



# Design and Implementation of Controller using MPC Toolbox

Omkar P. Dahale

Department of Electronics Engg  
Ramrao Adik Institute of Technology,  
Dr. D.Y. Patil Road, Sector 7, Phase-I,, Nerul,, Navi  
Mumbai – 400 706.

Sharad P. Jadhav

Department of Instrumentation Engg  
Ramrao Adik Institute of Technology,  
Dr. D.Y. Patil Road, Sector 7, Phase-I,, Nerul,,  
Navi Mumbai – 400 706.

## ABSTRACT

Model Predictive Control Toolbox provides tools for systematically analyzing, designing, and tuning model predictive controllers. Design and Simulation of model predictive controllers using functions in MATLAB or blocks can be done in Simulink. The predictive model, control and prediction horizons, input and output constraints, and weights can also be modified. The toolbox enables us to diagnose issues that could lead to run-time failures and provides advice on changing weights and constraints to improve performance and robustness. By running different scenarios in linear and nonlinear simulations, the controller performance can be evaluated. Adjustments to controller performance can be made as it runs by tuning weights and varying constraints. For rapid prototyping and embedded system design, the toolbox supports C-code generation.

## Keywords

Model Predictive Control (MPC), MPC Toolbox, Single Input Single Output (SISO), Single Input Multiple Outputs (SIMO), MATLAB, Simulink, MPC Controller.

## 1. INTRODUCTION

Model Predictive Control originated in the late seventies and has considerably developed since then. The term Model Predictive Control does not designate a specific control strategy but a very ample range of control methods which make an explicit use of a model of the process to obtain the control signal by minimizing an objective function. These design methods lead to linear controllers which have practically the same structure and present adequate degrees of freedom. The various MPC algorithms only differ amongst themselves in the model used to represent the process and the noises and the cost function to be minimized. This type of control is of an open nature within which many works have been developed, being widely received by the academic world and by industry. [14]

## 2. STRUCTURE OF MPC

The basic structure of MPC is shown in the figure 1. A model is used to predict the future plant outputs, based on the past and current values and on the proposed optimal future control actions. These actions are calculated by the optimizer taking into account the cost function as well as the constraints. The process model plays a decisive role in the controller. [1] The chosen model must be capable of capturing the process dynamics so as to precisely predict the future outputs as well as being simple to implement and to understand. [5] As MPC is not a unique technique but a set of different methodologies,

there are many types of models used in various formulations.

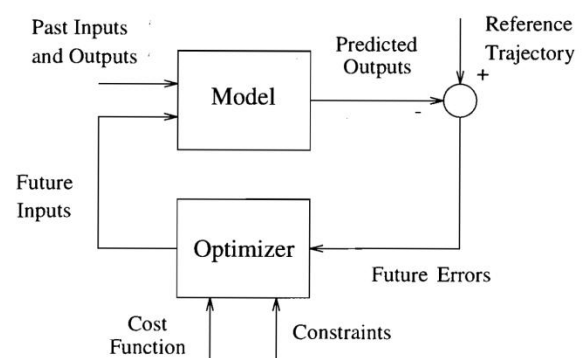


Figure 1: Basic Structure of MPC

## 3. DESIGNING AND SIMULATING MODEL PREDICTIVE CONTROLLERS

Model predictive controllers can be used to optimize closed-loop system performance of SISO/MIMO plants subject to input and output constraints. Because they base their actions on an internal plant model, model predictive controllers can forecast future process behavior and adjust control actions accordingly. [1] The ability to model process interactions often enables model predictive controllers to outperform multiple PID control loops, which require individual tuning and other techniques to reduce loop coupling. Model Predictive Control Toolbox provides functions, Simulink blocks, and a graphical tool for designing and simulating model predictive controllers in MATLAB and Simulink. Iteratively improvement of the controller design can be done by defining an internal plant model, adjusting controller parameters such as weights and constraints, and simulating closed-loop system response to evaluate controller performance. [3]

### 3.1 Defining Internal Plant Models

When designing a model predictive controller in Simulink, use Simulink Control Design to extract a linearized form of the Simulink model and automatically import it into the controller as the internal plant model. [8][10] Alternatively, use of linear time-invariant (LTI) systems from Control System Toolbox, such as a transfer function or a state-space model, to specify the internal plant model. Then import LTI models from the MATLAB workspace or from MAT-files into the toolbox. The toolbox also lets us directly import



models created from measured input-output data using System Identification Toolbox. [4]

### 3.2 Designing Controllers

Once the internal plant model is defined, complete the design of model predictive controller by specifying the following controller parameters:

- Prediction and Control Horizons
- Hard and Soft constraints on manipulated variables and output variables
- Weight on manipulated variables and output variables
- Models for measurement noise and for unmeasured input and output disturbances

In addition to constant constraints and weights, the toolbox supports time-varying constraints and weights, constraints on linear combinations of manipulated variables and output variables, terminal constraints and weights, and constraints in the form of linear off-diagonal weights. The toolbox also supports constraint softening. [4]

### 3.3 Running Closed-Loop Simulations

MATLAB functions or a graphical tool are used to run closed-loop simulations of model predictive controller against linear plant models. The graphical tool lets us set up multiple simulation scenarios. For each scenario one can specify controller set points and disturbances by choosing from common signal profiles, such as step, ramp, sine wave, or random signal. [7][9]

To assess the effects of model mismatch, one can simulate a controller against a linear plant model that is different from the internal plant model used by the controller.[12][13] One can also simulate multiple controller designs against the same plant model to see how different weight and constraint settings affect controller performance. The toolbox lets us disable constraints to evaluate characteristics of the closed-loop dynamics, such as stability and damping. [2]

Using Simulink blocks provided with Model Predictive Control Toolbox, closed-loop simulations of model predictive controller against a nonlinear Simulink model is performed. Also the block can be configured to accept time-varying constraint signals that are generated by other Simulink blocks.

## 4. EXAMPLES AND CASE STUDY

In this paper two examples one of a first order SISO system and the second is first order SISO system with time delay have been used. Also a case study of Servo mechanism controller is done for better understanding.

### 4.1 Example I

This example consists of a first order SISO system. The idea behind using this example is to study the MPC Toolbox and to understand it's working. The transfer function of the systems is given by:

$$f(s) = \frac{1}{s + 1}$$

**Equation 1**

For this transfer function is designed and implemented MPC controller using MPC Toolbox in MATLAB.

The control interval is set to 0.1, the prediction horizon is set to 10 and the control horizon is set to 2. The controller is designed for unconstrained input and output. The designing and simulation results are shown in the figures at the end of the paper. Refer figure 6 and 7.

### 4.2 Example II

This example consists of a first order SISO system with delay. The idea behind using this system is to study the effect of delay on the system and also on the designing, working of the controller. The transfer function of the system is given by:

$$f(s) = \frac{10 e^{-0.4s}}{5s + 1}$$

**Equation 2**

The control interval is set to 0.01, the prediction horizon is set to 10 and the control horizon is set to 2. The reason to use even lower control interval is to lesser the effect of delay on the system and the controller. The controller is designed for unconstrained input and output. The designing and simulation results are shown in the figures at the end of the paper. Refer figure 8 and 9.

### 4.3 Case Study

Servomechanism controller is used for case study. It is an example of Single input multiple output (SIMO) system. It has one measured output while the other is unmeasured.

#### 4.3.1 System Model

A position servomechanism consists of a DC motor; gearbox, elastic shaft, and a load. The differential equations representing this system are:

$$\begin{aligned} \omega_L &= -\frac{k_\theta}{J_L} \left( \theta_L - \frac{\theta_M}{\rho} \right) - \frac{\beta_L}{J_L} \omega_L \\ \omega_M &= \frac{k_T}{J_M} \left( \frac{V - k_T \omega_M}{R} \right) - \frac{\beta_M \omega_M}{J_M} + \frac{k_\theta}{\rho J_M} \left( \theta_L - \frac{\theta_M}{\rho} \right) \end{aligned}$$

Where  $V$  is the applied voltage,  $T$  is the torque acting on the load,  $\omega_L$  is the load's angular velocity,  $\omega_M = \theta_M$  is the motor shaft's angular velocity, and the other symbols represent constant parameters (see Table 1 for more information on these).

The state variables are defined as  $x_p = [\theta_L \omega_L \theta_M \omega_M]^T$ , hence, convert the above model to an LTI state-space form:

$$\dot{x}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_\theta}{J_L} & \frac{\beta_L}{J_L} & \frac{k_\theta}{\rho J_L} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_\theta}{\rho J_M} & 0 & -\frac{k_\theta}{\rho^2 J_M} & -\frac{\beta_M + k_T^2/R}{J_M} \end{bmatrix} x_p + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_T}{R J_M} \end{bmatrix} V$$

$$\theta_L = [1 \ 0 \ 0 \ 0] x_p$$

$$T = \left[ k_\theta \ 0 \ -\frac{k_\theta}{\rho} \ 0 \right] x_p$$



**Table 1. Parameters used in the Servomechanism Model**

Symbol	Value (SI units)	Definition
$k_{\theta}$	1280.2	Torsional rigidity
$k_T$	10	Motor constant
$J_M$	0.5	Motor Inertia
$J_L$	$50J_M$	Nominal Load Inertia
$\rho$	20	Gear ratio
$\beta_M$	0.1	Motor viscous friction coefficient
$B_L$	25	Load viscous friction coefficient
$R$	20	Armature resistance

### 4.3.2 Control Objectives and Constraints

The controller must set the load's angular position,  $\theta_L$ , at a desired value by adjusting the applied voltage,  $V$ . The only measurement available for feedback is  $\theta_L$ .

The elastic shaft has finite shear strength, so the torque,  $T$ , must stay within specified limits

$$|T| \leq 78.5 \text{ Nm}$$

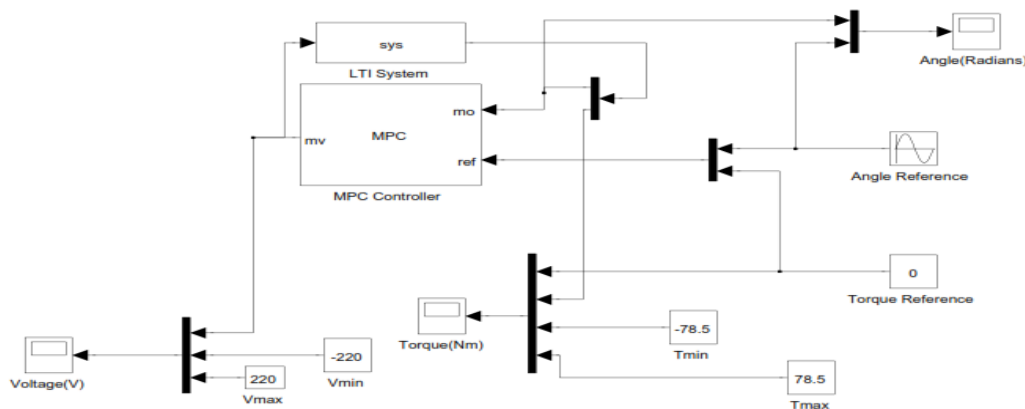
Also, the applied voltage must stay within the range

$$|V| \leq 220 \text{ V}$$

From an input/output viewpoint, the plant has a single input,  $V$ , which is manipulated by the controller. It has two outputs, one measured and fed back to the controller,  $\theta_L$ , and one unmeasured,  $T$ .

### 4.3.3 Defining the plant model

The first step in a design is to define the plant model. The following commands are from the `mpcdemos` file `motor_model.m`, which we can run instead of entering the commands manually.



**Figure 2: Simulink model of Servomechanism Controller**

### 4.3.4 Creating Simulink model and designing controller

After running the m-file of the servo plant, one can now design and simulate the controller for the system. After designing the servomechanism system in Simulink as shown

in the figure 2, run the `MPCTOOL` command in MATLAB command window. Then by selecting the import plant command, import the servo plant model from the workspace. The MATLAB creates the MPC controller after importing the plant model. Now just specify the constraints and parameters in the controller as shown in the figures

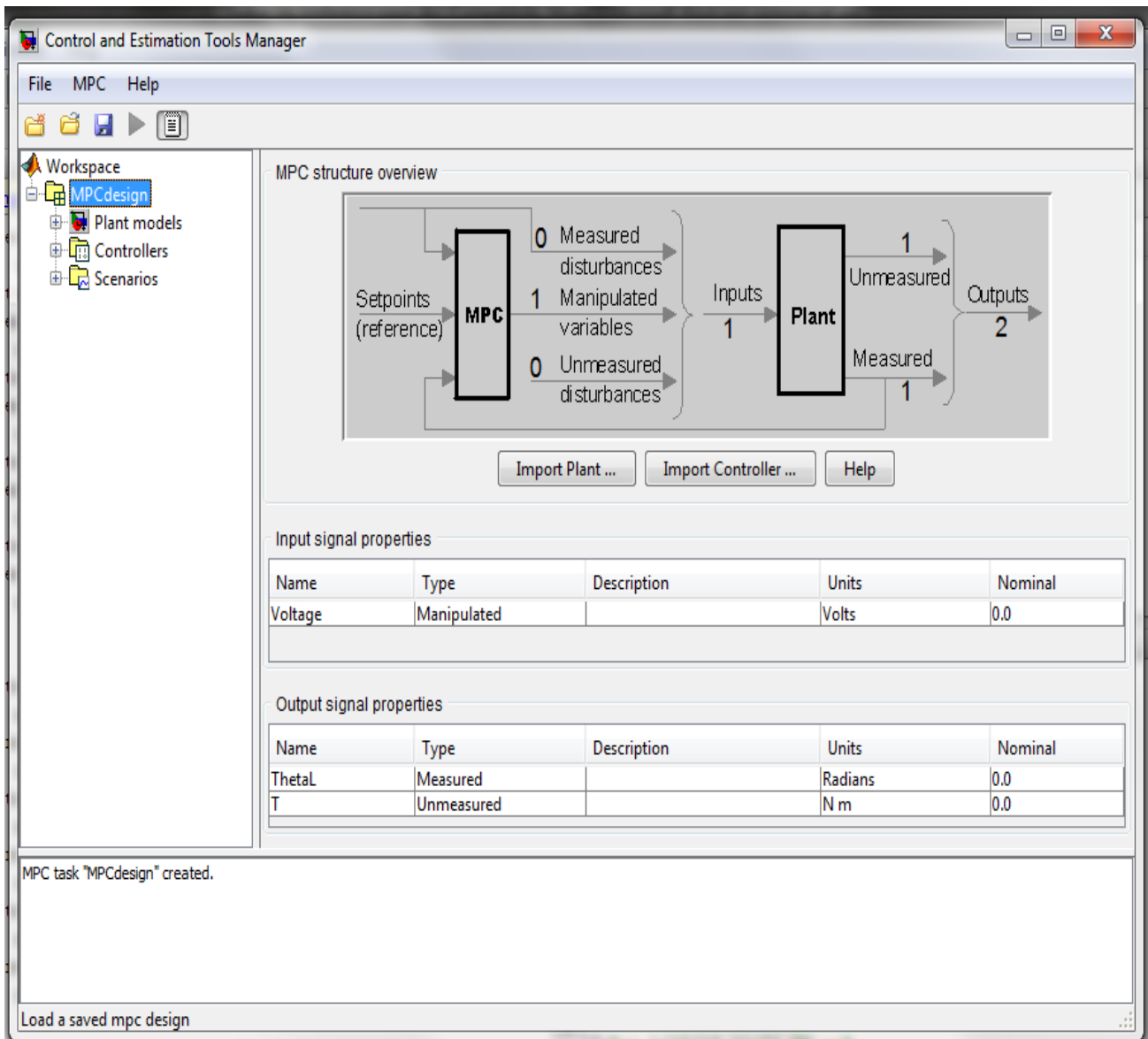


Figure 3: Design tools after importing the Plant model and Specifying Signal Properties



#### 4.3.5 Outputs

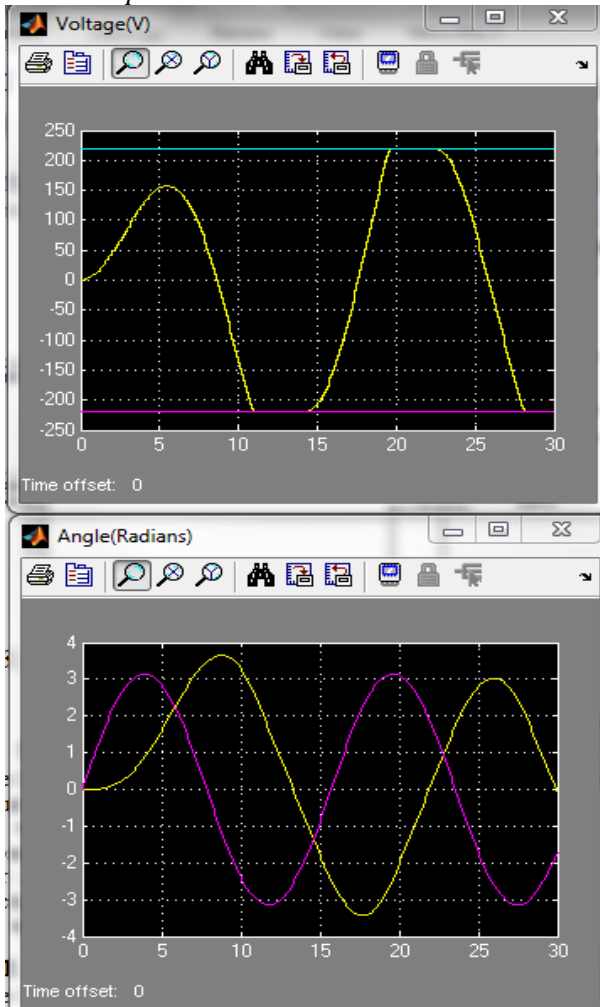


Figure 4: Voltages and Angle

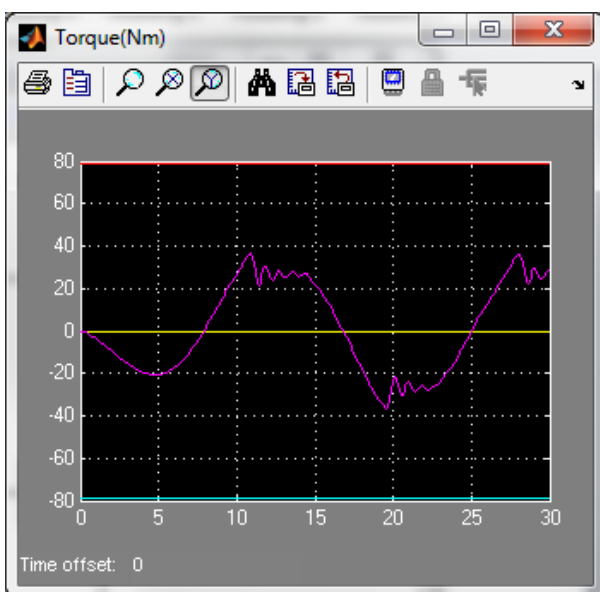


Figure 5: Torque

From the results it is clearly seen that the controller is taking actions in defined constraints on the input and also on the output.

### 5. CONCLUSION

The aim of this paper is to give insight into model predictive control and run Matlab simulations to illustrate some of the theory for linear systems using a generalized system. MPC is popular due to its ability to yield high performance control systems capable of operating without expert intervention for long periods of time and also its ability to handle constraints. In this work, the importance of MPC, components of MPC, and mentioned some practical applications of MPC are presented.

It can be concluded that using MPC toolbox of MATLAB is very beneficial in designing of controller. It is less time consuming and also robust. Input and output constraints are also taken care of by the controller.

MATLAB can also be interfaced with the hardware using a suitable intermediate interface cards like DAQ cards, Elvis board of National Instruments. So these controllers can be used with any type of hardware and also in practical applications and also for the industrial purpose too.

The future work will be to implement the controller developed to multiple input multiple output (MIMO) system. It also includes incorporating the controller for nonlinear systems considering the noise and disturbances.

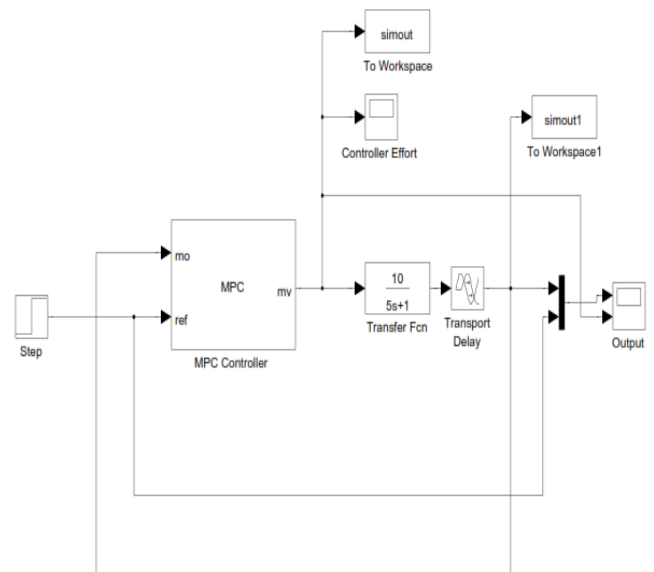


Figure 6: First order system

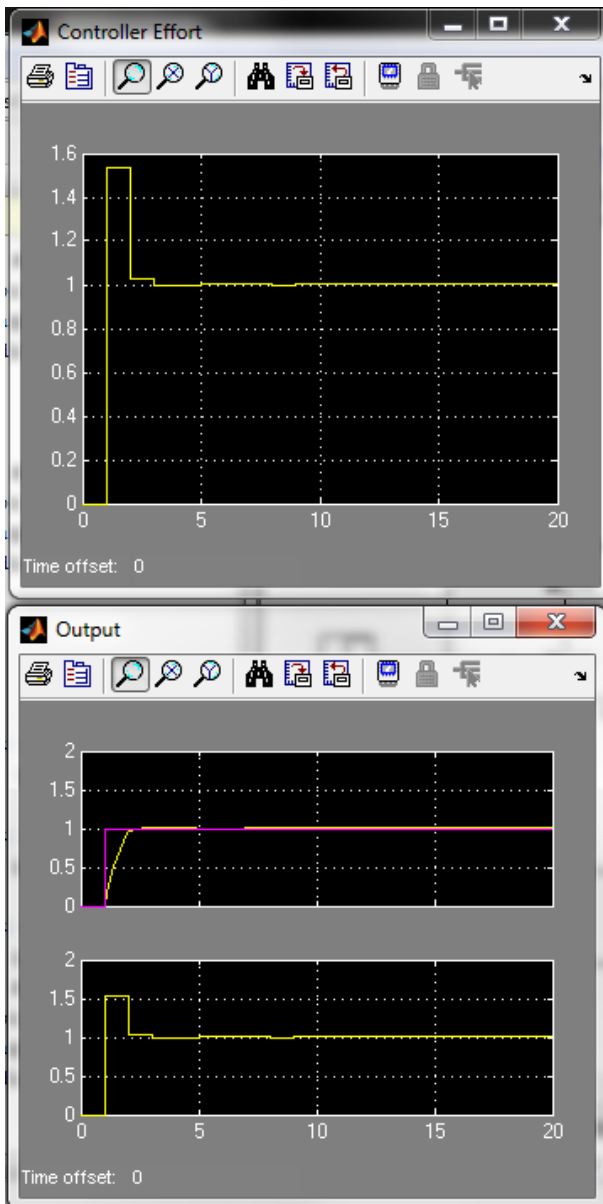


Figure 7: Result of First Order System

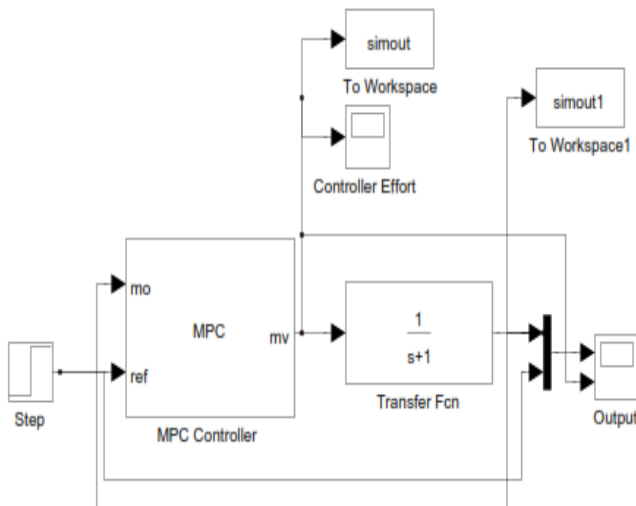


Figure 8: First Order System with Delay

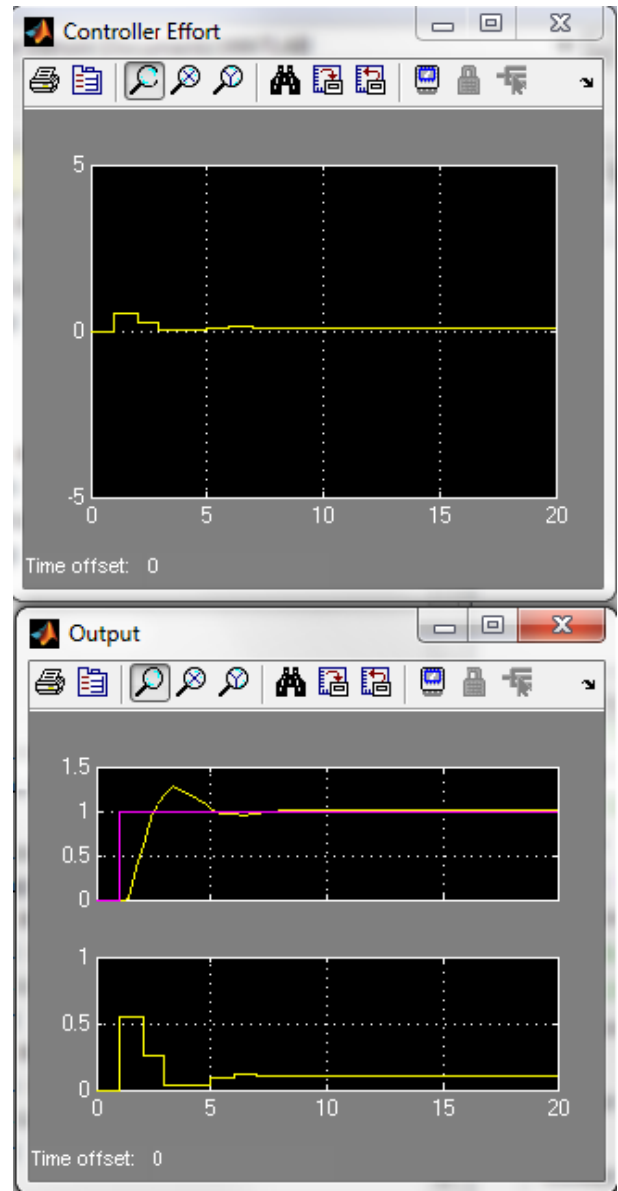


Figure 9: Result of First order system with delay

## 6. REFERENCES

- [1] Eduardo F. Camacho and Carlos Bordons. Model Predictive Control. Springer, 2007.
- [2] J.M. Maciejowski. Predictive Control with Constraints. Pearson Education Limited, 2002.
- [3] J. A. Rossiter. Model-Based Predictive Control, A Practical Approach. CRC Press LLC, 2005.
- [4] Liuping Wang. Model Predictive Control System Design and Implementation Using MATLAB. Springer, 2009.
- [5] B. Wayne Bequette. Process Control: Modeling Design and Simulation. Prentice Hall PTR, 2002.
- [6] Mathworks. Matlab documentation.



- [7] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, pp. 733–764, 2002.
- [8] J. B. R. D Q Mayne and C. V. Rao, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, 2000.
- [9] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: theory and practice,” *Automatica*, vol. 25, 1989.
- [10] J. B. Jorgensen, “Moving horizon estimation and control,” 2005.
- [11] C. Brosilow and B. Joseph, *Techniques of Model-Based Control*. Prentice Hall, 2002.
- [12] W. H. Fleming and R. W. Rishel, “Deterministic and stochastic optimal control,” *Springer*, 1975.
- [13] B. L. Kleinman, “An easy way to stabilize a linear constant system.,” *IEEE Transactions on Automatic Control*, vol. 15, no. 12, p. 693, 1970.
- [14] S. J. Qin and T. A. Badgwell, “An overview of industrial model predictive control technology.” *Fifth International Conference on Chemical Process Control*, pp. 232–256, 1997.