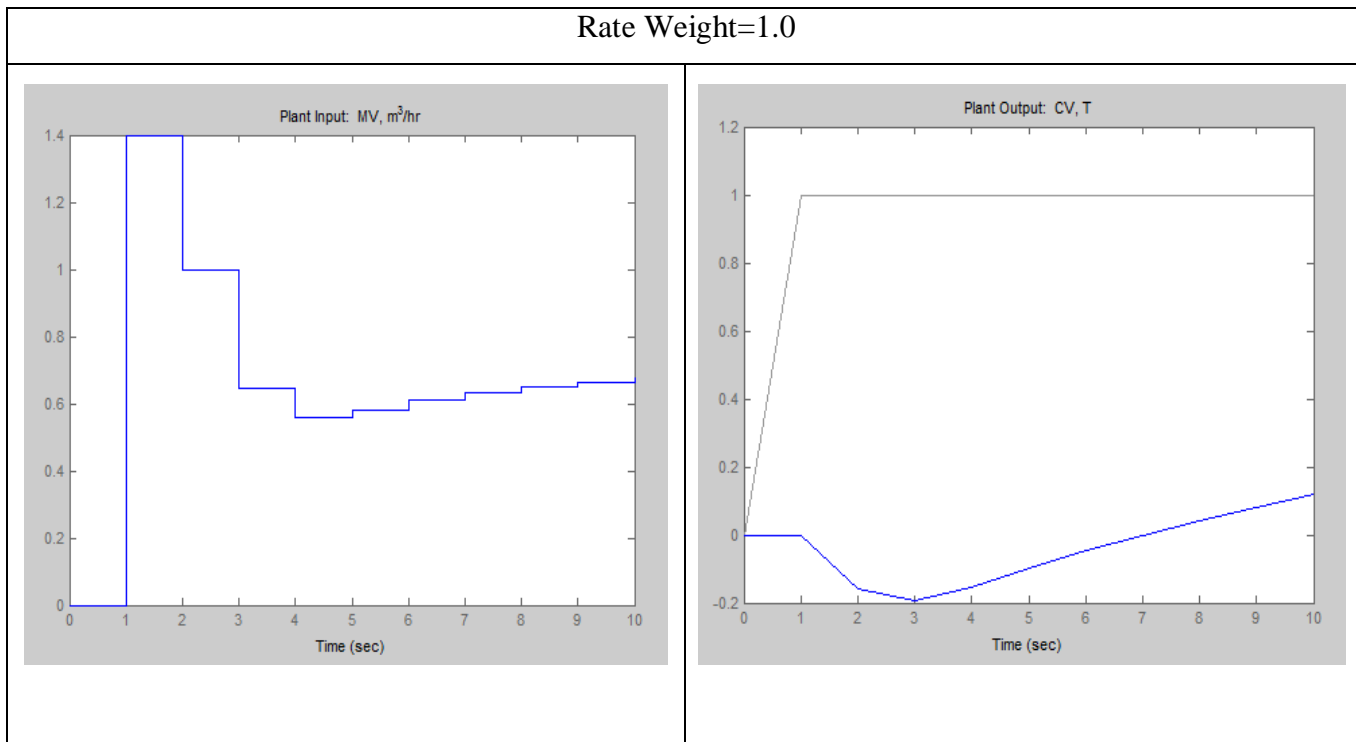
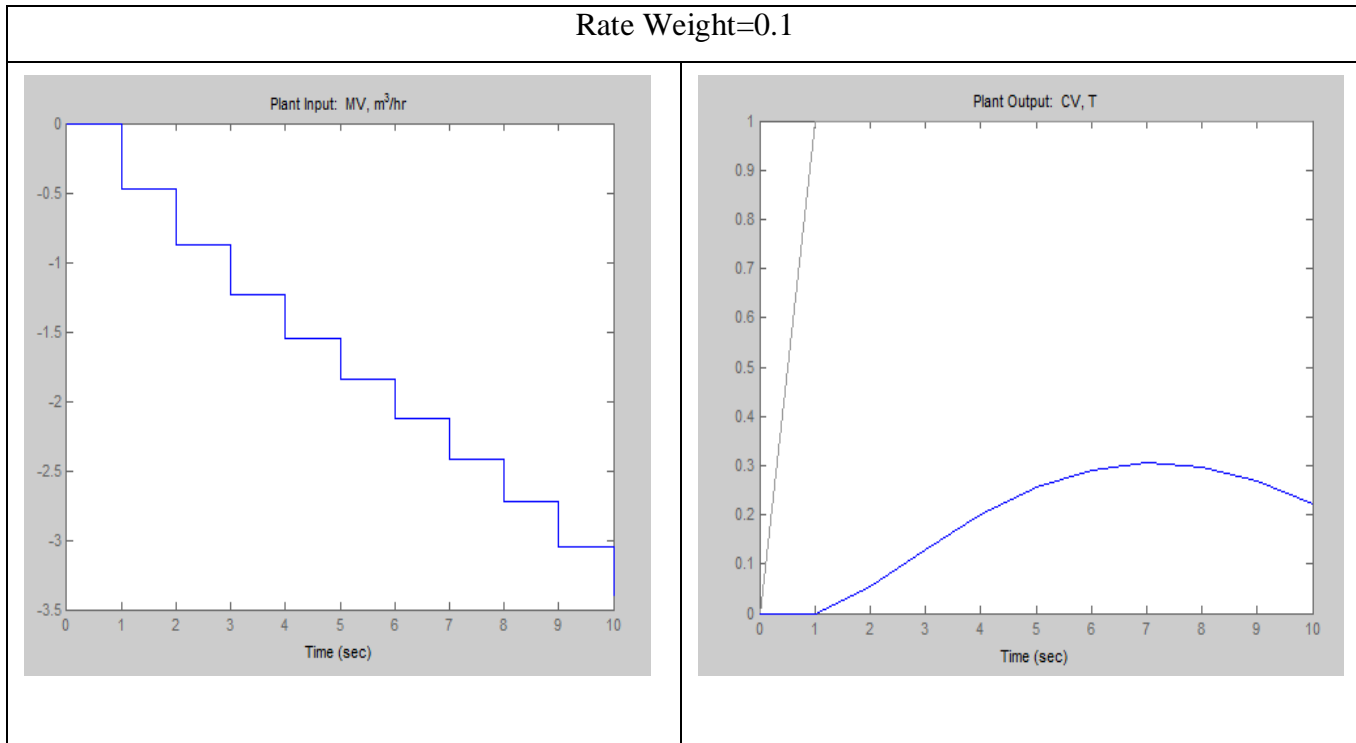
weight=0.1			weight=1			weight=10	
0			0			0	
1.4	1.4		-0.471	0.471		-0.00639	0.00639
0.998	0.402		-0.876	0.405		-0.0128	0.00641
0.646	0.352		-1.23	0.354		-0.0191	0.0063
0.558	0.088		-1.54	0.31		-0.0255	0.0064
0.579	0.021		-1.84	0.3		-0.0319	0.0064
0.612	0.033		-2.13	0.29		-0.0382	0.0063
0.634	0.022		-2.42	0.29		-0.0446	0.0064
0.651	0.017		-2.72	0.3		-0.0509	0.0063
0.665	0.014		-3.05	0.33		-0.0573	0.0064
Total	<b>2.349</b>		-3.41	0.36		-0.0636	0.0063
			Total	<b>3.41</b>		Total	<b>0.0636</b>
<b>IAE</b>	<b>10.4738</b>		<b>IAE</b>	<b>8.092</b>		<b>IAE</b>	<b>9.9656</b>
ISE	11.0621		ISE	6.6703		ISE	9.9314
ITAE	50.6487		ITAE	37.6566		ITAE	49.7679

Table 4: Error with Input Rate Weight Tuning

DISCUSSION FOR THE EFFECT OF MPC INPUT RATE WEIGHT= 0.1, 1, 10  
TUNING FOR  $T_i = 6$ , OTHERS ARE DEFAULT.

For this tuning of input rate weight, we only use  $T_i=6$  with different value of input rate weight which are 0.1, 1.0 and 10. The default value of input rate weight is 0.1. Based on the graph above, the increment of input rate weight value from 0.1 to 1.0 decreases the error value for the controlled variable and decrease the move sizes significantly, especially the initial move. The controller also minimizes the weighted sum of manipulated variable deviations from their nominal values. But for manipulated variable, the error is increase a bit only when the input rate weight is increase, then it decreases to 0.0636 when we increase the weight rate to 10. The value of input rate weight is affected to both manipulated and controlled variable error. So it is important to tune the input rate weight to ensure that the error is the least for manipulated and controlled variables respectively.

EFFECT OF MPC OUTPUT RATE WEIGHT= 0.1, 1, 10 TUNING FOR  $T_i = 6$ ,  
OTHERS ARE DEFAULT.



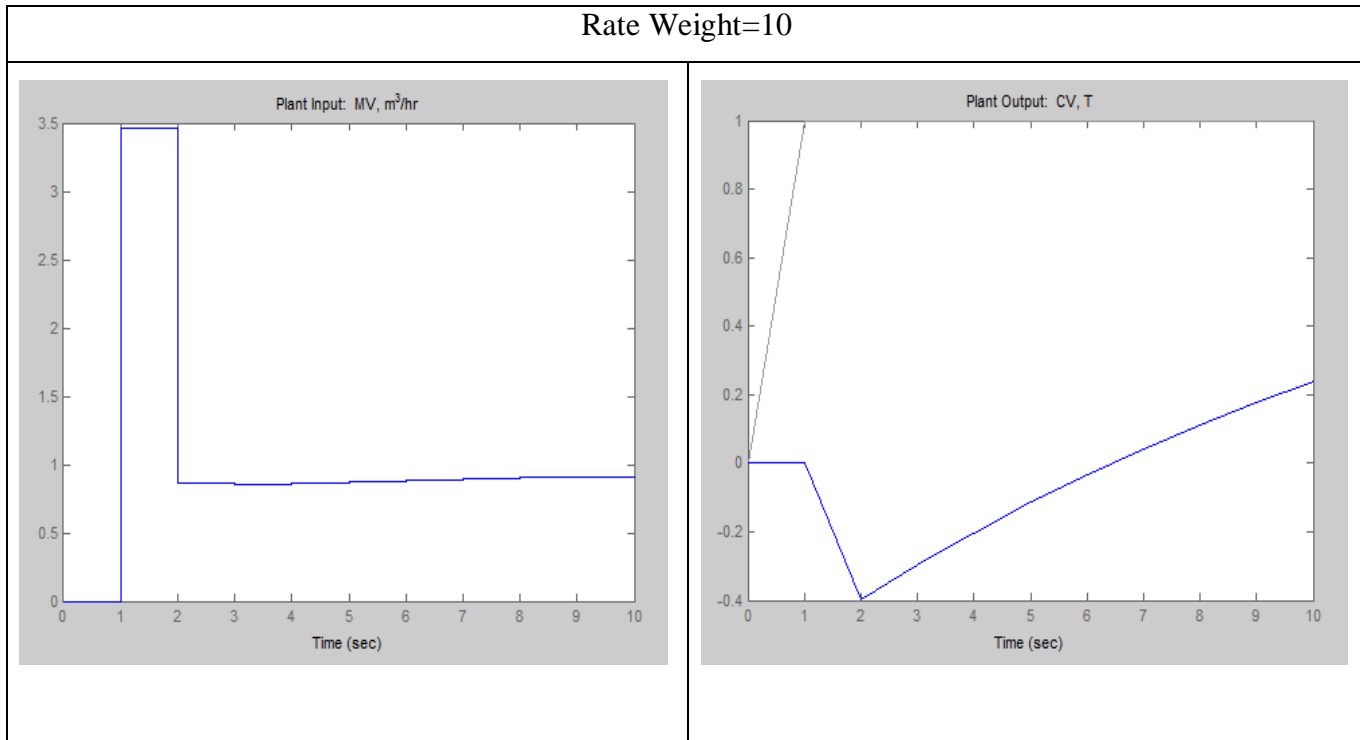


Figure 9: Graph with Output Rate Weight Tuning

output rate weight							
weight=0.1			weight=1			weight=10	
0			0			0	
-0.471	0.471		1.4	1.4		3.47	3.47
-0.876	0.405		0.998	0.402		0.867	2.603
-1.23	0.354		0.646	0.352		0.852	0.015
-1.54	0.31		0.558	0.088		0.868	0.016
-1.84	0.3		0.579	0.021		0.878	0.01
-2.13	0.29		0.612	0.033		0.887	0.009
-2.42	0.29		0.634	0.022		0.895	0.008
-2.72	0.3		0.651	0.017		0.903	0.008
-3.05	0.33		0.665	0.014		0.91	0.007
-3.41	0.36		Total	<b>2.349</b>		Total	<b>6.146</b>
Total	<b>3.41</b>						



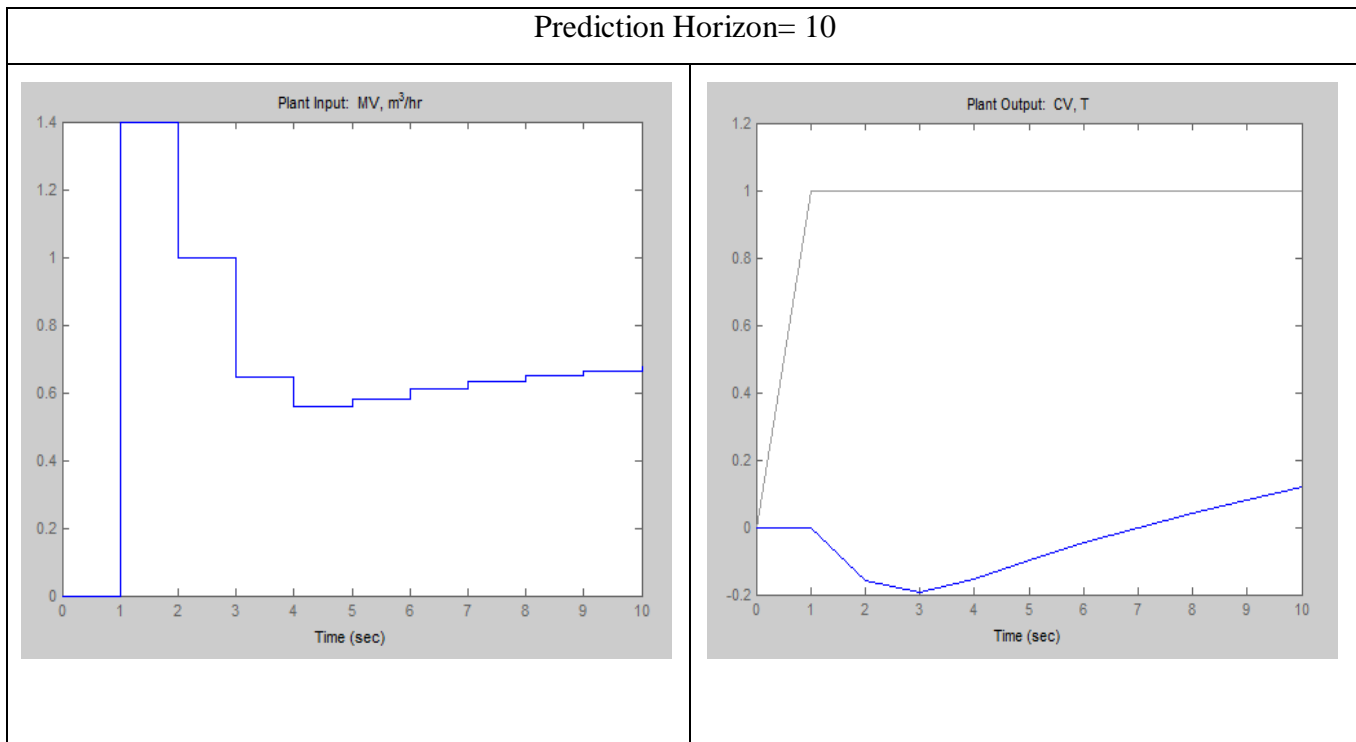
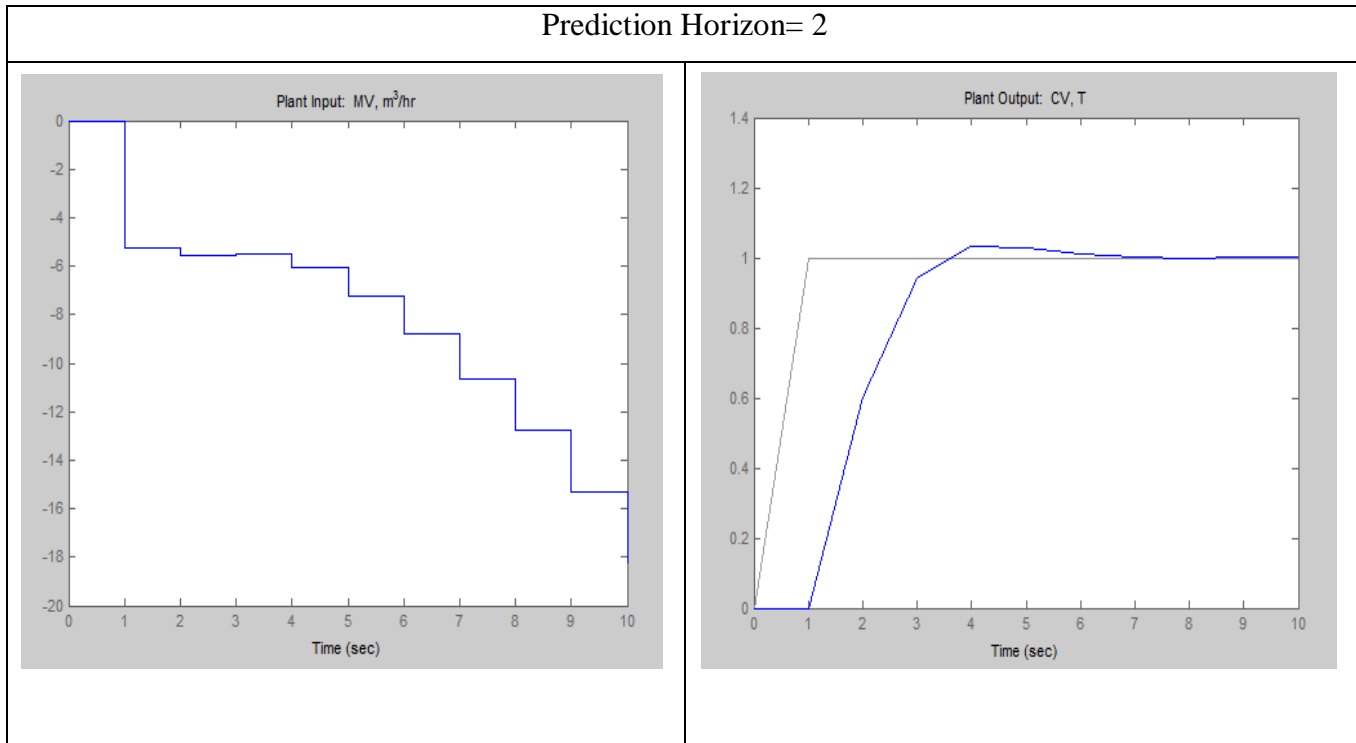
<b>IAE</b>	<b>8.092</b>		<b>IAE</b>	<b>10.4738</b>		<b>IAE</b>	<b>10.5936</b>
ISE	6.6703		ISE	11.0621		ISE	11.5651
ITAE	37.6566		ITAE	50.6487		ITAE	49.2717

Table 5: Error with Output Rate Tuning

DISCUSSION FOR THE EFFECT OF MPC OUTPUT RATE WEIGHT= 0.1, 1, 10  
TUNING FOR  $T_i = 6$ , OTHERS ARE DEFAULT.

For this tuning of input rate weight, we only use  $T_i=6$  with different value of output rate weight which are 0.1, 1.0 and 10. The default value of output rate weight is 1. The output weights let you dictate the accuracy with which each output must track its setpoint. Specifically, the controller predicts deviations for each output over the prediction horizon which we set it as default, 10. It multiplies each deviation by the output's weight value. The weights must be zero or positive. A large weight on a particular output causes the controller to minimize deviations in that output. Based on the graph above, the increment of output rate weight value from 0.1 to 1.0 decreases the error value for the controlled variable and decrease the move sizes significantly, especially the initial move. The controller also minimizes the weighted sum of manipulated variable deviations from their nominal values. But for manipulated variable, the error is decrease a bit only when the input rate weight is increase, then it increases to 6.146 when we increase the weight rate to 10. The value of output rate weight is affected to both manipulated and controlled variable error. So it is important to tune the input rate weight to ensure that the error is the least for manipulated and controlled variables respectively.

EFFECT OF MPC PREDICTION HORIZON= 2, 10, 50 TUNING FOR  $T_i = 6$ ,  
OTHERS ARE DEFAULT.



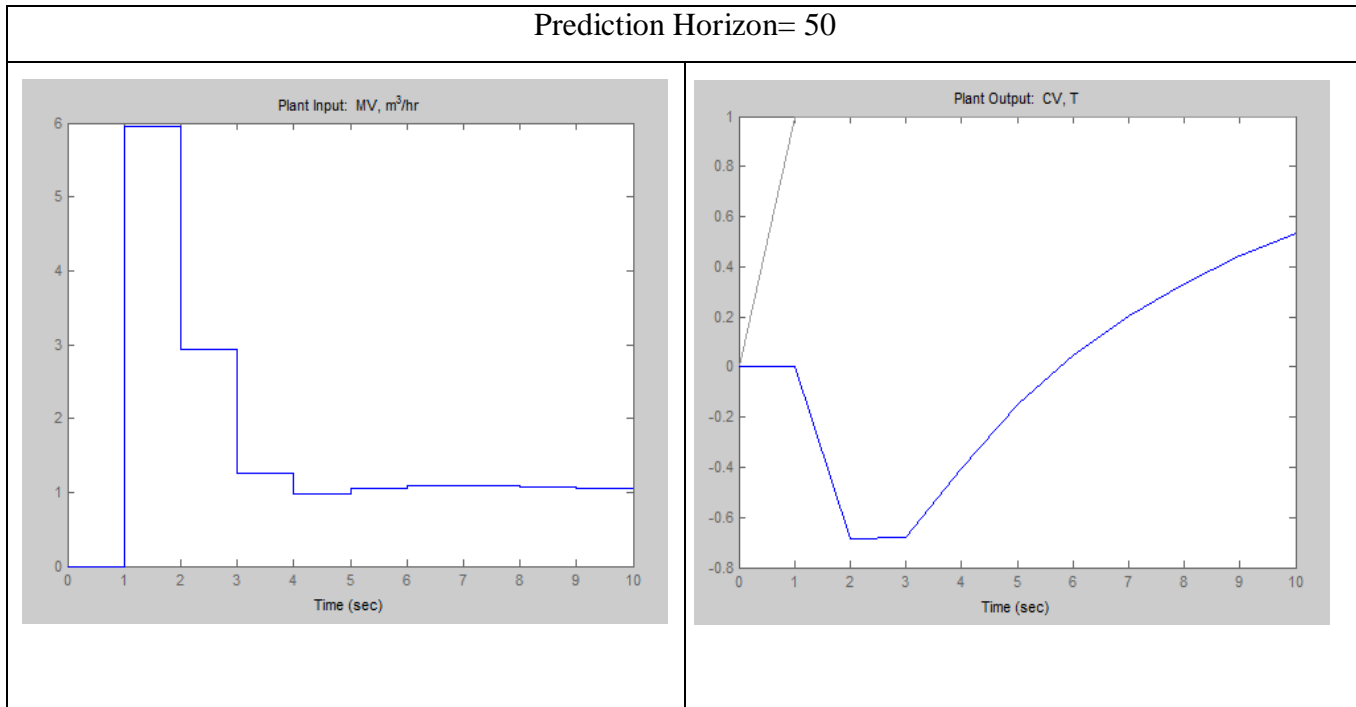


Figure 10: Graph with Prediction Horizon Tuning

PH=2			PH=10			PH=50	
0			0			0	
-5.25	5.25		1.4	1.4		5.95	5.95
-5.58	0.33		0.998	0.402		2.93	3.02
-5.48	0.1		0.646	0.352		1.25	1.68
-6.07	0.59		0.558	0.088		0.982	0.268
-7.25	1.18		0.579	0.021		1.05	0.068
-8.8	1.55		0.612	0.033		1.09	0.04
-10.6	1.8		0.634	0.022		1.08	0.01
-12.8	2.2		0.651	0.017		1.07	0.01
-15.3	2.5		0.665	0.014		1.06	0.01
-18.3	3		Total	<b>2.349</b>		Total	<b>11.056</b>
Total	<b>18.5</b>						

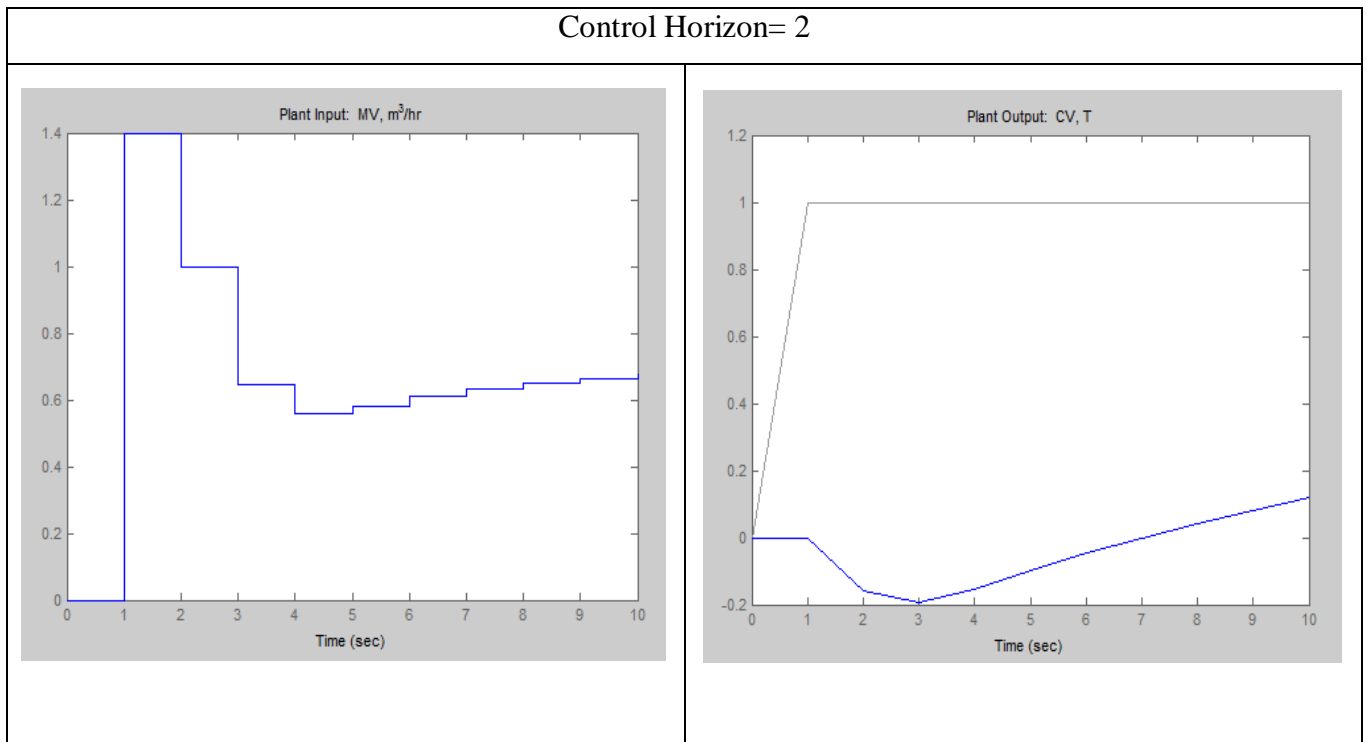
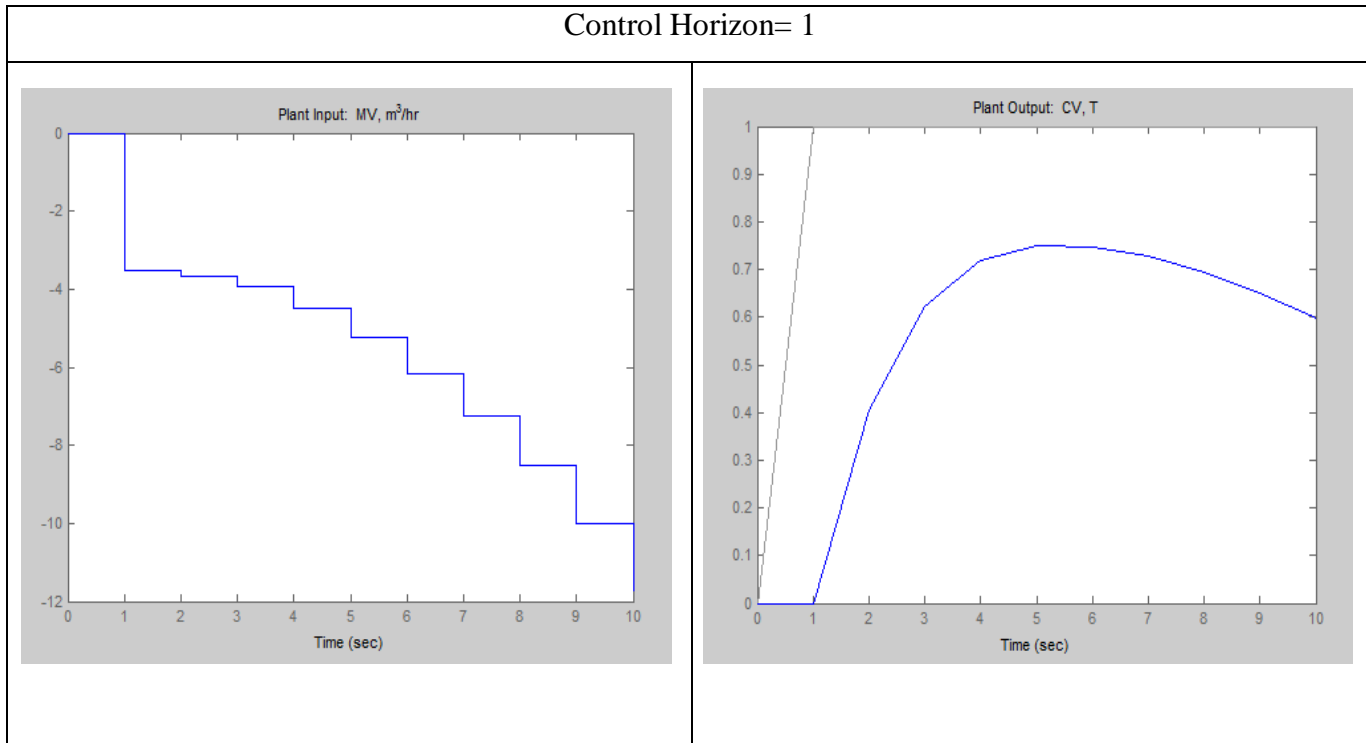
<b>IAE</b>	<b>2.0298</b>		<b>IAE</b>	<b>10.4738</b>		<b>IAE</b>	<b>10.6223</b>
ISE	1.6624		ISE	11.0621		ISE	12.8529
ITAE	2.331		ITAE	50.6487		ITAE	44.7367

Table 6: Error with Prediction Horizon Tuning

DISCUSSION FOR THE EFFECT OF MPC PREDICTION HORIZON= 2, 10, 50  
TUNING FOR  $T_i = 6$ , OTHERS ARE DEFAULT.

Prediction horizon is the number of control intervals over which the outputs are to be optimized. Based on the data calculated, the error for the changes of prediction horizon is not in sequence, means it is not increasing or decreasing based on the prediction horizon number. When we change the prediction horizon from 2 to 10, the input error is decreasing too much but the output error is increasing from 2.0298 to 10.4738. After that when the prediction horizon is increase to 50, the input error is increasing and the output error is increasing a bit only. So it is important to tune the prediction horizon number to ensure that the error is the least for manipulated and controlled variables respectively. Different tuning will give different behavior of input and output graph which may affect the efficiency and time taken for the process to reach the set point.

EFFECT OF MPC CONTROL HORIZON= 1, 5, 10 TUNING FOR  $T_i = 6$ , OTHERS ARE DEFAULT.



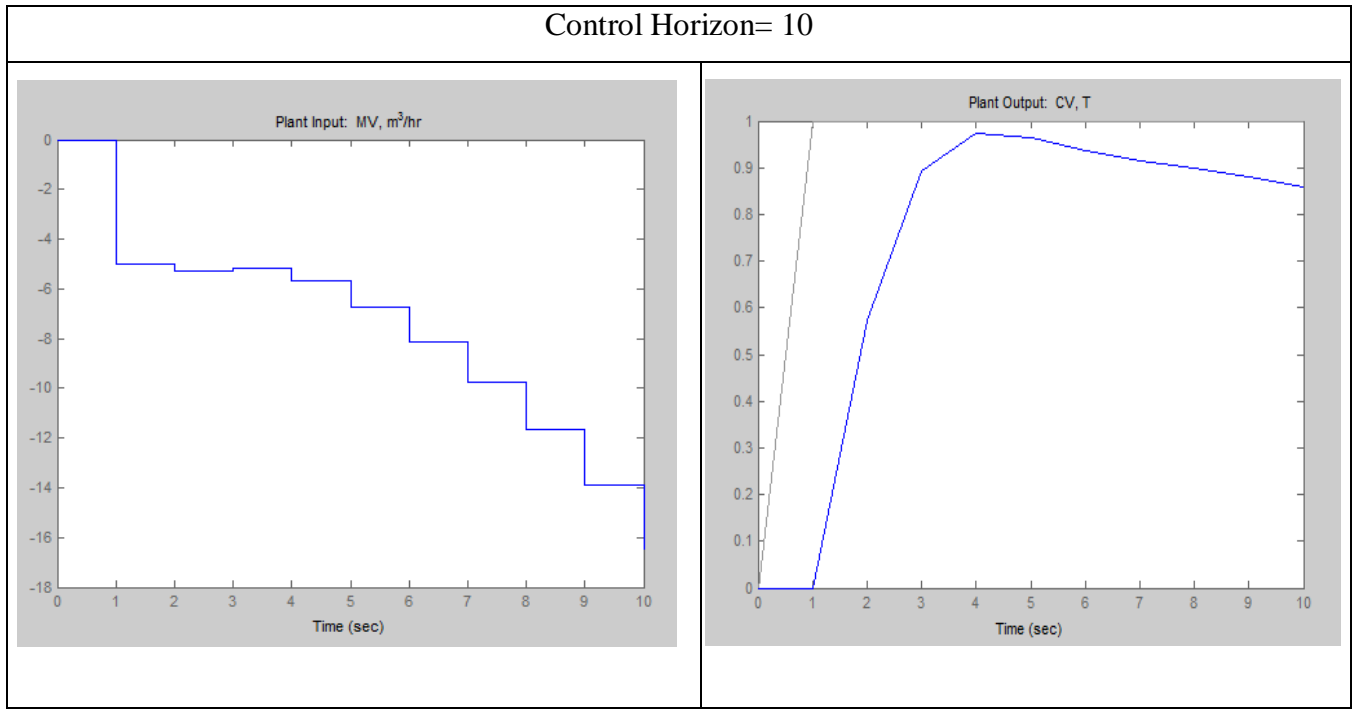


Figure 11: Graph with Control Horizon Tuning

CH=1			CH=5			CH=10	
0			0			0	
-3.54	3.54		-3.29	3.29		-5	5
-3.65	0.11		-3.27	0.02		-5.27	0.27
-3.94	0.29		-2.97	0.3		-5.16	0.11
-4.48	0.54		-3.06	0.09		-5.68	0.52
-5.23	0.75		-3.45	0.39		-6.73	1.05
-6.15	0.92		-3.98	0.53		-8.12	1.39
-7.25	1.1		-4.58	0.6		-9.75	1.63
-8.53	1.28		-5.24	0.66		-11.7	1.95
-10	1.47		-5.96	0.72		-13.9	2.2
-11.7	1.7		-6.77	0.81		-16.5	2.6
<b>Total</b>	<b>11.7</b>		<b>Total</b>	<b>7.41</b>		<b>Total</b>	<b>16.72</b>

<b>IAE</b>	<b>4.3855</b>		<b>IAE</b>	<b>6.4548</b>		<b>IAE</b>	<b>2.5361</b>
ISE	2.5723		ISE	4.5287		ISE	1.7408
ITAE	16.7406		ITAE	31.5615		ITAE	6.0361

Table 7: Error with Control Horizon Tuning

DISCUSSION FOR THE EFFECT OF MPC CONTROL HORIZON= 1, 5, 10 TUNING FOR  $T_i = 6$ , OTHERS ARE DEFAULT.

Control horizon sets the number of control intervals over which the manipulated variables are to be optimized. Based on the data calculated, the error for the changes of control horizon is not in sequence, means it is not increasing or decreasing based on the control horizon number inserted. When we change the control horizon from 1 to 5, the input error is decreasing too much but the output error is increasing from 4.3855 to 6.4548. After that when the control horizon is increase to 10, the input error is increasing and the output error is decreasing to 2.5361. So it is important to tune the control horizon number to ensure that the error is the least for manipulated and controlled variables respectively. Different tuning will give different behavior of input and output graph which may affect the efficiency and time taken for the process to reach the set point.

Feedback Controller.

Feed forward controllers also exist but are more complicated to implement. Here we will describe the use of feedback controllers.

The purpose of a servomechanism is to provide one or more of the following objectives:

- I. accurate control of motion without the need for human attendants (automatic control)
- II. maintenance of accuracy with mechanical load variations, changes in the environment, power supply fluctuations, and aging and deterioration of components (regulation and self-calibration)
- III. control of a high-power load from a low-power command signal (power amplification)
- IV. control of an output from a remotely located input, without the use of mechanical linkages (remote control, shaft repeater).

In feedback control the variable required to be controlled is measured. This measurement is compared with a given set-point. The controller takes this error and decides what action should be taken by the manipulated variable to compensate for and hence remove the error.

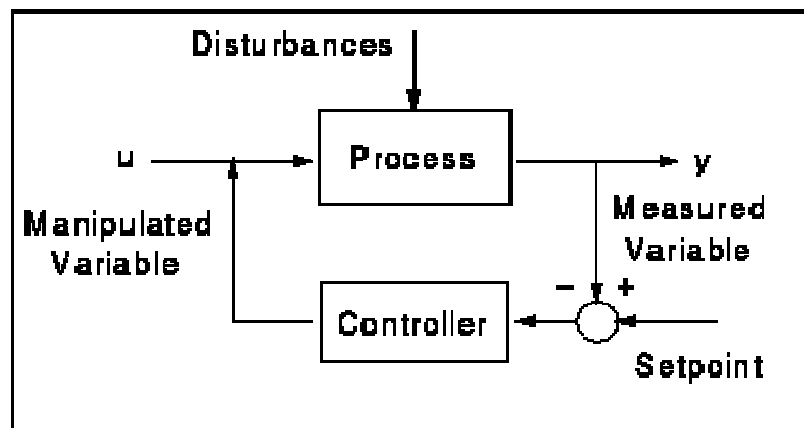


Figure 12: Feedback Control Loop



The advantage of this type of control is that it is simple to implement. Not only does the feedback control system require no knowledge of the source or nature of the disturbances, but it also requires minimal detailed information about how the process itself works. Feedback control action is entirely empirical. So long as an adjustment is being made in the correct sense then the control system should remove the effect of an external disturbance.

A servo control loop is one which responds to a change in setpoint. The setpoint may be changed as a function of time (typical of this are batch processes), and therefore the controlled variable must follow the setpoint.

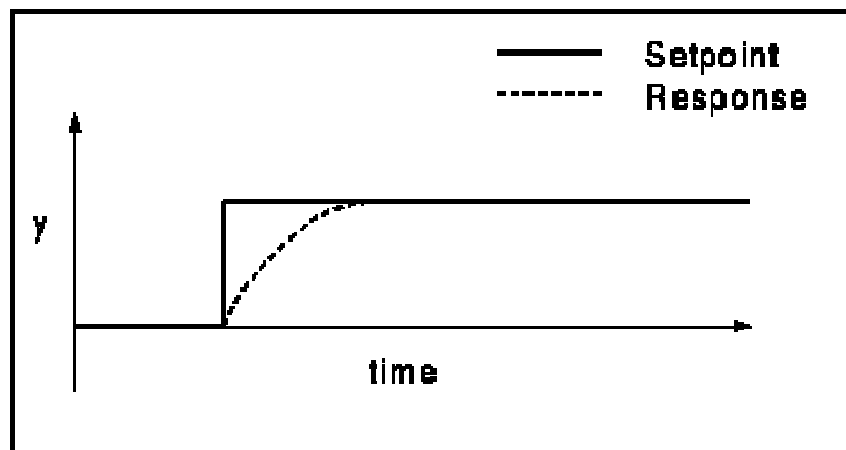
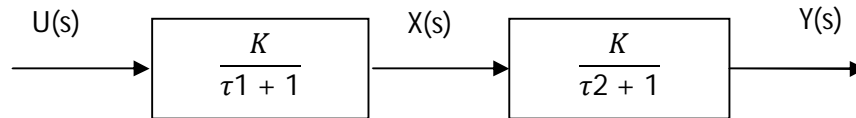


Figure 13: Servo Control

Response of second order process

Second order transfer function can arise physically whenever two first-order processes are connected in series. For example, two stirred-tank blending processes, each with a first-order transfer function relating inlet to outlet mass fraction, might be physically connected so the outflow stream of the first tank is used as the inflow stream of the second tank. Figure below illustrates the signal flow relation for such a process. Here

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K_1 K_2}{(\tau_1 s + 1)(\tau_2 s + 1)} = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$$



### Calculating the Error

Controller tuning have been developed that optimize the response for a simple process model and set point change. The optimum settings minimize an integral error criterion. Three popular integral error criteria are:

1. Integral of the absolute value of the error (IAE)

$$IAE = \int_0^{\infty} |e(t)| dt$$

Where the error signal  $e(t)$  is the difference between the set point and the measurement.

2. Integral of the squared error (ISE)

$$ISE = \int_0^{\infty} e(t)^2 dt$$

3. Integral of the time-weighted absolute error (ITAE)

$$ITAE = \int_0^{\infty} t|e(t)| dt$$

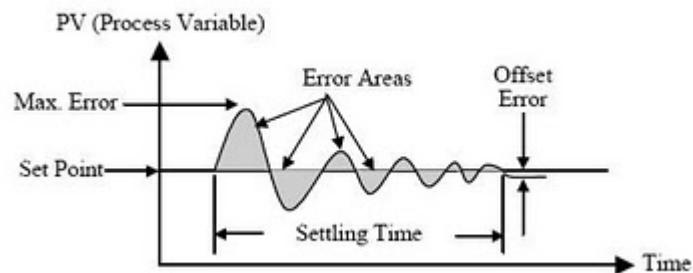


Table 14: Graphical interpretation of IAE.

The ISE criterion penalizes large errors, while the ITAE criterion penalizes errors that persist for long periods of time. In general, the ITAE is the preferred criterion because it usually results in the most conservative controller settings. By contrast, the ISE criterion provides the most aggressive settings, while the IAE criterion tends to produce controller settings that are between those for the ITAE and ISE criterion.

## **CHAPTER 5**

### **CONCLUSION**

In conclusion, this project is to analyze the behavior of Model Predictive controller for parameter tuning due to a system of inverse response control process. As the result of the behavior measurement, we should have a basic knowledge on how to tuning the controller for inverse response process. This is because inverse response is difficult to control and the tuning may affect either the input or output error measurement. Besides, by using Model Predictive controller is better than that not using any controller for any process to minimize the error and increase the efficiency of the process.

## REFERENCES

- [1] HesamKomariAlaei, KarimSalahshoor, HamedKomariAlaei, 2010, *Model Predictive Control of Distillation Column bases Recursive Parameter Estimation Method Using HYSYS Simulation*. Page 308.
- [2] Adrian Wills, March 2003, Doctor of Philosophy, *Barrier Function Based Model Predictive Control*.
- [3] Charles R. Nippert, *Simple Models That Illustrate Dynamic Matrix Control*
- [4] Sachin C. Patwardhan, *Workshop on Model Predictive Control*.
- [5] RashmiRanjankar, AngshumanKalita, *Dynamic Matrix Control*
- [6] D.T. Lemma, M. Ramasamy and M. Shuhaimi, 2010, *System Identification using Orthonormal Basis Filters*.Pages 2517.
- [7] JieZeng, Raymond de Callafon, November 5-11, 2005, *Filtersparametrized by Orthonormal Basis Functions for active noise control*. Pages 1.
- [8] Jan Maciejowski, Introduction to Predictive Control,
- [9] Manfred Morari, Jay E. Lee, and Carlos E. Garcia, March 15, 2002, *Model Predictive Control*,pages 35.
- [10] Glorinevsky, 2005, Lecture Note, *The Concept of Model Predictive Control*
- [11] *Model Predictive Control Toolbox*, <<http://www.mathworks.com/products/mpc/>>.
- [12] Jan Maciejowski, Introduction to Predictive Control,
- [13] Glorinevsky, 2005, Lecture Note, *The Concept of Model Predictive Control*
- [14] F.G. Shinskey. Process control system.Mcgraw-Hill, 2<sup>nd</sup>. Edition,1979.
- [15] S.Alcantara, C. Pedret, R. Vilanova, P. Balaguer and A. Ibeas, Systems Engineering and Automatic Control Group, 2008.
- [16] Charles R. Nippert, Widener University, Simple Model that Illustrate DMC, 2002



# APPENDICES

## APPENDIX 1

### MATLAB Coding

#### I. Input Variation Calculation

```
function [Iv]=inputvariation(MV)
dif=filter([1 -1],[1],MV);
absdif=abs(dif);
Iv=sum(absdif);
```

#### II. Output Variation Calculation

```
function IE=outputvariation(t,y,SP)
%This function calculates the integral error
IT1=trapz(t,y);
IT2=trapz(t,SP);
IE=IT2-IT1;
```

#### III. Error Calculation

```
function IE=outputvariation(t,y,SP)
%This function calculates the integral error
IT1=trapz(t,y);
IT2=trapz(t,SP);
IE=IT2-IT1;
```

#### IV. Error Calculation for Model Predictive Controller

```
h = findobj(gcf, 'type', 'line');
xvalues = get(h, 'xdata');
yvalues = get(h, 'ydata');
p=cell2mat(yvalues(3));p=p';
q=cell2mat(xvalues(3));q=q';
yvsp=1
e=abs(p-yvsp);
IAE=trapz(q,e)
e2=e.^2;
ISE=trapz(q,e2)
e3=e.*q;
ITAE=trapz(q,e3)
```

V. Transfer Function  $T_i = 2$

```
>> Gp1=tf([-2 1],[4 1])
```

```
Gp2=tf([1],[10 1])
```

Transfer function:

$$\frac{-2s + 1}{4s + 1}$$

-----

$$4s + 1$$

Transfer function:

$$1$$

-----

$$10s + 1$$

```
>> Gp=Gp1*Gp2
```

Transfer function:

$$\frac{-2s + 1}{40s^2 + 14s + 1}$$

-----

$$40s^2 + 14s + 1$$

```
>>
```



VI. Transfer Function  $T_i = 4$

```
>> Gp1=tf([-4 1],[4 1])
```

```
Gp2=tf([1],[10 1])
```

Transfer function:

$$\frac{-4s + 1}{4s + 1}$$

-----

$$4s + 1$$

Transfer function:

$$\frac{1}{10s + 1}$$

-----

$$10s + 1$$

```
>> Gp=Gp1*Gp2
```

Transfer function:

$$\frac{-4s + 1}{40s^2 + 14s + 1}$$

-----

$$40s^2 + 14s + 1$$

```
>>
```

VII. Transfer Function  $T_i = 6$

```
>> Gp1=tf([-6 1],[4 1])
```

```
Gp2=tf([1],[10 1])
```

Transfer function:

$$\frac{-6s + 1}{4s + 1}$$

-----

$$4s + 1$$

Transfer function:

$$1$$

-----

$$10s + 1$$

```
>> Gp=Gp1*Gp2
```

Transfer function:

$$\frac{-6s + 1}{40s^2 + 14s + 1}$$

-----

$$40s^2 + 14s + 1$$

```
>>
```

VIII. Transfer Function  $T_i = 8$

```
>> Gp1=tf([-8 1],[4 1])
```

```
Gp2=tf([1],[10 1])
```

Transfer function:

$$\frac{-8s + 1}{4s + 1}$$

-----

$$4s + 1$$

Transfer function:

$$\frac{1}{10s + 1}$$

-----

$$10s + 1$$

```
>> Gp=Gp1*Gp2
```

Transfer function:

$$\frac{-8s + 1}{40s^2 + 14s + 1}$$

-----

$$40s^2 + 14s + 1$$

```
>>
```