Applied Model Predictive Control – a brief guide do MATLAB/Simulink MPC toolbox

Technical Report · August 2014

DOI: 10.13140/RG.2.1.2334.7369

Stefano Ciannella
Texas Tech University

Chemical Engineering & Applied Chemistry
UNIVERSITY OF TORONTO

APPLIED MODEL PREDICTIVE CONTROL

A brief guide to MATLAB/Simulink® MPC toolbox

by

STEFANO CIANNELLA

under the guidance and supervision of Professor W.R. Cluett, Ph.D.

August 2014

Contents

List of symbols

y	Measured output
$\hat{y}$	Predicted output
$\widehat{y^c}$	Predicted output after model correction
u	Manipulated variable
Δu	Change in the manipulated variable from one sampling instant to the next
k	Current sampling instant
M	Control horizon, or number of control moves
P	Prediction horizon
r	Setpoint
d	Difference between predicted and measured outputs
w	Weight for changes in the manipulated variable
S_N	N step-response coefficients
MV	Manipulated variable
CV	Controlled variable
DV	Disturbance variable

Acknowledgments

In first place, I would like to acknowledge Professor W.R. Cluett at the University of Toronto whom taught Process Dynamics and Control as an undergraduate course and supervised me during my Summer research, stimulating and encouraging my interest in this field of study. Also, the Chemical Engineering and Applied Chemistry department staff for orientation and support over all the courses and academic work I have engaged during my stay in Canada.

Secondly, my scholarship agency CNPq for financial support and the Science without Borders academic exchange program, which offered the opportunity to study abroad and learn about North America high education level and lifestyle.

Finally, my friends and colleagues whom provided me a comfortable and enjoyable working experience, and naturally the support from my family whom, even standing far in Brazil, supported me in all possible ways.

1. Introduction

Model Predictive Control (MPC) is a class of control techniques first derived from Internal Model Control, or IMC, and is widely applied in the process industries due to its capability to deal with constraints in an optimal fashion; as the name suggests, MPC is based on predictions of setpoint tracking behavior or disturbance rejection over both past controlled and manipulated variables measurements, in which each prediction is followed by an optimization routine to find the optimal input for the closed loop response imposed by a certain criteria, such as maximizing a profit function or production rate.

This text provides a succinct background on the MPC philosophy and modeling equations, followed by a step-by-step guide to how to implement predictive techniques using MATLAB/Simulink for SISO and MIMO systems. Furthermore, the role of the various parameters within MPC Simulink toolbox and how to find proper values to obtain a desired closed response shall be addressed.

After reading this text material, one should become familiar with the basic ideas behind SISO and MIMO MPC, including the following:

- MPC block insertion and set-up in a Simulink diagram;
- Evaluate parameters such as prediction and control horizons and input/output weights;
- The effect of such parameters on the control performance and system stability.

2. Model predictive control overview

As briefly discussed in the introduction, MPC controllers are widely applied over process and oil/petrochemical industries due to its capability to deal in an optimal form with input/output process constraints (upper or lower values for a specific variable, for instance); in this sense, one should be able to safely operate their system by restricting it to be conducted in a limited region of operation, such as maximum and minimum liquid level within a distillation column or a maximum opening degree in a valve.

Basically, MPC calculations are performed at each sampling time which can be also set by the control designer; these calculations are based on current measurements and predictions of future output values. Two types of computation are primordial in a MPC controller: setpoint calculations and control calculations, this last one includes process constraints and other parameters that are able to be manually specified. The main task of a MPC controller is to determine a sequence of control moves in the manipulated variable, so the system can be tracked to its setpoint in an optimal fashion.

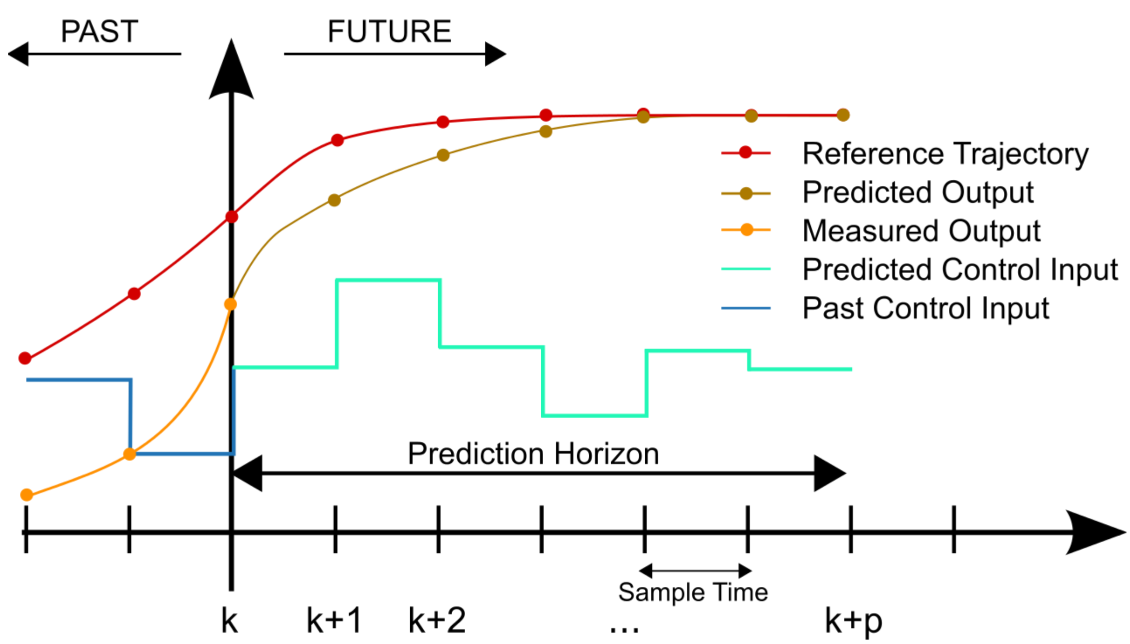


Figure 1 - Basic concept of MPC

The control calculations are based on minimizing the predicted deviations from the reference trajectory. The key idea behind a MPC controller is illustrated in Figure 1. When using MPC, an optimization problem is solved at each time step k through an objective function based on output predictions over a prediction horizon of P time steps; this objective function (usually a quadratic one) is minimized by a selection of manipulated variables moves over a control horizon of M control moves. It's important to emphasize that, even though at each time step a group of M moves is calculated, only the first one u_k is implemented. After this step, the measurement at the next time instant y_{k+1} is obtained, followed by a correction due to model error, and then a new optimization problem is solved again. These procedures are carried every for every time step k.

Before starting to enlighten the reader with MPC equations, it's important to assert the definition of a finite step response (FSR) model; these models are used in almost all MPC modeling equations and are obtained by simply introducing a unit step change to a process operating at a steady state. The model coefficients $(S_1, S_2, S_3, \dots, S_N)$ are the output values at each time step.

The predicted output for an instant time k is calculated through the following equation:

$$\widehat{y_k} = \sum_{i=1}^{N-1} S_i \Delta u_{k-i} + S_N u_{k-N} \tag{i}$$

The model predicted output is hardly equal to the actual measured output at a certain time step k; this discrepancy is denoted by:

$$d_k = y_k - \widehat{y_k} \tag{ii}$$

The corrected prediction is then:

$$\widehat{y_k^c} = \widehat{y_k} + d_k \tag{iii}$$

By combining equations (i), (ii) and (iii), the corrected prediction for the *j*th step into the future can be demonstrated as:

$$\widehat{y_{k+j}^c} = \sum_{i=1}^{j} S_i \Delta u_{k-i+j} + \sum_{i=1}^{N-1} S_i \Delta u_{k-i+j} + S_N u_{k-N+j} + d_{k+j} \quad \text{(iv)}$$

Equation (iv) can be easily seen as a "compilation" of the effect of future control moves (first term to the right), and past control moves (second and third terms), as well as a correction term d_{k+j}.

Furthermore, the difference between setpoint trajectory and future predictions in jth step, with j ≤ P, is given by:

$$r_{k+j} - \widehat{y_{k+j}^c} = r_{k+j} - \left[\sum_{i=1}^{N-1} S_i \Delta u_{k-i+j} + S_N u_{k-N+j} + d_{k+j}\right] - \sum_{i=1}^{j} S_i \Delta u_{k-i+j} \quad \text{(v)}$$

Equation (v) is essential for the optimization problem and can be used in a quadratic objective function for a prediction horizon P and a control horizon of M moves:

$$f_{obj} = \sum_{i=1}^{P} \left(r_{k+i} - \widehat{y_{k+i}^c}\right)^2 + w \sum_{i=0}^{M-1} \Delta u_{k+1}^2 \quad \text{(vi)}$$

For MIMO systems, the derivation of prediction equations is primarily based over the previous analysis for SISO systems. As an example, let's consider a 2x2 system: the predictive control model will consist of two equations and four individual step-response models,

$$\widehat{y_1}(k+j) = \sum_{i=1}^{N-1} S_{11,i} \Delta u_1(k-i+j) + S_{11,N} u_1(k-N+j) + \sum_{i=1}^{N-1} S_{12,i} \Delta u_2(k-i+j) + S_{12,N} u_2(k-N+j) \quad \text{(vii)}$$

And,

$$\widehat{y_2}(k+j) = \sum_{i=1}^{N-1} S_{21,i} \Delta u_1(k-i+j) + S_{21,N} u_1(k-N+j) + \sum_{i=1}^{N-1} S_{22,i} \Delta u_2(k-i+j) + S_{22,N} u_2(k-N+j) \quad \text{(viii)}$$

where $S_{12,i}$ denotes the *i*th step-response coefficient for the model that relates $y_1$and u_2, and vice-versa. It can be easily noticed that a 2x2 system will feature two objective functions, one for y_1 and another for y_2, each one with their own parameters and input weights w_1 and w_2.The reader should feel encouraged to derive such objective functions by following the same procedure performed to SISO systems, obtaining two equations that are analogous to equation (vi).

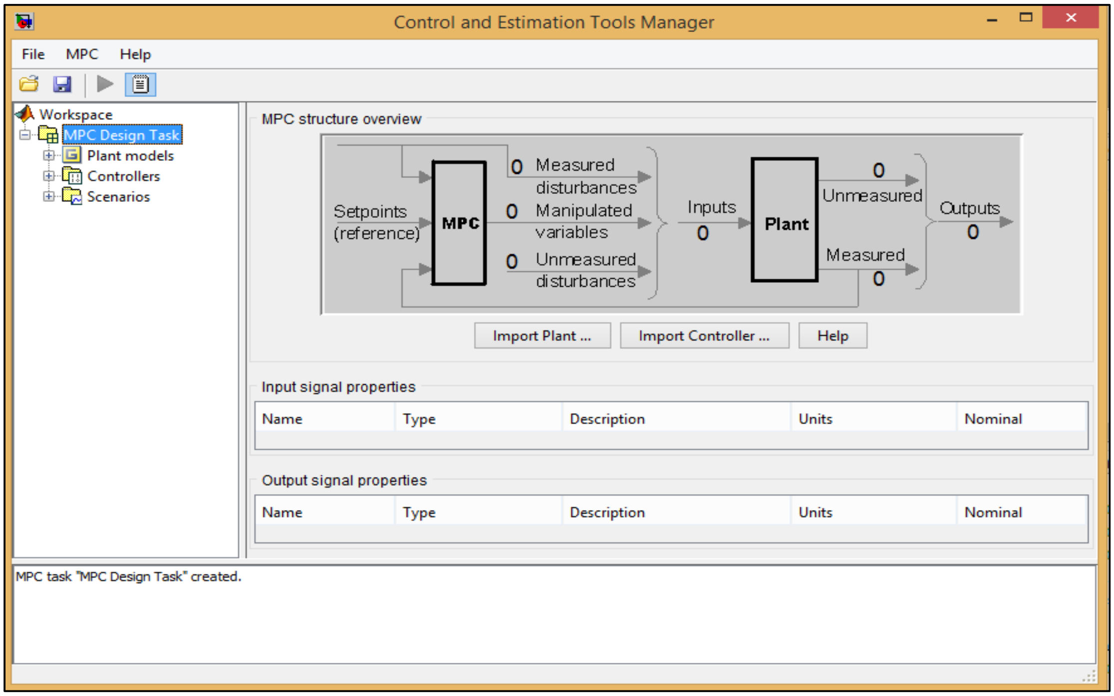
3. MATLAB/Simulink® MPC Toolbox

3.1. MPC Toolbox interface overview

To open MPC toolbox in MATLAB, type the following command at the command window:

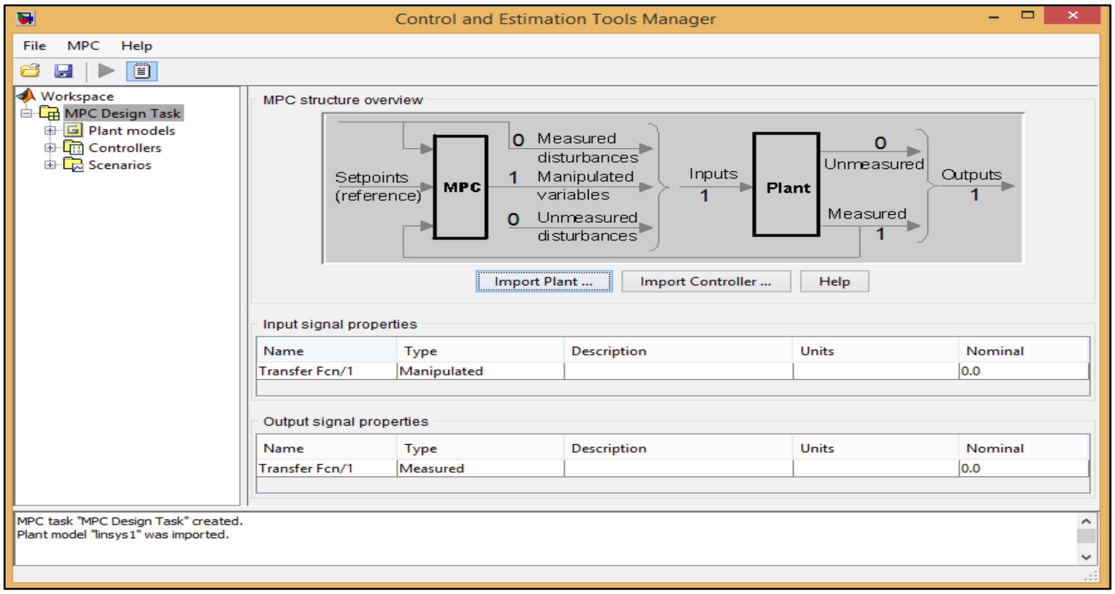
>>mpctool

After the toolbox is loaded, the following window becomes visible:



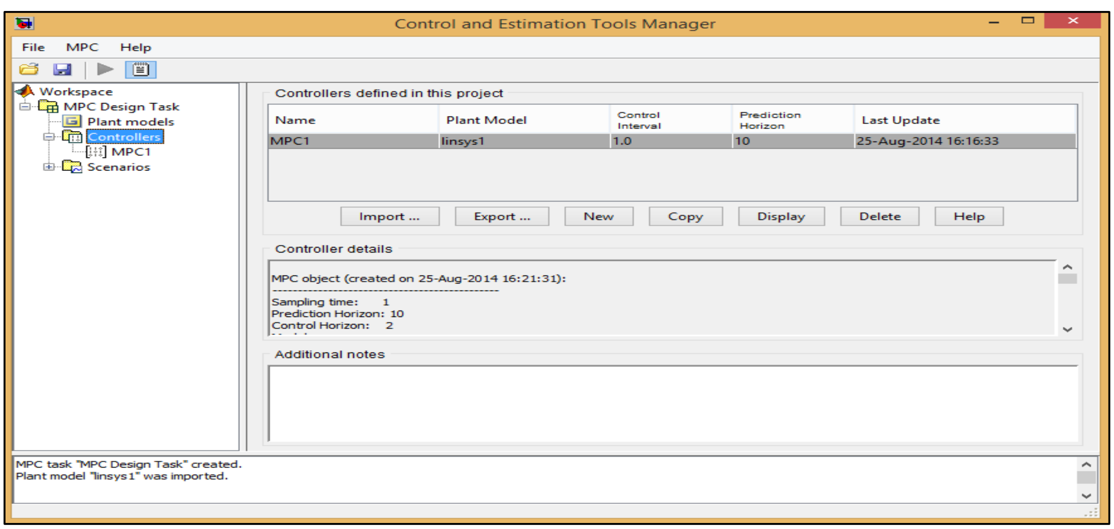
This is the "MPC Design Task" tab, in which one may import a plant from MATLAB workspace or even a controller which was previously designed; the plant can be a linearized model obtained from a Simulink block diagram (right click on a block or subsystem then choose "Linear analysis" to linearize it), or even a transfer function or state space model. Upon importing a plant or a controller, one may set input and output signal properties, such as name, type of variable (manipulated, measured disturbance or unmeasured disturbance) and their description along with physical units and a nominal value.

In order to enlighten the reader about the various tabs and options in MPC toolbox, a generic single 1x1 linearized model collected from a first-order transfer function will imported as a plant, so we can start to design a controller. After clicking on "Import Plant" and choosing a model from MATLAB workspace, the following interface becomes visible:

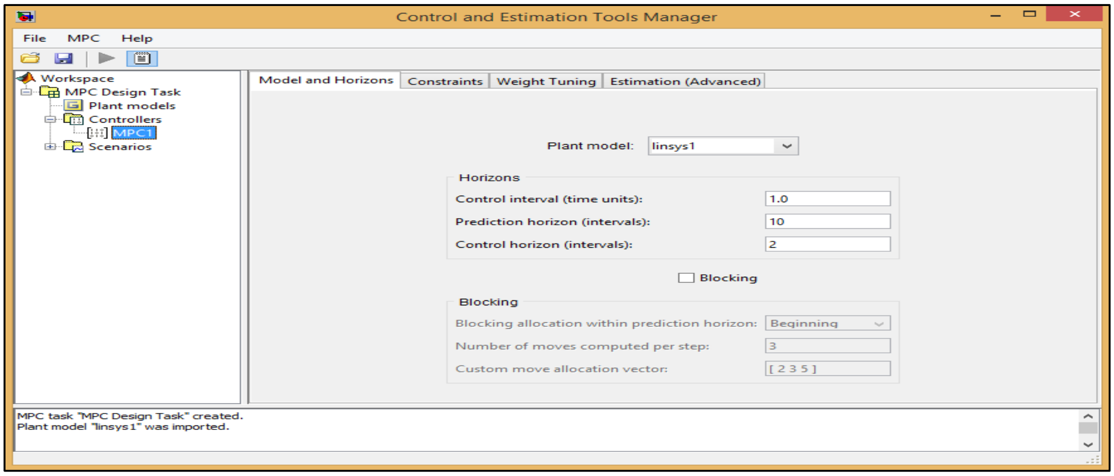


3.1.1. Controllers tab

By clicking on "Controllers" tab, you can notice the existence of a rough MPC controller archive which was created when the plant was imported into the toolbox (MPC1). In this tab, one may import or export and create or delete controllers, as well as check its configuration and control/prediction horizon values.

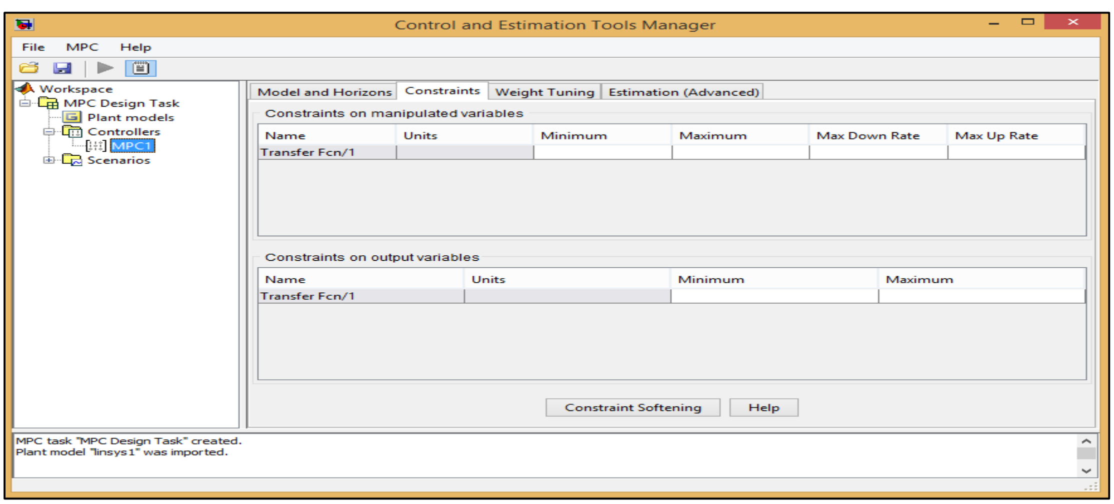


To configure the controller, click on "MPC1". This interface is the most important when it comes to design controllers and set all of its parameters; the toolbox provides default values for prediction and control horizons as well as for sampling time, however in most of the cases they are not appropriate for a first guess when designing a controller. The recommendation is to set all the parameters to one except for the sampling time, which must be equal to integration step size, and that may differ between each system.



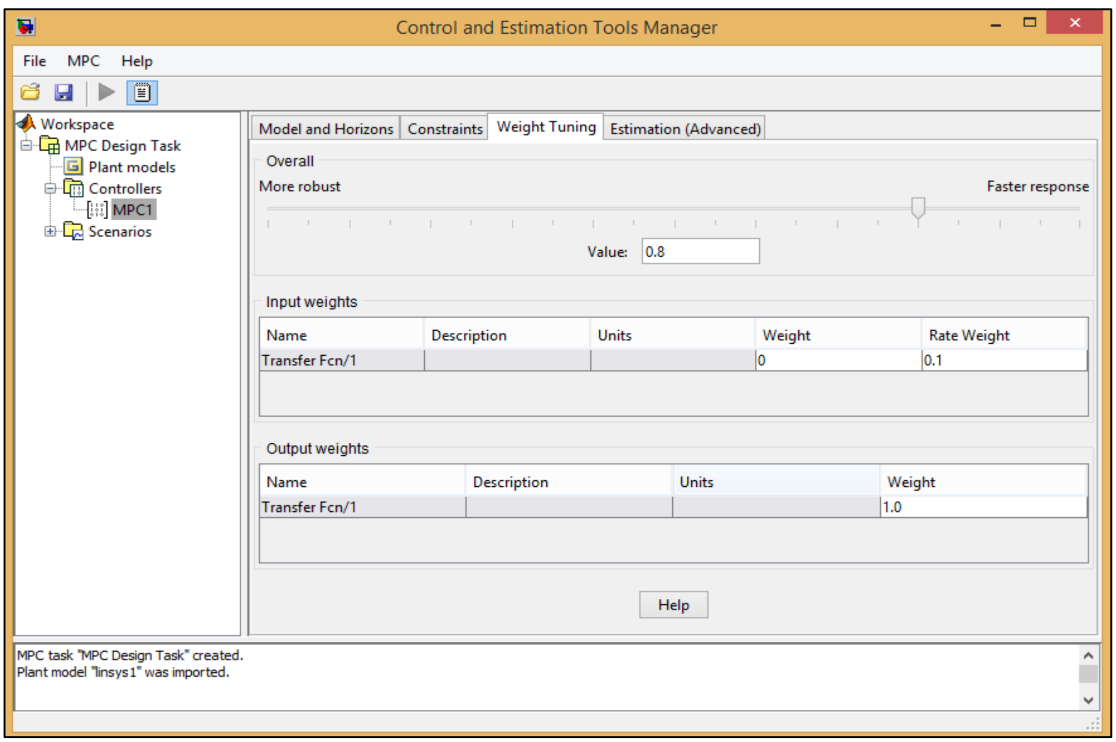
3.1.2. Constraints tab

In the "Constraints" tab it is possible to define maximum and minimum values for each one of the MV's and CV's, including their maximum down and up rates. This is a very important component when designing a controller for the MPC approach is able to deal with process constraints in an optimal fashion, so one may feel free to test and evaluate closed-loop responses with different constraints.



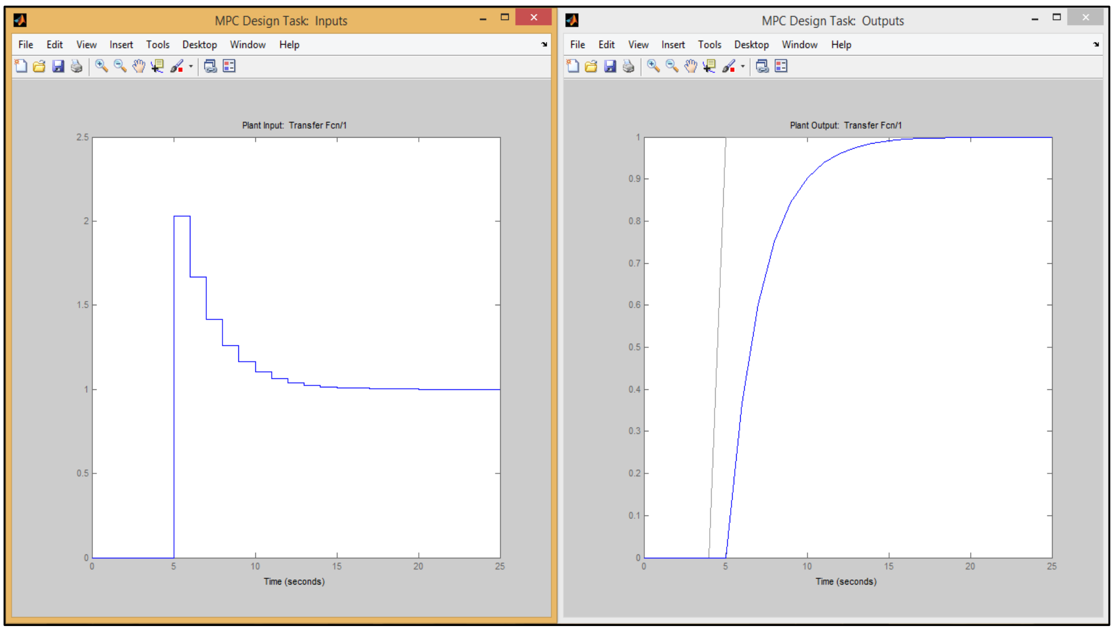
3.1.3. Weight tuning tab

The “Weight Tuning” tab provides an interface for one to choose values for input and output weights according to their desired closed-loop response. If the system is multivariable, it is possible to set a tight control routine over a certain CV by tuning output weights, or if the controller is being too aggressive or too sluggish when dealing with MV’s, tuning the input weights may turn to be a good option. Moreover, this tab provides a quick way to choose between a more robust or faster response by managing the “Overall weight” (upper slide bar); one can designate values between 0 (more robust response) and 1.0 (faster response), being aware that overall weight’s values close to one may destabilize the system depending on which values were set to prediction and control horizons.



3.1.4. Scenarios tab

Finally, the tab “Scenarios” provides a quick simulation of your system before exporting the controller to Simulink, thus being a good way to evaluate the effect of each parameter in the closed-loop response in a quick and practical form. In this tab, one can set the simulation length and a variety of types of setpoint changes (step, ramp, impulse, etc) over the CV’s, their size and time instant in which they are applied. By pressing the green arrow button on the upper left corner, the toolbox simulates the system and shows two graphs: one for the MVs and another one for the CVs; the next diagrams illustrate this feature for a step change over CV’s setpoint with P = 5, M = 1, sample of 1.0 second and a length of 25 seconds,



3.1.5. Exporting a controller

Once the designer is satisfied with their controller, it must be imported to MATLAB workspace. To do so, click on “Controllers” tab and then on “Export”; it is possible to assign a name or choose between various controllers that were designed within the toolbox, if that’s the case.

3.2. Setting up and starting a simulation

In order to introduce the reader to MPC toolbox and its configurations, a first example which covers a simple first-order transfer function will be explored and the effect of model and prediction horizons shall be addressed.

3.2.1. Example 1: First Order Process

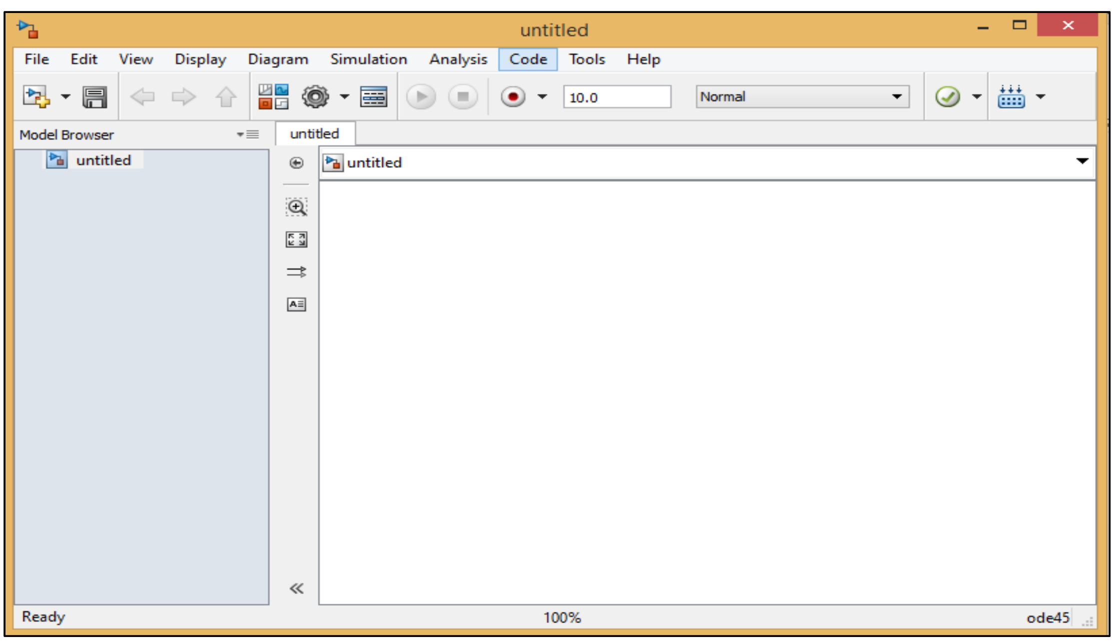
The first-order process, where the time unit is minutes, is:

$$G_p(s) = \frac{1}{5s+1} \tag{ix}$$

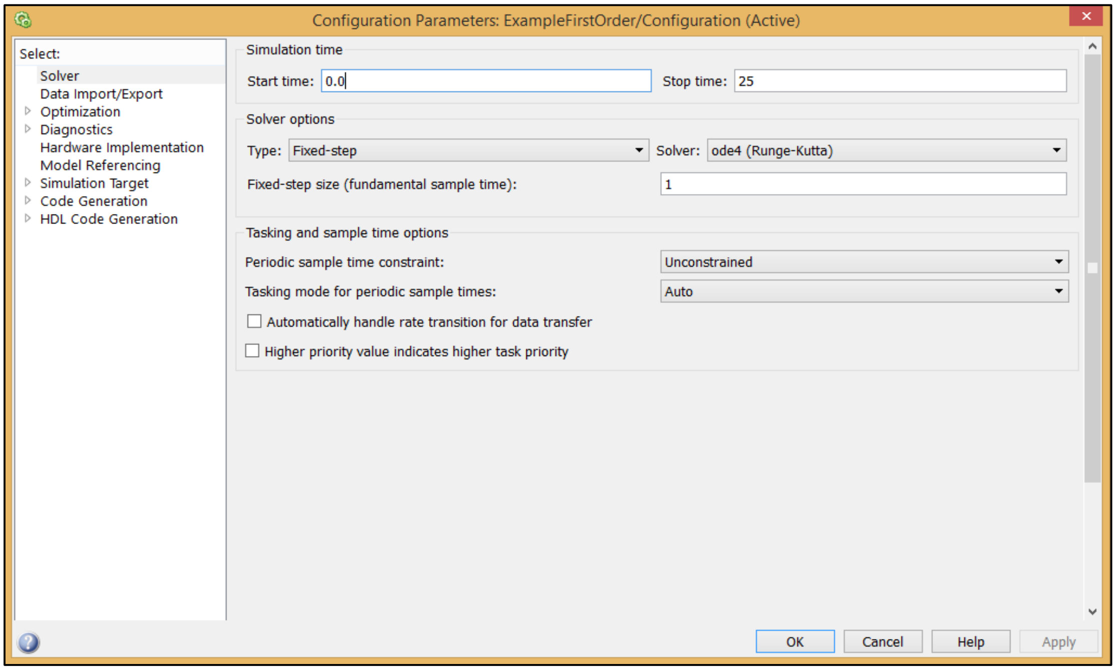
For its simulation, we will keep a time sample of 1 minute so the effect of other tuning parameters should become more visible and easy to understand. First, equation (ix) must be inserted in a Simulink block diagram along with a MPC controller block; in order to do so, one must type the following command into MATLAB's command window:

>>simulink

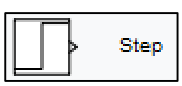
This command will open the Simulink Library Browser; next, click on "File" → "New" → "Model". The following window should become visible:

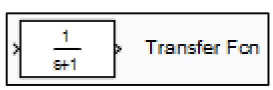


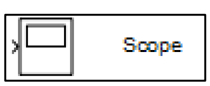
Several simulation parameters are able to be set in order to turn the calculations faster and more accurate; to do so, click on “Simulation” → “Model Configuration Parameters”, then below “Solver options” choose “Fixed step” as the type of integration step, and then choose “ode4 (Runge-Kutta)” as a solver for the simulation. Also, for this case a fixed-step size of 1 can provide a fair accuracy upon the solution of equation (ix).

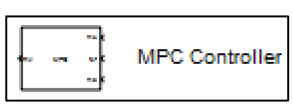


Once the basic simulation settings are defined, a block diagram can be built. For this example just a few blocks are necessary, which are listed below:

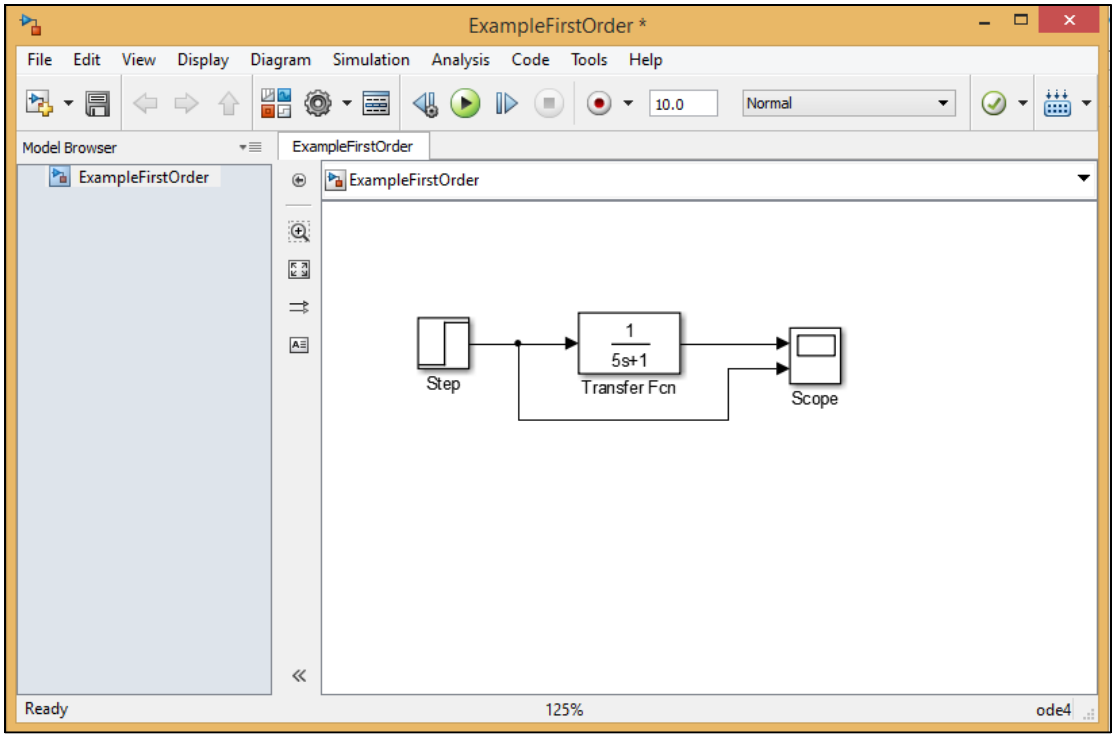
- Step block 

- Transfer function block 

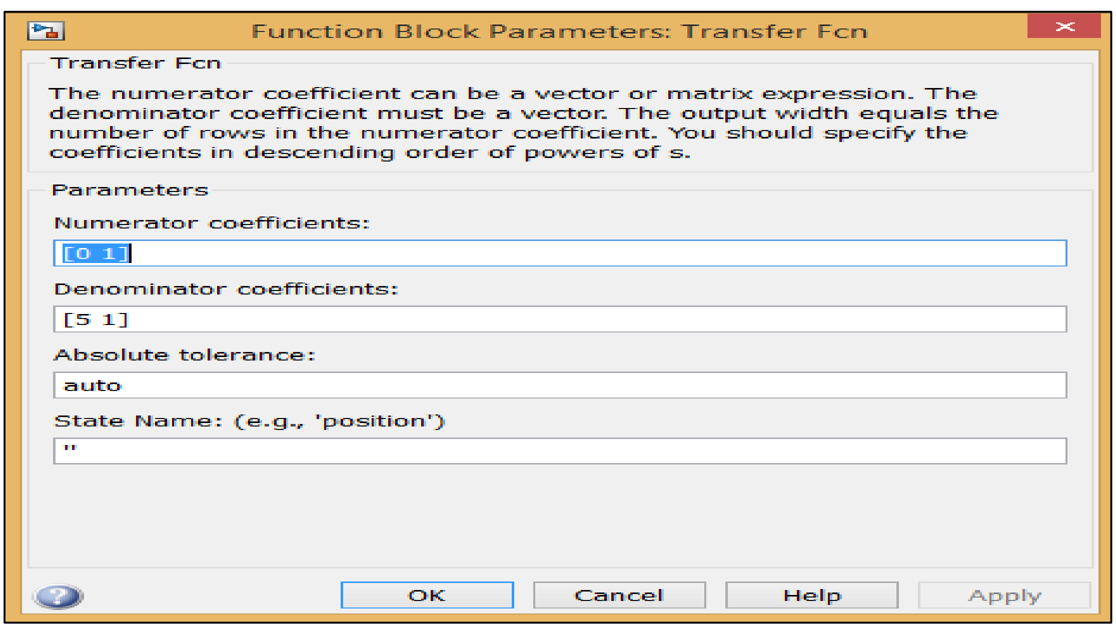
- Scope block 

- MPC controller block 

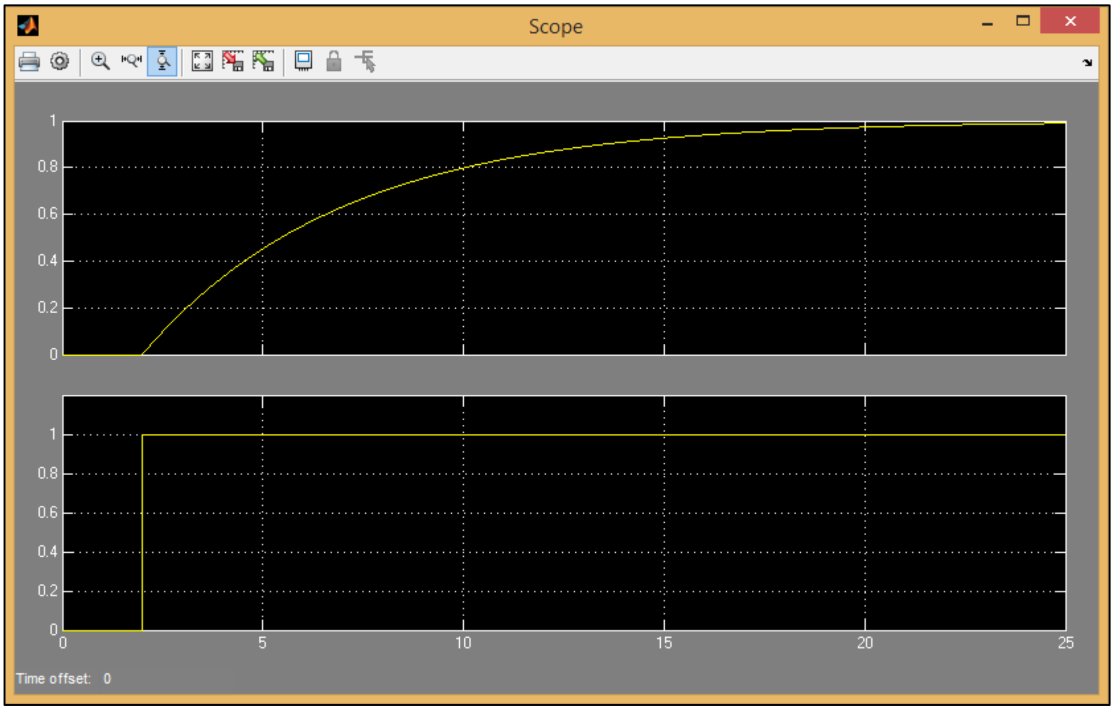
The non-controlled open-loop block diagram is showed below:



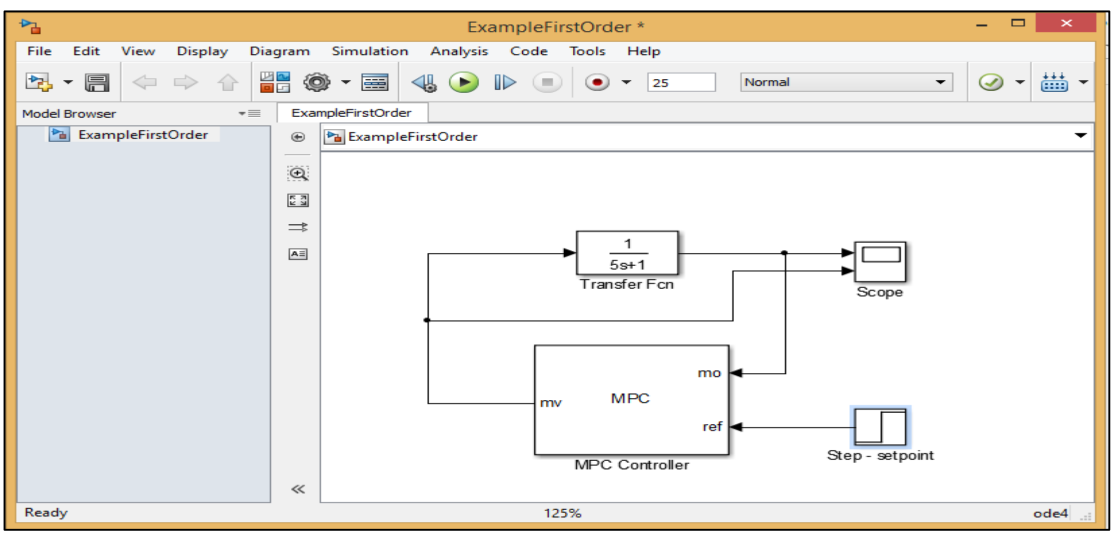
To set the coefficients within transfer function block, double click on it and specify values for numerator and denominator (zeros and poles):



Finally, after all these steps the simulation is ready to be run. To do so, click on the green circle button on the upper bar. Once the simulation is done, the results for the non-conrolled open-loop response due to a unitary step change over the MV are as follows:



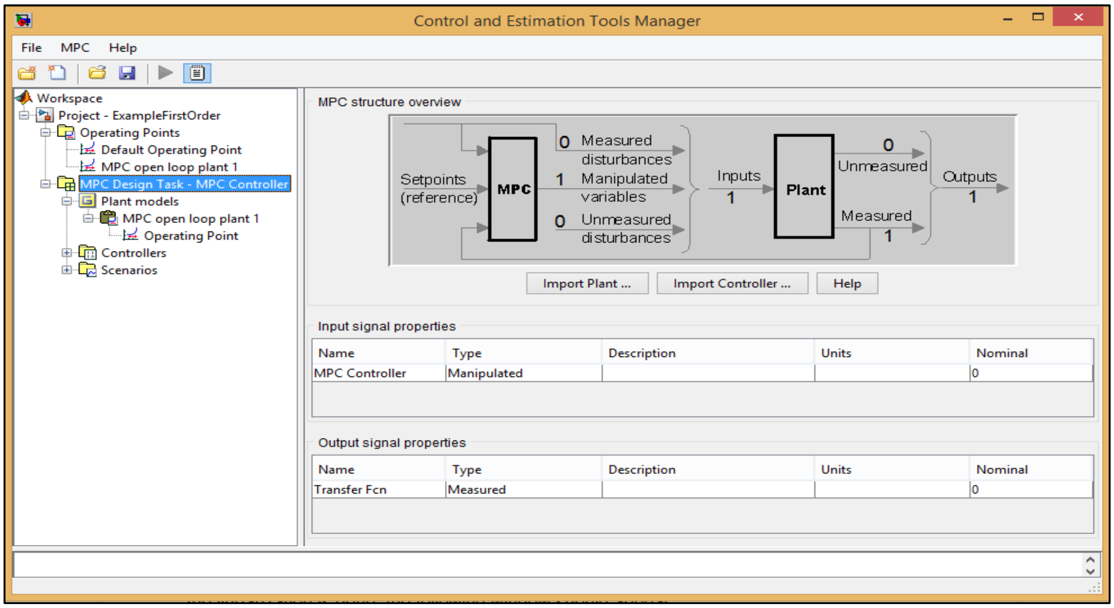
The next step is to introduce a MPC controller block in the diagram. The original diagram must be altered in a couple of ways to well insert the controller; the closed-loop block diagram is demonstrated below (the setpoint for the system will be denoted by zero):



To configure the controller, double click on the MPC controller box (uncheck the "Measured disturbance" box first) and the click on "Design". A small window will show up asking the number of manipulated and measured variables, as well the sample time for the simulation;

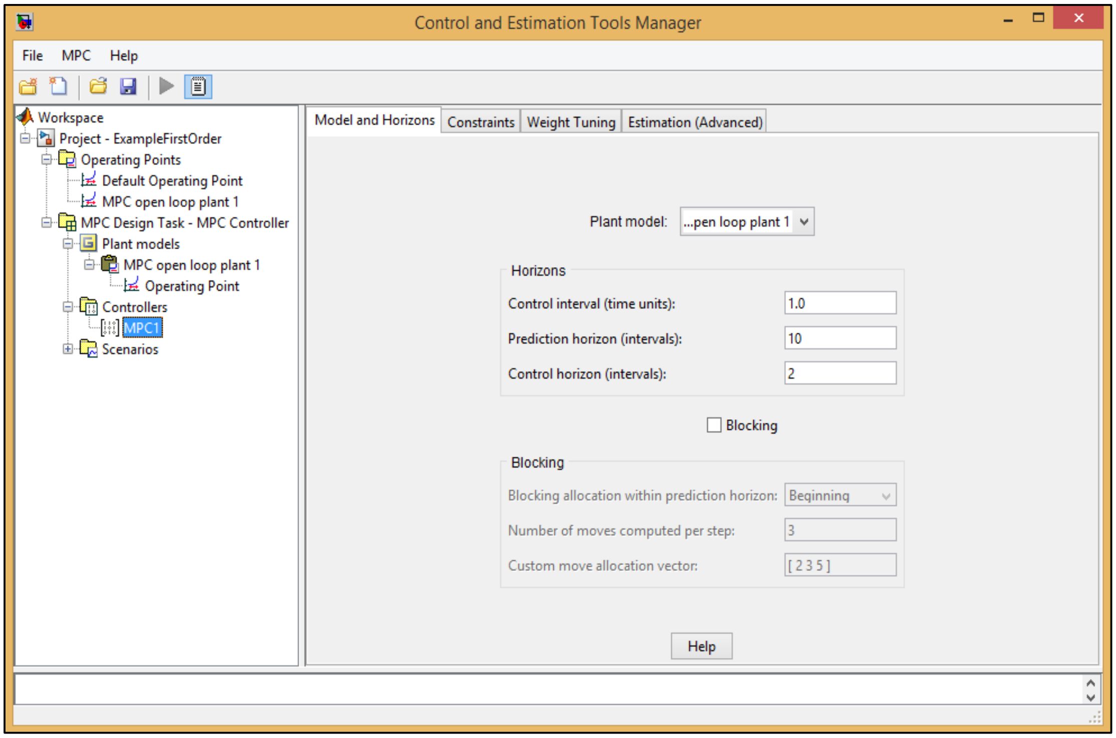
because this is just a 1x1 system, leave the default values for manipulated and measured variables, however for the sample time choose a value of 1.0 (same value as the integration step size).

After clicking OK, the MPC block will linearize the system and compute operating points, this phase can take a few seconds or a couple of minutes depending on how complex is the system. Once the linearization is done, the following window should appear:



This window features a diagram providing the number of MVs, measured outputs, measured disturbances (for a feedforward approach), among others. One can also define nominal values for input and output signals and its properties, such as name, description and units; this first window is basically to check whether the provided settings in the block diagram are correct or not, and primarily characterize the system. After feeling done with this tab, one can advance to the controller parameters.

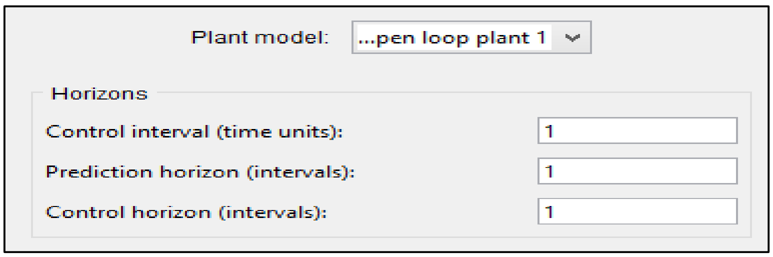
Click on “Controllers” tab and then on “MPC1”; the following window should become visible:



Here one can set values for basic parameters within a MPC controller, such as prediction and control horizons. The control designer is also able to define constraints for manipulated and CVs besides input and output weights; it can be affirmed that all the turning work is done in this tab, hence one should feel encouraged to further explore the effect of such parameters when working with any control design and simulation.

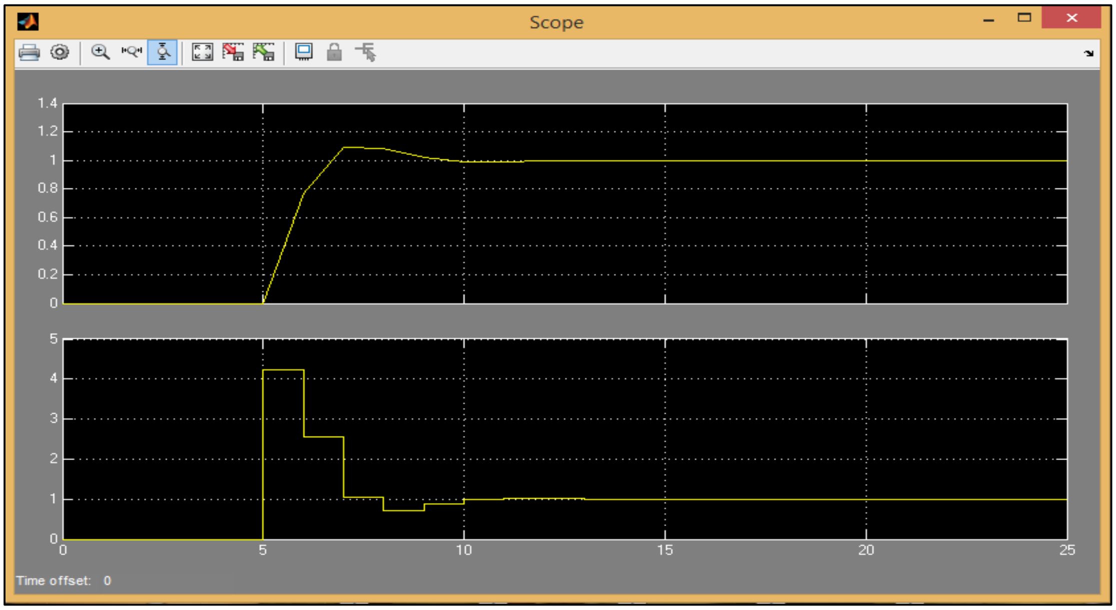
3.2.2. First Order Process: Effect of Prediction horizon P

In order to explore the effect of prediction horizon in this example, let's set the sample time as 1.0 (same as the integration step size) and a control horizon of 1 move. For a first simulation the value of 1.0 time unit for the prediction horizon is suggested, so the controller will predict the behaviour of the system along one sample time and perform the control move calculation in the same step.

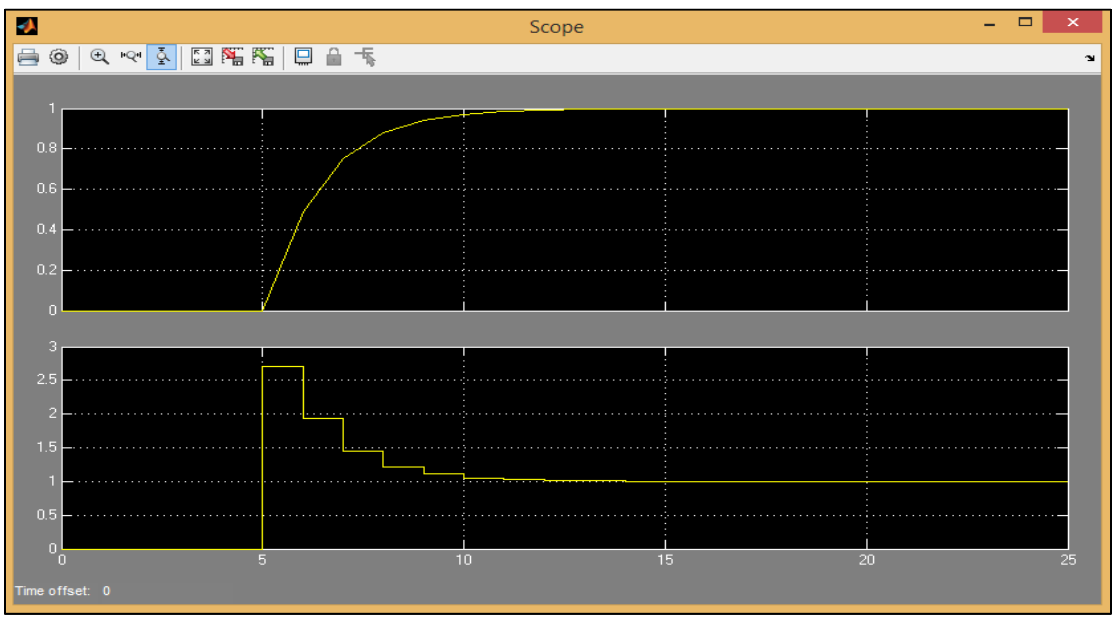


After setting values above, the controller must be exported to the workspace; click on the "Controller" tab and then on "Export". One can provide a name for the controller if necessary, then click again on "Export".

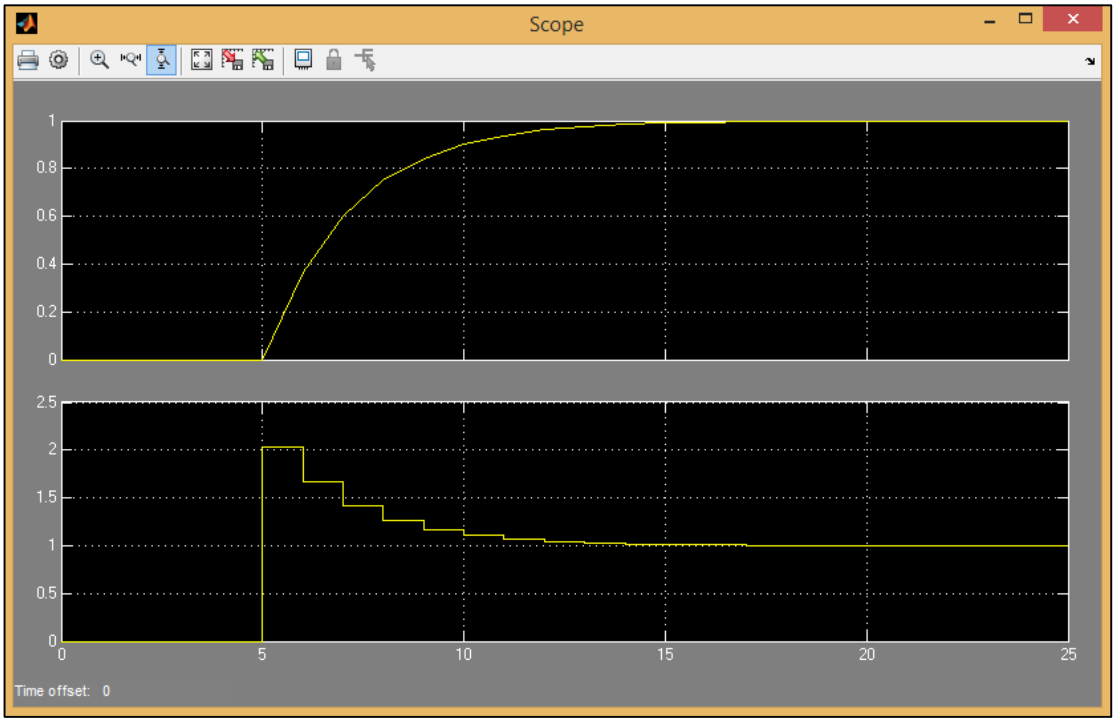
To perform a unitary step change over the CV, double click on the "Step" block and choose its final value as 1.0, leaving the initial value as zero; also, you can set the time instant in which this step change will be applied on the "Step time" bar. After performing these procedures, click "Ok". The results from this first controller design for such unitary setpoint change are as follows:



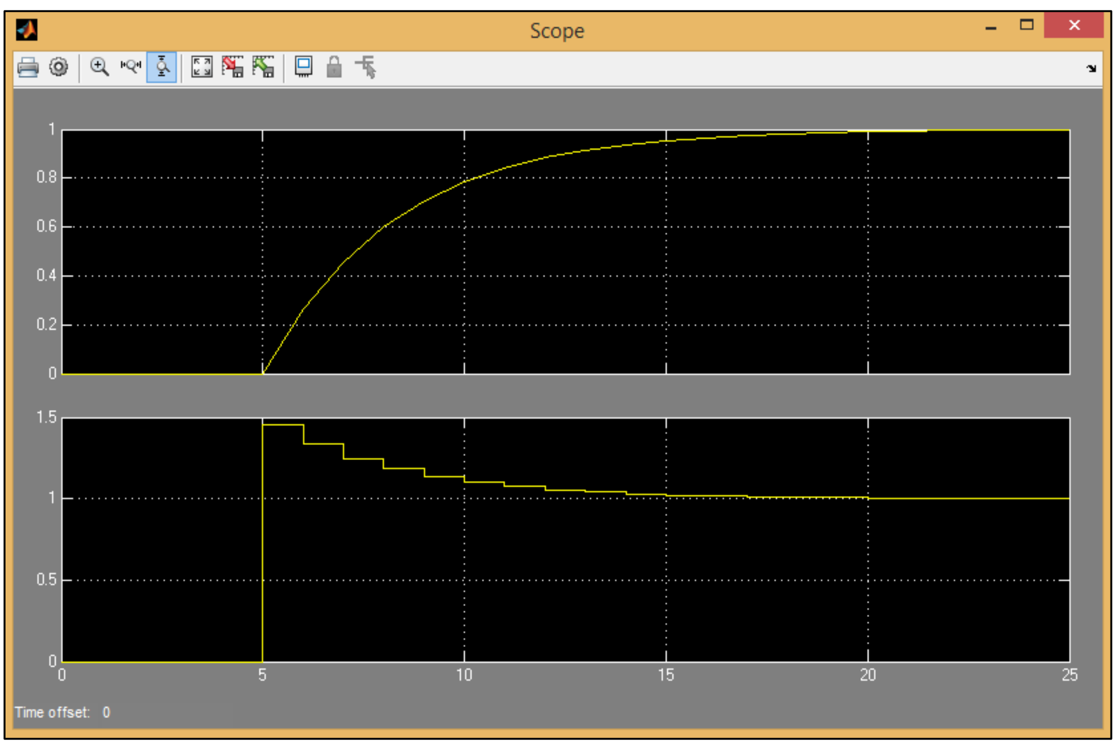
Essentially, the closed-loop response can be improved by increasing the prediction horizon; by changing the value of P to 3, the following response is obtained (after changing any parameter within the "Controller" tab, remember to export the controller again to the workspace for there is not a link between MPC toolbox and the workspace to provide a simultaneous update of such values):



For a prediction horizon of 5:



And finally for a prediction horizon of 10:

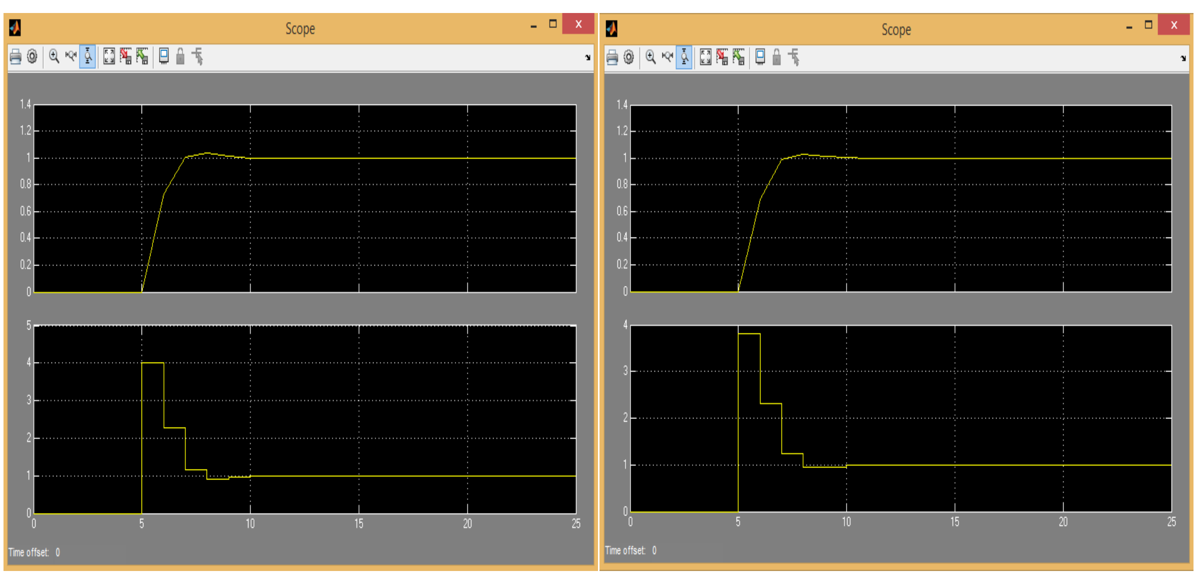


It can be noticed that, as the prediction horizon is increased, the magnitude of the control moves becomes smaller due to the optimization routine which is embedded in the controller; moreover, the system is smoothly tracked towards its setpoint with less variation. The reader should feel encouraged to try more substantial values for P and check the effect over both MV and CV, as well for system stability.

3.2.3. First Order Process: Effect of Control horizon M

The outcome of varying the prediction horizon in a MPC simulation was briefly addressed above and the reader should feel introduced to how to start designing this type of controller. The control objective is to calculate a set of control moves (MV changes) which make the corrected predictions as close to a reference trajectory as possible. The next step is to evaluate the effect of the control horizon M, which is related to the number of control moves calculated per each sample time.

For this specific example, after trying few different values for M, the reader should notice that the control horizon has a tiny effect over this type of system, and distinct values of M lead to practically the same closed-loop response. The following figures show the results for a controller with prediction horizon of 5 and control horizons of 2 and 4, respectively.



In some cases, the prediction horizon has a more significant effect in the closed-loop response performance than the control horizon; for more complex systems, such as multivariable ones, both P and M are generally equally relevant when designing a controller, thus one should feel inspired to spend more time on trying to find a successful combination of these parameters to obtain a satisfactory control performance.

3.3. General instructions to implement MPC

Two basic properties of any system are not independent: model-length and sample-time. The model length must be chosen as approximately the process settling time, which is time required for the system to reach a new steady state after a step input change. Generally, one should set the sample time as one tenth the dominant time constant, so the model length is roughly the settling time of the process.

Prediction and control horizons also differ in length; for most of the cases, prediction horizon is selected to be greater in value than the control horizon. This is notably true if the there is no control weighting over inputs or outputs. Usually, if the prediction horizon is longer than the control horizon, the control system is less sensitive to model error. As the control horizon becomes more substantial, the control moves tend to become more aggressive and a smaller weight on the input is needed to soften the control moves.

The following procedures summarize the steps involved in implementing MPC on any process:

- Develop a discrete step response model with length N and sample time Δt;
- Specify the prediction and control horizons in such a way that: $N \geq P \geq M$;
- If a dominant time delay exists in the process, the prediction horizon must be longer or equal to the time delay, otherwise the close-loop response may become unstable;

- Specify weighting w on the control action;
- Convert all variables to deviation form.

4. Case Study Problems

4.1. Case study #1: Single variable control – Continuous Heat Transfer System (CHTS)

Consider the Continuous Heat Transfer System composed of one shell-tube heat exchanger. Hot glycol circulates through the shell (which is assumed to be perfectly mixed), and then heat flow between hot media and tube increases the energy amount inside the tube, in which flows water. The following scheme depicts such heat transfer unit:

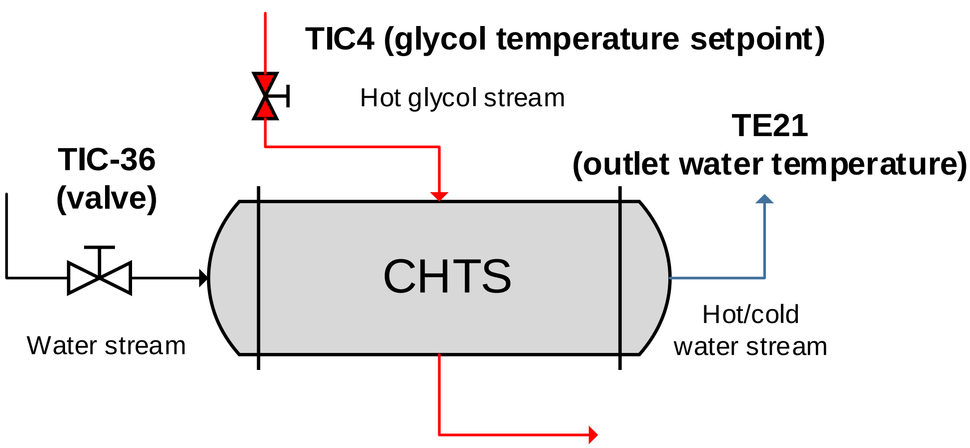


Figure 2 - Continuous Heat Transfer System

The objective is to control outlet water temperature (TE21) by manipulating water flow rate through TIC36 valve opening, thus consisting on a 1x1 system; furthermore, both servo (outlet water temperature setpoint change) and regulation problems (disturbance rejection) are able to be performed using MPC. In sum, the variables in a CHTS unit are the following:

- Manipulated variable: TIC36 (valve);
- Controlled variable: TE21 (outlet water temperature);

- Disturbance variable: TIC4 (glycol temperature setpoint).

The transfer function model which relates MV and CV is a first-order plus delay type:

$$\frac{TE21(s)}{TIC36(s)} = \frac{0.7105}{5.966s+1}e^{-5.64s} \tag{x}$$

And the one relating DV and CV is third-order plus delay and numerator dynamics type:

$$\frac{TE21(s)}{TIC4(s)} = \frac{2.516s+0.2605}{3145s^3+426.6s^2+62.4s+1}e^{-24s} \tag{xi}$$

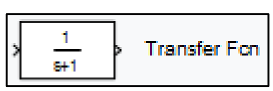
Steady state values for each variable are listed below in Table 1:

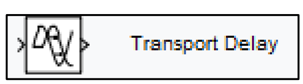
Table 1 – Steady state values for CHTS

TIC36 (percentage of closure)	**TIC4 (°C)**	**TE21 (°C)**
75.02	60.0	16.45

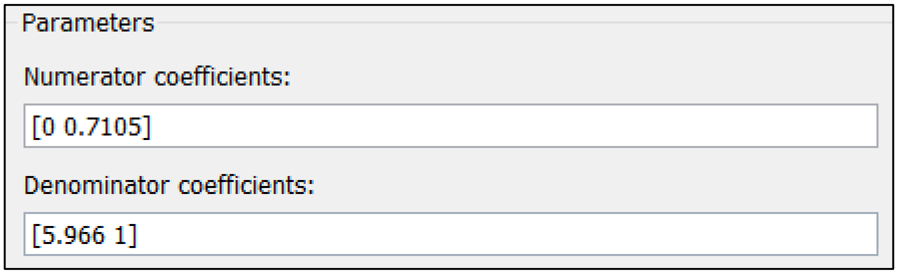
4.1.1. CHTS – servo problem (changes on TE21 setpoint)

This topic will discuss how to implement equation (x) in a Simulink block diagram in order to simulate step changes over the outlet water temperature (TE21); later on, equation (xi) will be also included to simulate disturbances in the system as a setpoint change over the hot glycol stream. To introduce the plant in Simulink, only two blocks will be necessary:

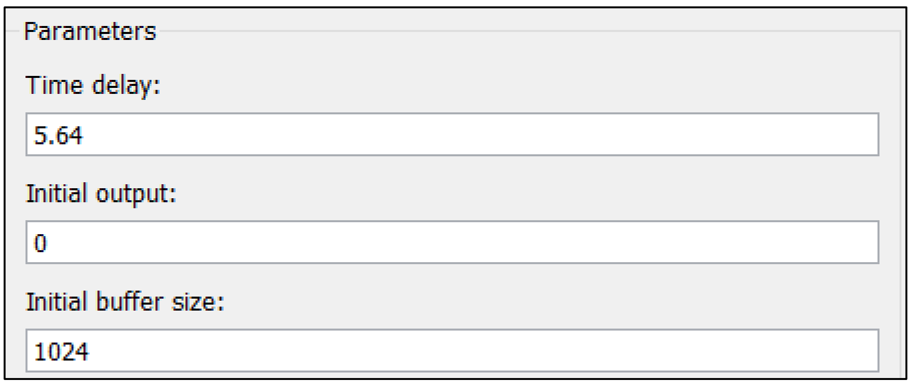
- Transfer function block 

- Transport delay block 

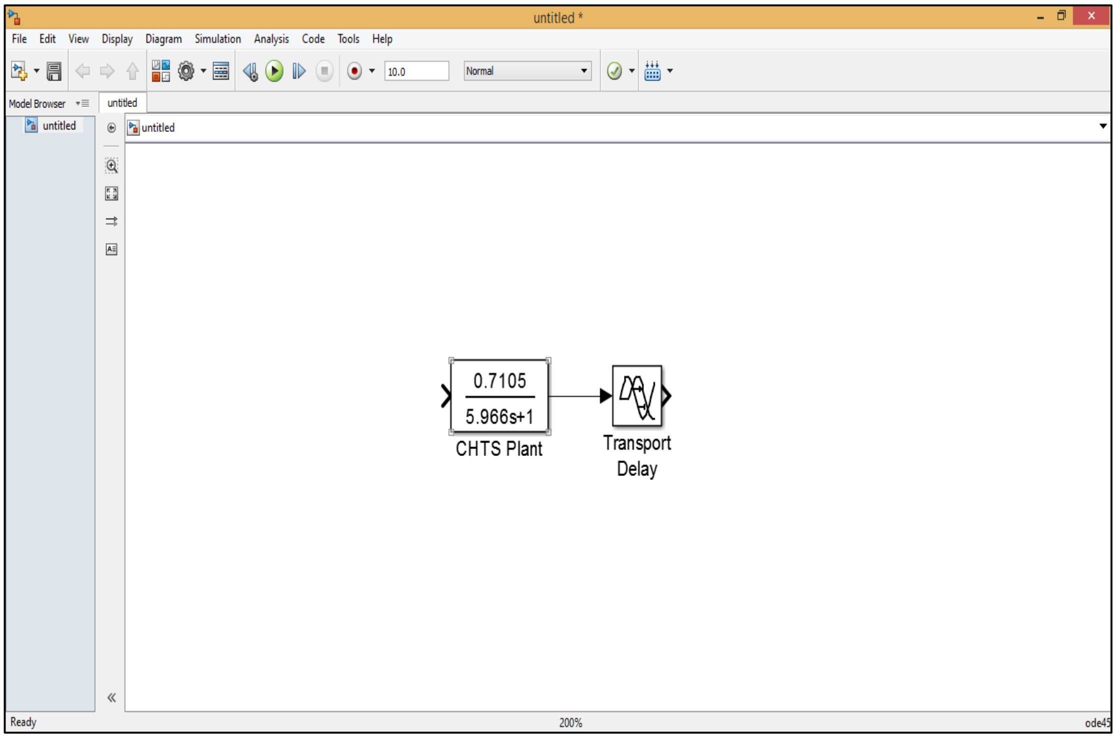
To create a new Model in Simulink, review the steps on topic 3.2.1. and do not forget to set the simulation parameters by using the “Simulation” menu. For this case, we will use a fixed step integration step-size of 1.0 second and ode4 (Runge-Kutta) solver; one is welcomed to test other step-size values or solvers, however let us stick with these configuration for the time being. After creating a blank model, insert the blocks previously indicated and connect then with a straight arrow. Double click on “Transfer Fcn” block and set poles and zeros as follows and click “Apply”:



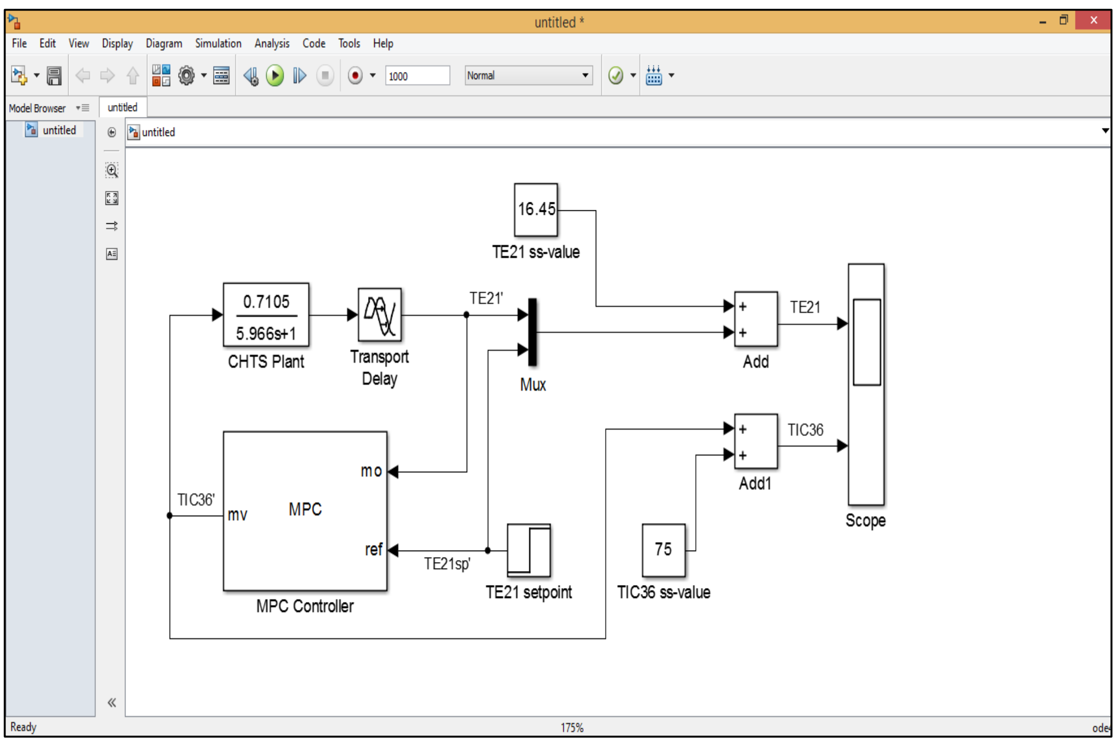
This will properly define equation (x); the next step is to set the time delay. To do so, double click on “Transport delay” block and in the Time delay bar, set its value as 5.64, leaving the other setting as their default values. Finally, click “Apply” and then “Ok”.



The diagram should look like the following:



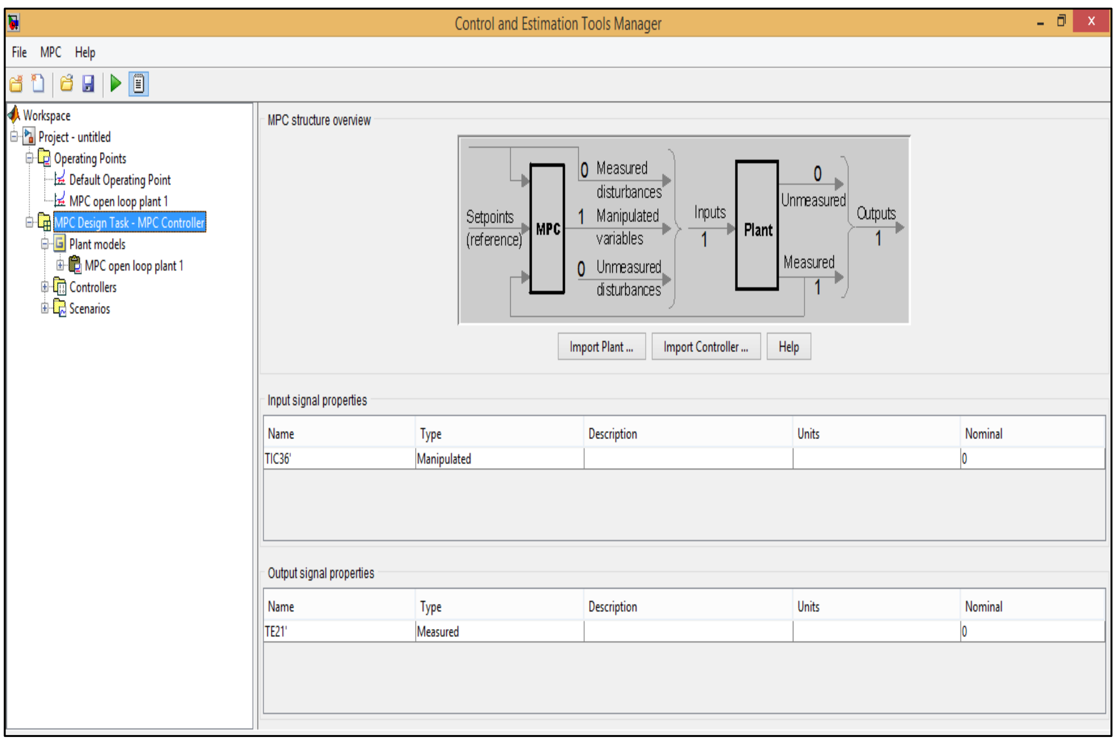
Next step is to connect the MPC block and start to design a controller. The final diagram which will be used for this example is demonstrated below:



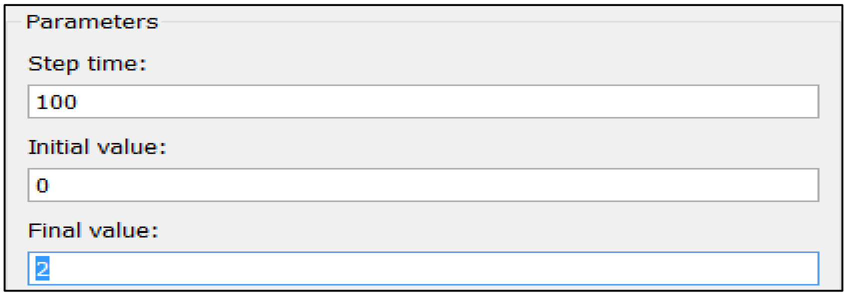
To designate signal names, right-click on the arrow which corresponds to a signal that you want to name and then click on “Properties”; on the “Signal name” bar, type the desired name and then click “Apply” and “Ok”.

Before starting to design the controller, it is important to reassure that the system is primarily in its steady state; double click on “TE21 setpoint” step block and set both initial and final values as zero, and the Step time as 100. Double click on the “MPC Controller” block and then click on “Design”. A small window will show up asking the number of MVs and CVs, as well as the sample time; leave the default values of 1 and then click “Ok”.

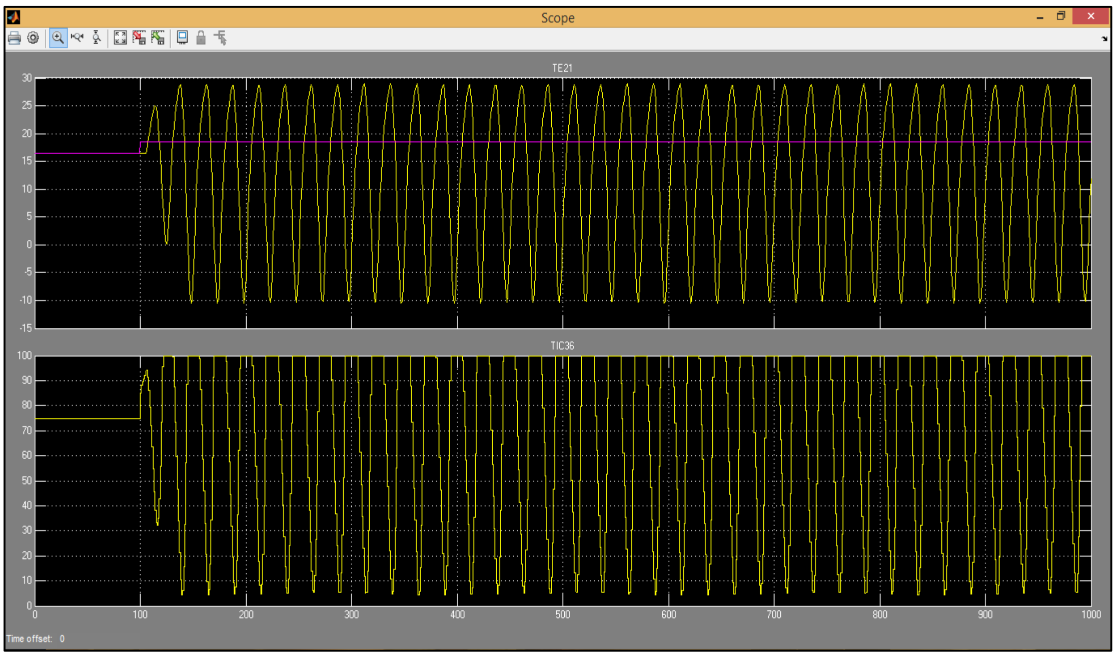
The following window should become visible:



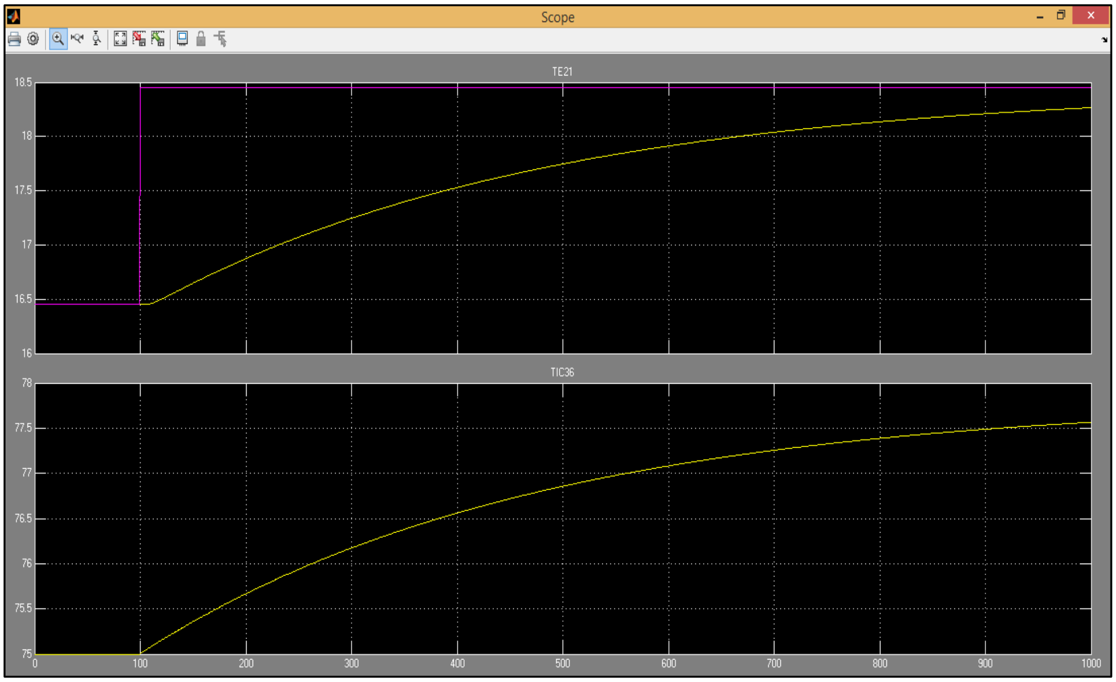
We are ready to start our controller design. First, we have to set prediction and control horizons as one and check if a stable close-loop response is obtained. Click on “Controllers” tab on the left-side menu and then on “MPC1”. On “Model and Horizons” tab, leave the Control interval as 1.0 and set prediction and control horizons as 1; also, click on “Constraints” and set the minimum value as -75 and the maximum as 25 (remember that all the variables within MPC are in deviation form, thus -75 corresponds to zero opening and 25 corresponds to total opening). After all these configurations are done, export the controller to MATLAB's workspace and run the simulation with a TE21 setpoint change of 2°C.



The following closed-loop response is obtained:



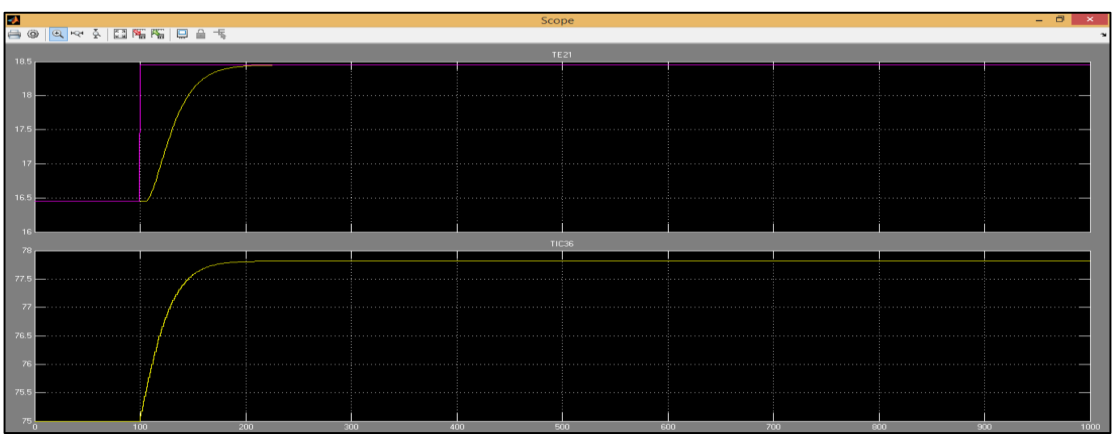
It is clear that the system became unstable with such parameters by being too fast, still this situation can be easily fixed by looking at the Overall Weight; on the "Weight tuning" tab within "MPC1", the default value for Overall Weight is 0.8 which coincides with a faster response. To turn the system slower and more robust, set its value to 0.4 (or lower values) and export the controller. The response for this case should be similar to the following:



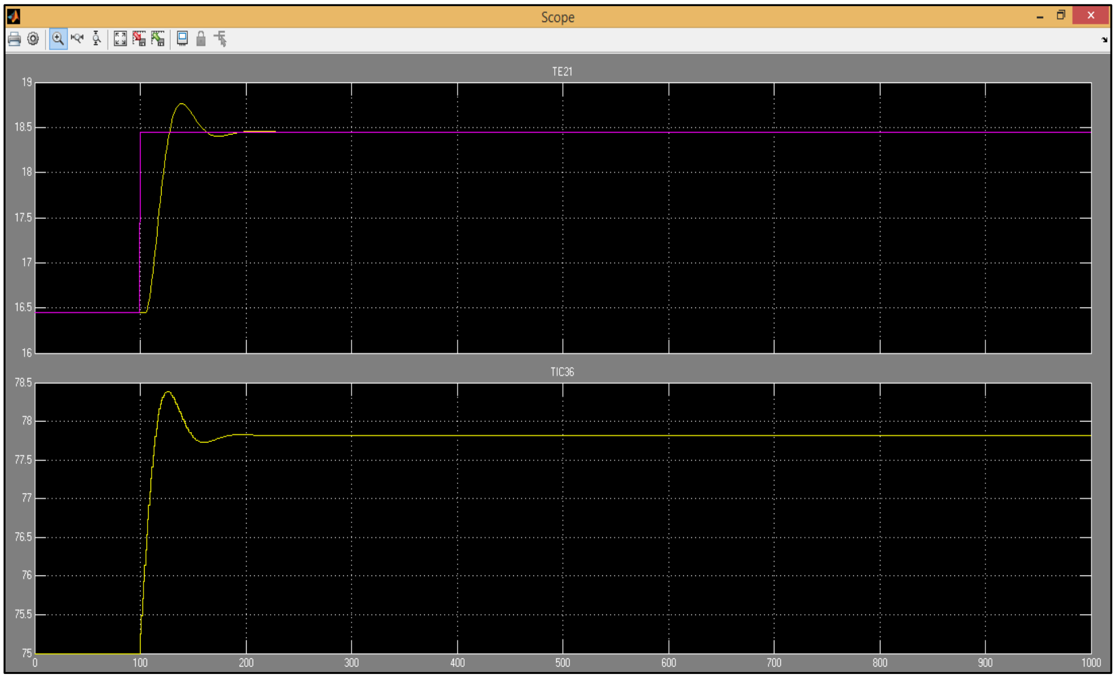
By a simple change over the speed of the system, it was able to obtain a stable yet sluggish closed-loop response.

4.1.2. CHTS – Effect of prediction and control horizons on servo problem

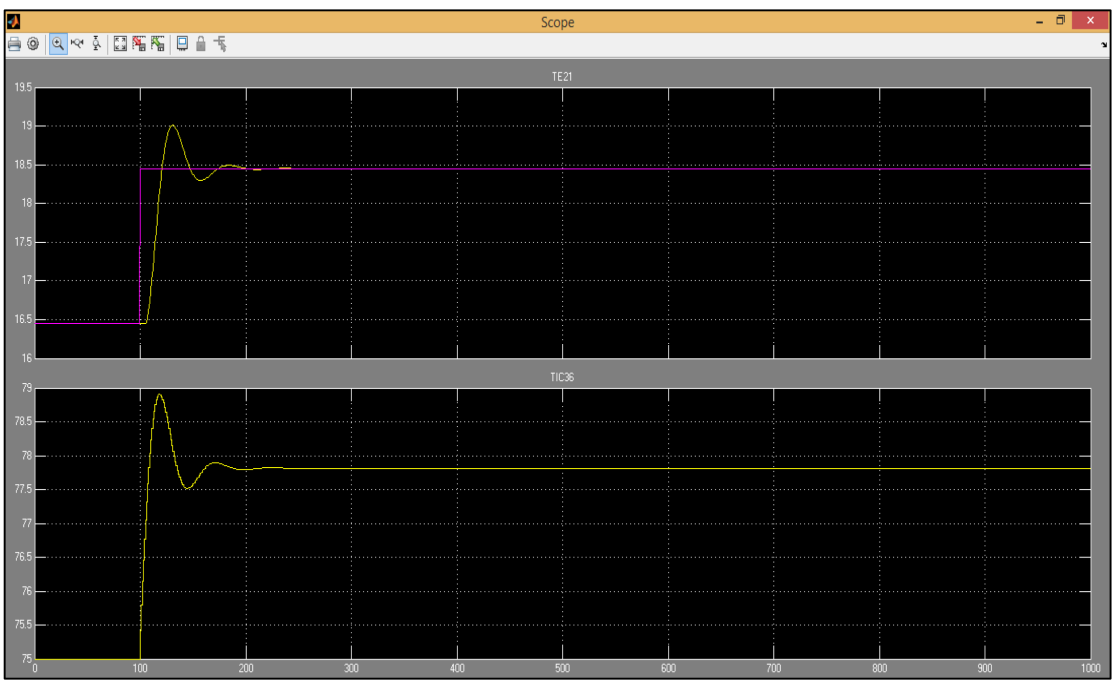
It is possible to turn the system response faster by adjusting accordingly the value for prediction horizon. Let us try several different values and choose one which fits a middle ground between robustness and speed. First, try a prediction horizon of 5 seconds by going to "MPC1" → "Model and horizons" tab; after exporting the controller again and running the system, the response should look similar to the next:



According to the previous graph, a better response for a TE21 setpoint change was obtained featuring a settling time of around 120 seconds and no oscillations over both MV and CV; it is possible to obtain a faster response yet oscillatory when the prediction horizon is set as 10 seconds or huger values:

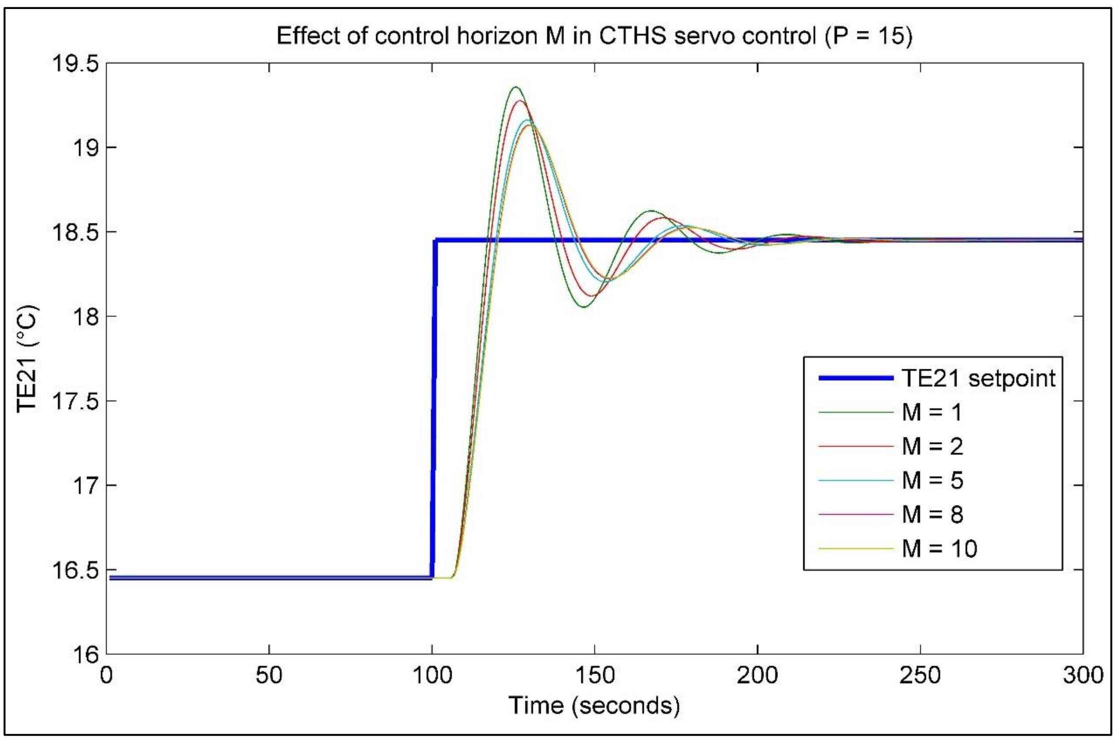


and for a prediction horizon of 15 seconds:



Again, one is encouraged to try different values for the prediction horizon and further explore system's stability by increasing P.

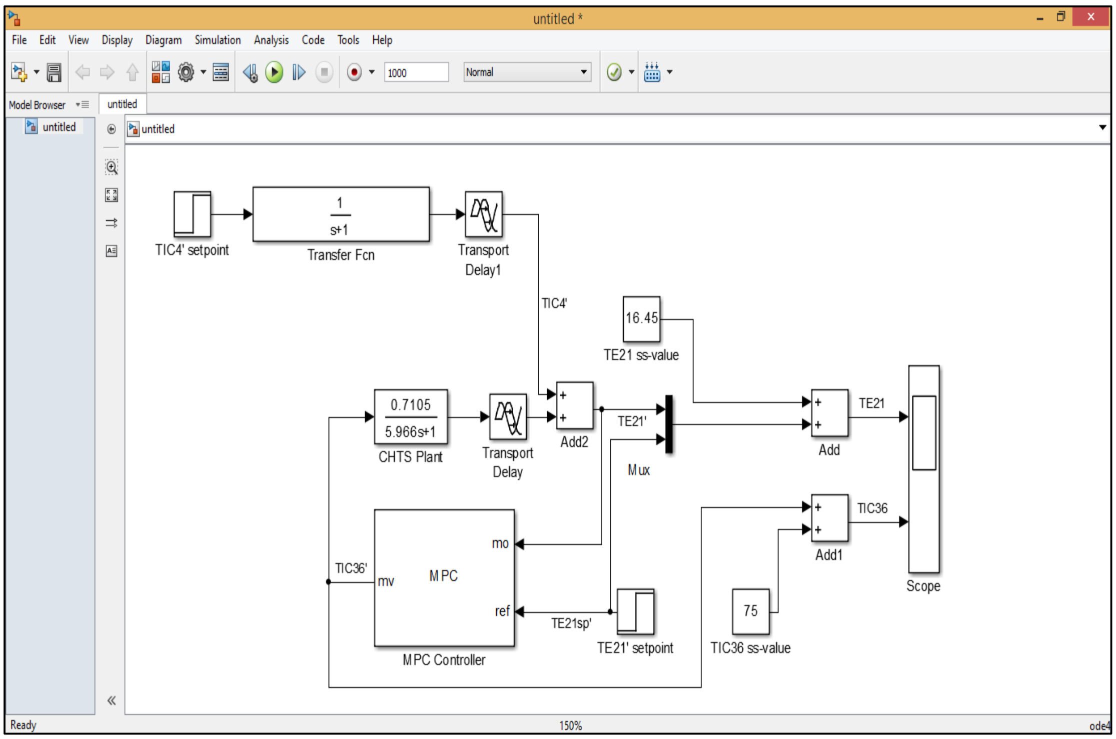
In the same way as example 1, the control horizon M has a very slight effect in the system, thus being the prediction horizon the main parameter in this case. Of course, the "Rate weight" over TIC36 in the "Weight tuning" tab has a significant relevance when tuning the controller for a certain desired response, and one is welcomed to explore the effect of such parameter by increasing (slower response) and decreasing (faster response) its value. The effect of weight tuning will be better explored when simulating a multivariable system.



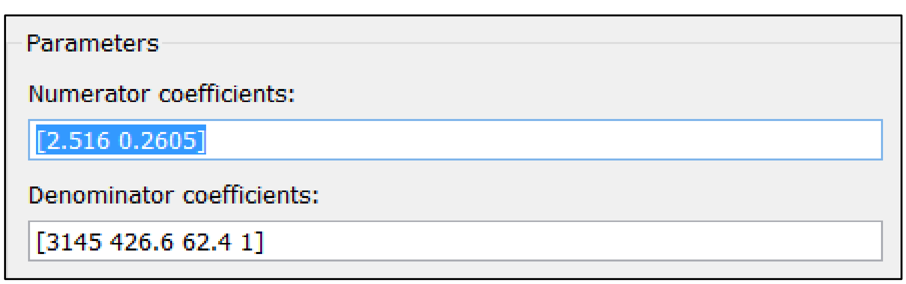
4.1.3. CHTS – regulation problem (TIC4 disturbance rejection)

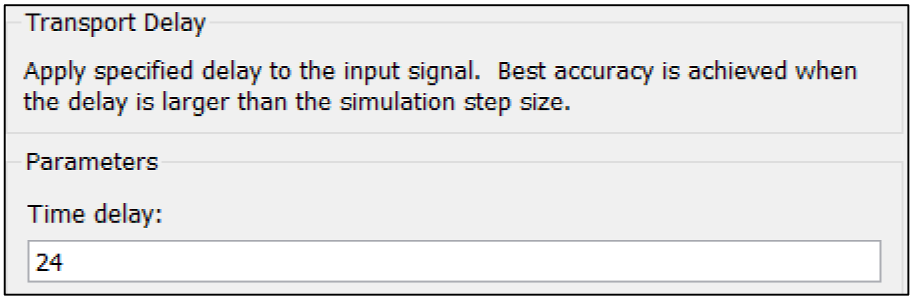
In order to include a disturbance model into CHTS, it is necessary to consider equation (xi) during the simulation, which is the transfer function model relating TE21 (outlet temperature) to TIC4 setpoint (hot glycol temperature). Usually, glycol temperature is kept constant during the process by another heat transfer apparatus which is connected to the heat exchanger shell outlet, however in case of failure in this apparatus the glycol temperature may be altered, leading the system to a new steady state. This topic will simulate the disturbance rejection in the system as a step change over TIC4, again exploring the effects of prediction and control horizons.

To insert equation (xi) add another Transfer Fcn, Transport Delay and Steps blocks into the diagram as demonstrated below:

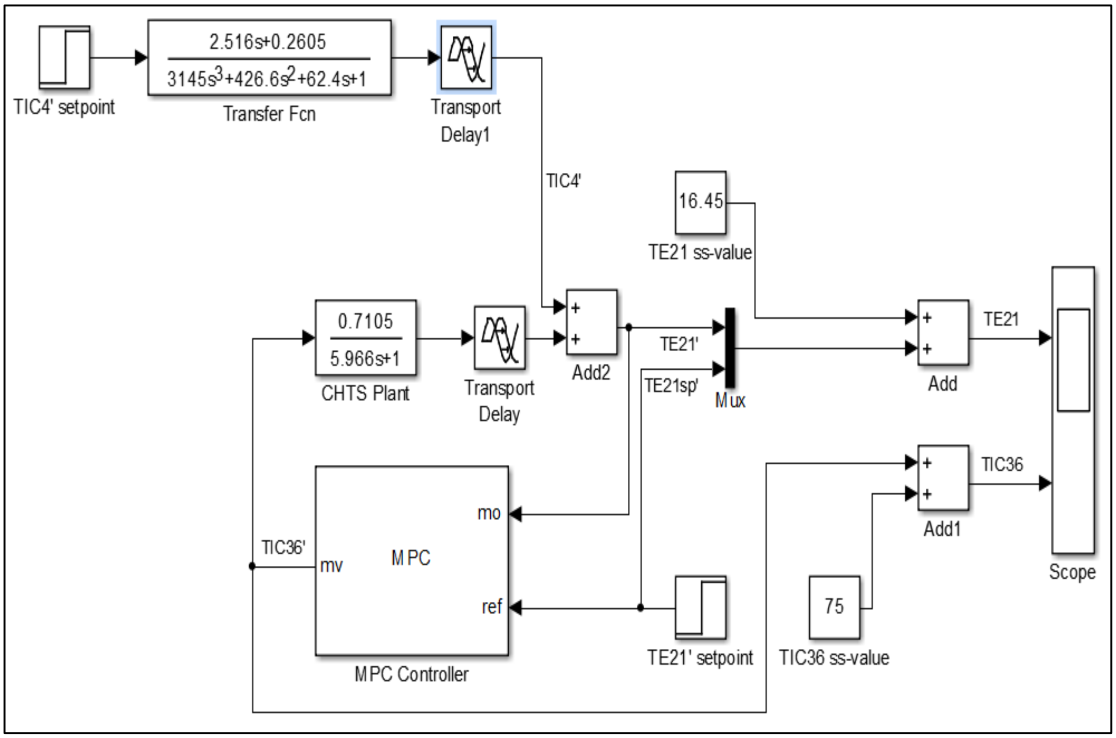


To properly define equation (xi) in Transfer Fcn block along with the Transport Delay, double click on such blocks and define its coefficients as showed below:

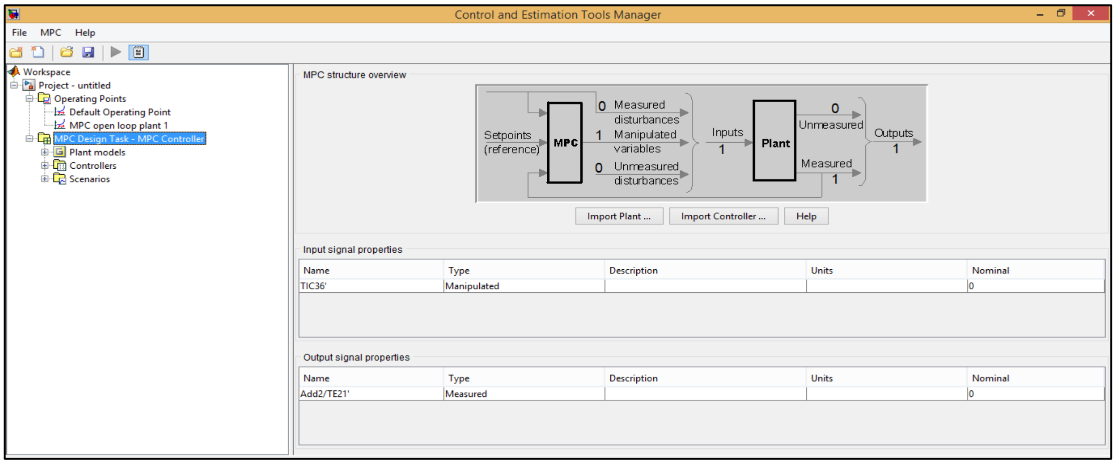




The final diagram including the disturbance model has the following appearance:



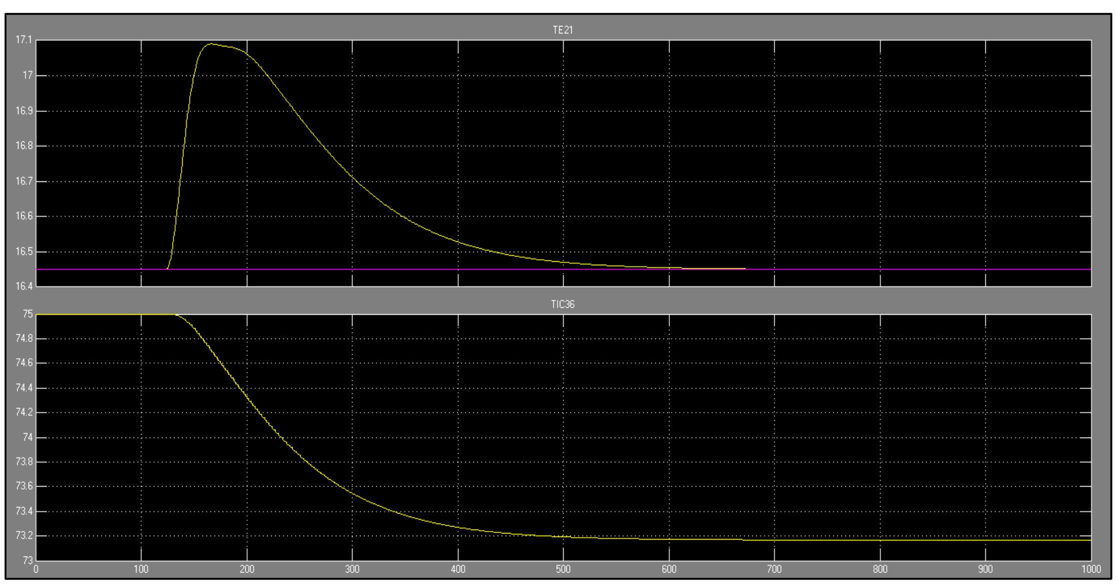
Again, prior to designing the controller, it is important to reassure that the system is primarily in its steady state; double click on “TE21’ setpoint” and “TIC4’ setpoint” step blocks and set both initial and final values as zero, and the Step time as 0 and 100, respectively. Double click on the “MPC Controller” block and then click on “Design”. A small window will show up asking the number of MV’s and CV’s, as well as the sample time; leave the default values of 1 and click “Ok”.



As for the servo problem, the configuration listed below must be set before exporting the controller to MATLAB workspace:

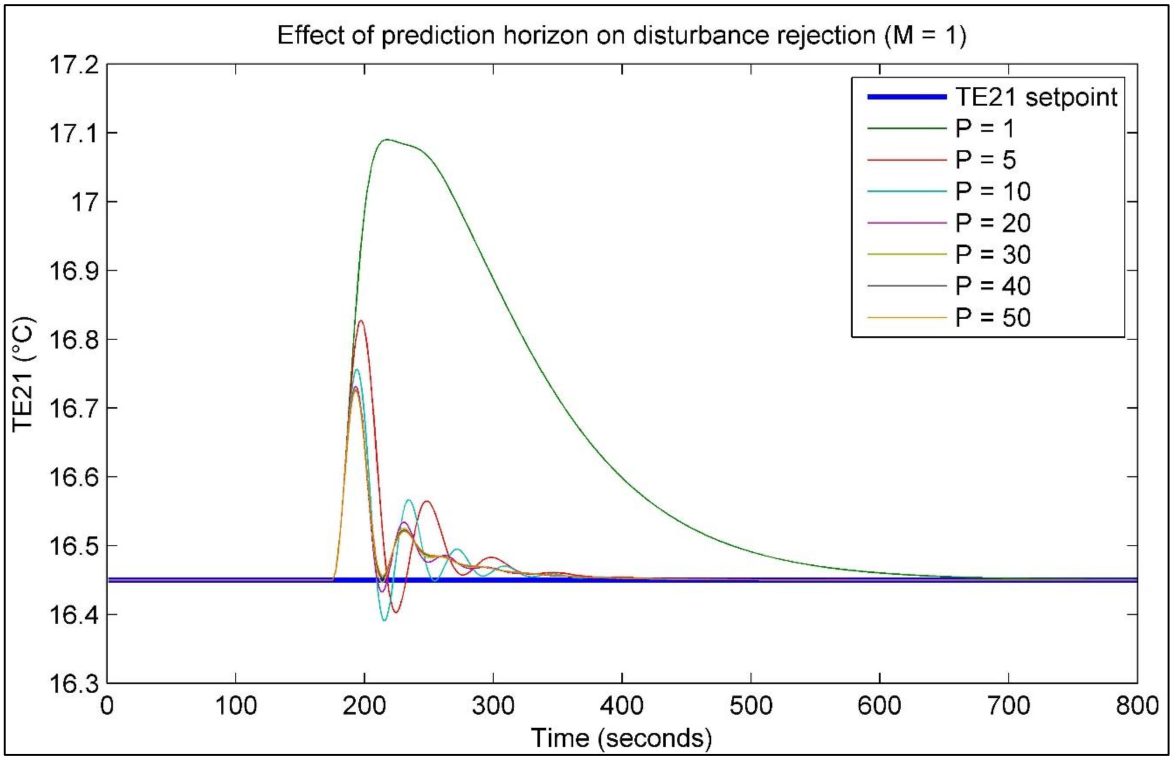
- On “Model and Horizons” tab, leave the Control interval as 1.0 and set prediction and control horizons as 1 move;
- Click on “Constraints” and set the minimum value as -75 and the maximum as 25 for the MV;
- Go for “Weight tuning” tab and slide the Overall Weight bar until a value of 0.4 is reached.

Once these steps are done, export the controller and run the Simulink diagram; the close-loop response for a setpoint change over TIC4 of 5°C should look similar to the following:



4.1.4. CHTS – Effect of prediction and control horizons on regulation problem

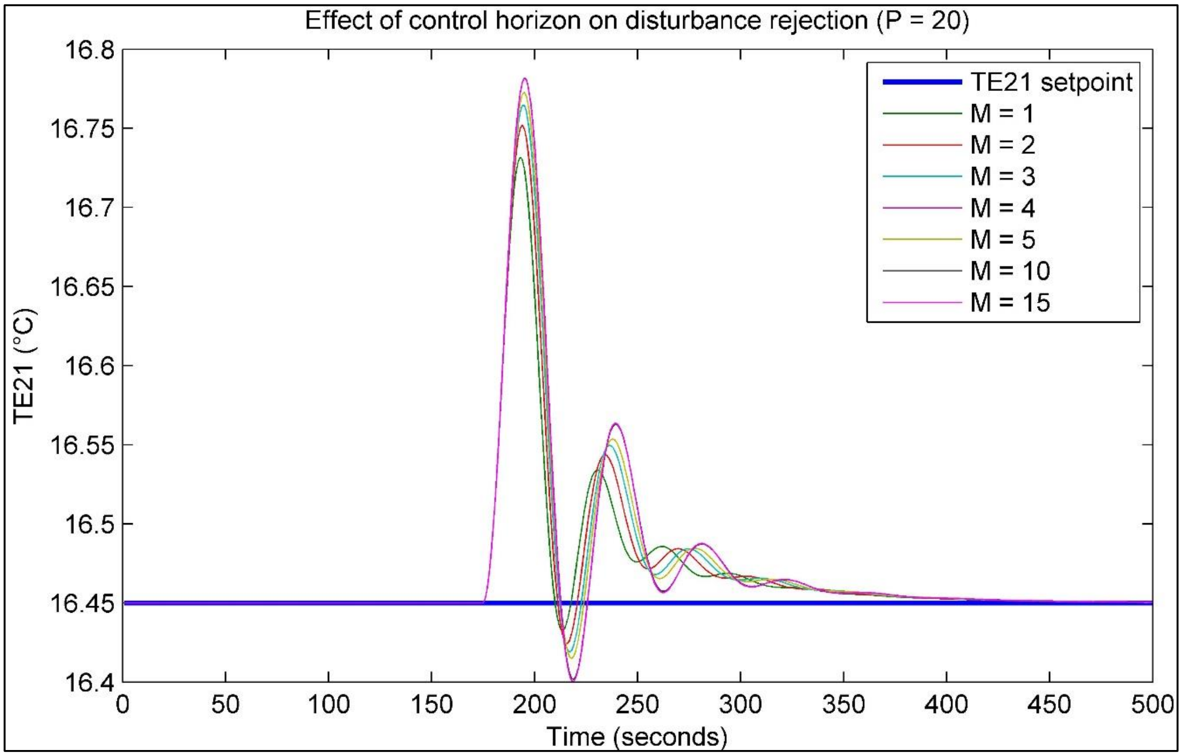
The next figure depicts the influence of prediction horizon when rejecting a disturbance in the system (hot glycol temperature setpoint); values within a range of 50 seconds were tested and results are showed below:



To apply changes in any parameter when designing a controller, make sure to re-export the controller to MATLAB’s workspace so that Simulink will be able to update the controller.

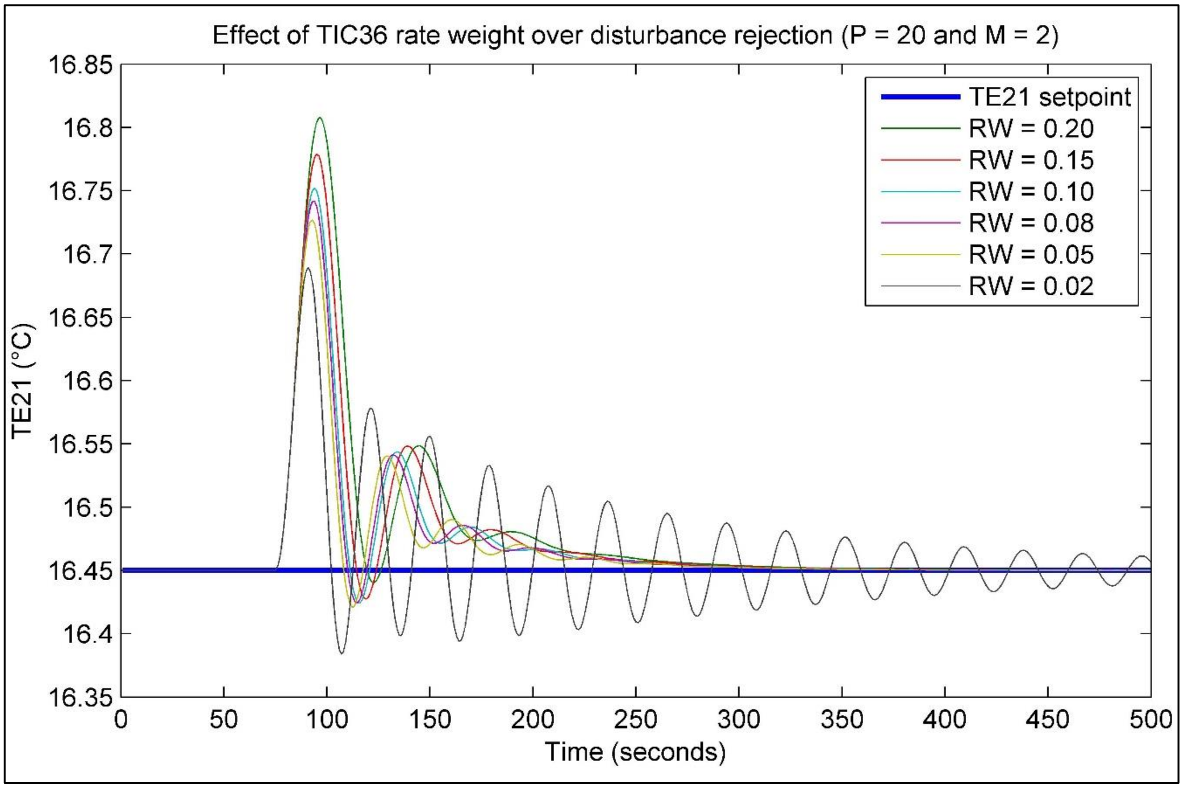
It is clear that a prediction horizon greater than 20 seconds undergoes minimal improvements to TE21 setpoint tracking; this sort of behavior is common in MPC controllers both to prediction and control horizons, however such “steady” values varies from system to system and one should look for these values in order to assess sensibility of a plant regarding predictive parameters.

Let’s choose a value of 20 seconds for the prediction horizon and start to explore the role of the control horizon when a disturbance input are introduced in CHTS. The next diagram demonstrates control horizon's influence when a +5°C step change in TIC4 setpoint is detected; values within a range of 15 control moves were tested (remember to set $M \leq P$):



4.1.5. CHTS – Effect of input rate weight on regulation problem

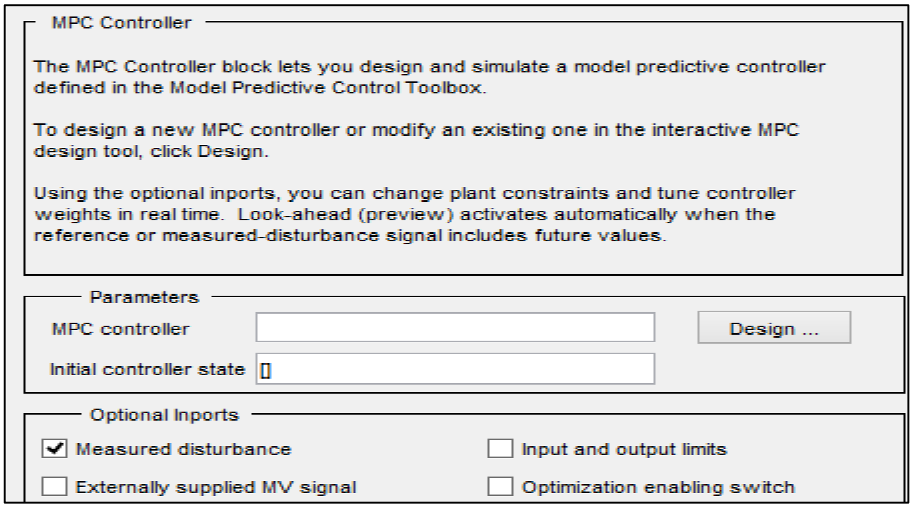
As previously discussed in topic 3.3, increasing the control horizon turns the controller to be more aggressive, thus the control moves assume more considerable magnitudes when dealing with MV's, undergoing considerable overshoots and undershoots; a simple way to compensate such variations is to introduce a smaller weight on the input by going to "Weight tuning" tab and try rate weight values less than 0.1. For instance, you may choose a control horizon of 2 moves and test rate weights of from 0.2 to 0.01 and check for oscillations in the closed-loop response as well as its stability. The following figure shows such sensitivity analysis:



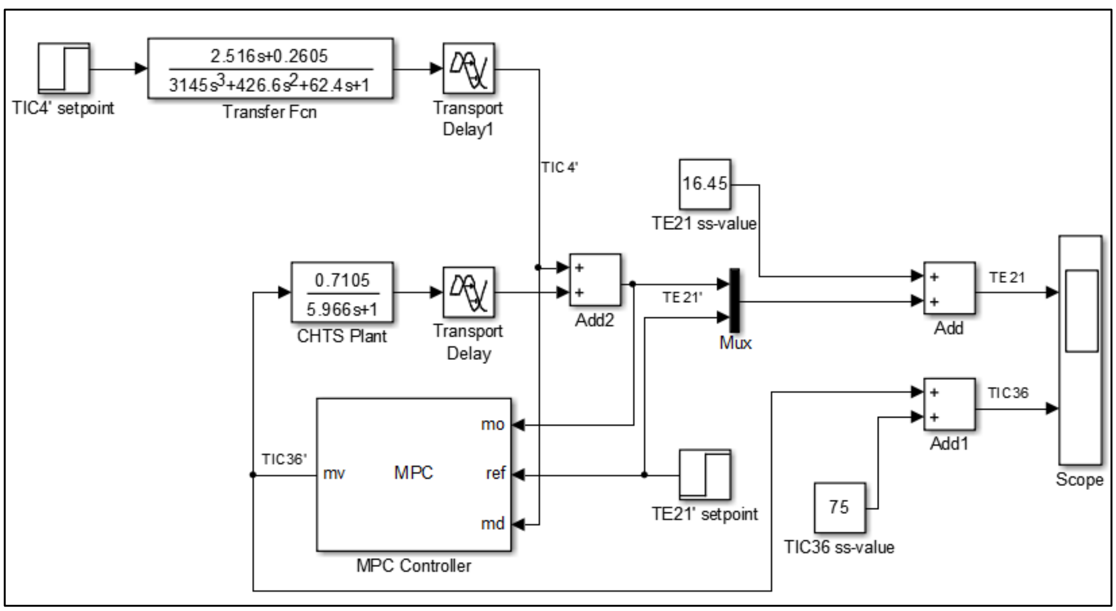
As can be noticed that, as the rate weight is decreased, the control moves (and so the controlled variable behaviour) have a role of diminishing overshoots, yet the system becomes more oscillatory until the point of turning completely unstable, which is case for input rate weight values less than 0.02.

4.1.6. CHTS – adding a feedforward controller routine

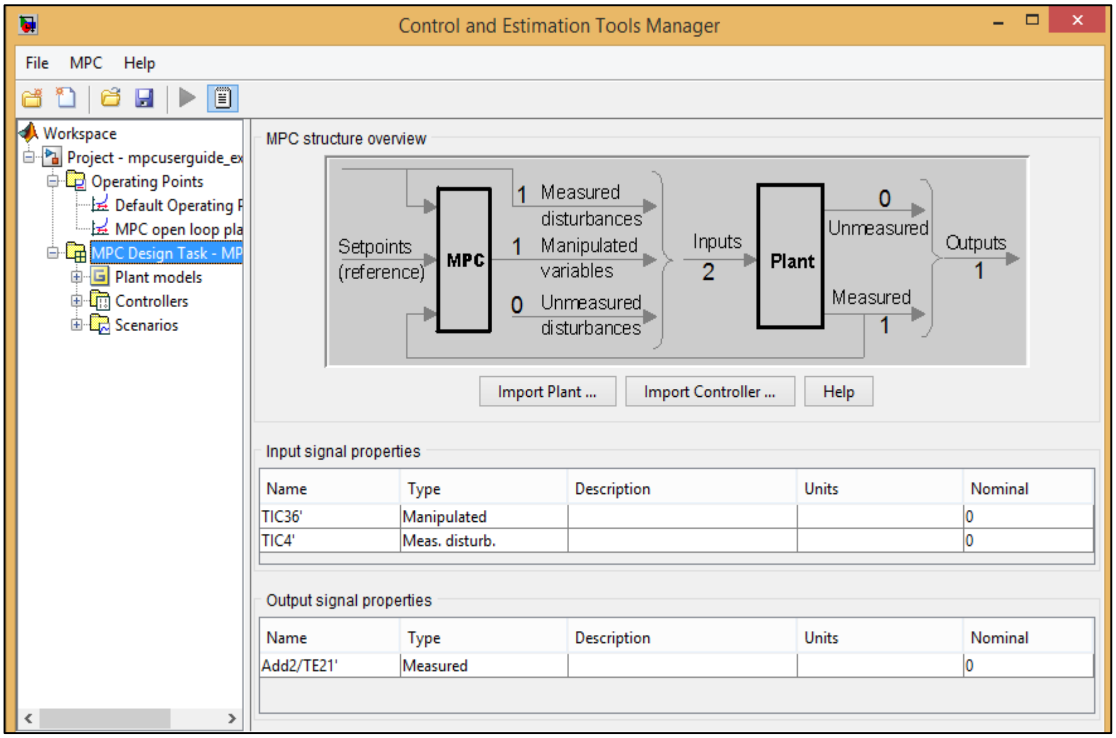
The MPC block features a feedforward controller add-on which measures important disturbance variables and take corrective action before they lead the process to a new steady state. To make an effective use of feedforward control, an approximate process model should be available, in this case such model is equation (ix). To activate feedforward control, go back to the original Simulink diagram and double on the MPC block; the following window should become visible:



Check the “Measured disturbance” box and connect TIC4 signal to a new port which becomes available in MPC block:



Then start to design the controller by clicking on “Design”; again start with prediction and horizon values as 1 to check system stability; remember to include constraints (-75 and +25) in the manipulated variable and set the overall weight 0.4. Also check system specification through this window:



The control designer is encouraged to explore the effect of prediction and control horizons over a MPC/feedforward controller by introducing several distinct values for these parameters, as well as the influence of different input rate weights in the system. The final designing task is to properly tune all these parameters inside a MPC controller to obtain a specific closed-loop response within a range determined by input/output constraints.

4.2. Case study #2: Multivariable control – Evaporator system

The objective of this case study is to simulate an evaporator plant and develop a MPC controller for a multivariable system, which consists of an evaporator unit connected to a gas-liquid separator. The system consists of 3 inputs and 3 outputs, defined as follows:

- Manipulated variables:
    - Product flow rate (F2);
    - Steam pressure (P100) – MV#1;
    - Cooling water flow rate (F200) – MV#2;
- Main disturbance variables:
    - Feed flow rate (F1) – DV#1;
    - Feed composition (X1) – DV#2;
    - Feed temperature (T1) – DV#3;
- Controlled variables:
    - Separator level (L2).
    - Product composition (X2) – CV#1;
    - Operating pressure (P2) – CV#2;

The control performance can be evaluated by considering positive and negative disturbances on these three main disturbance variables and checking if the manipulated variables were able to keep the desired outputs by managing their values within a reasonable range.

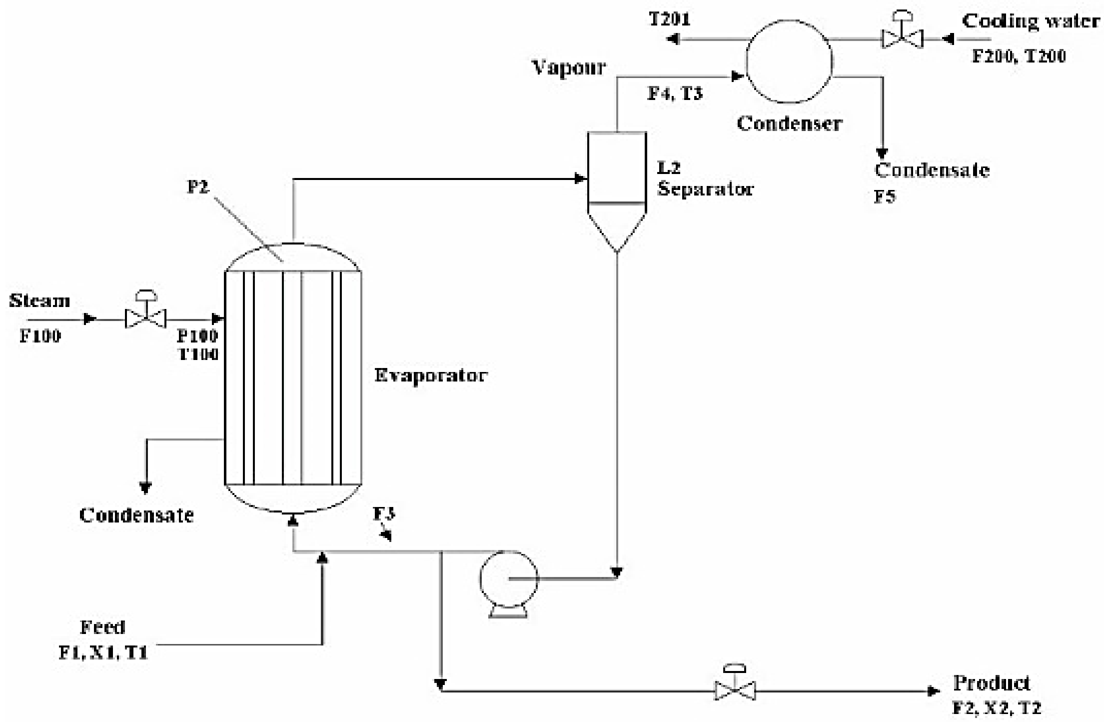


Figure 3 - Simplified process flow diagram of evaporator system

The steady state information for evaporator system is given in Table 2:

Table 2 – Evaporator system steady state values

X2 (%)	P2 (kPa)	P100 (kPa)	F200 (kg/min)	F1 (kg/min)	X1 (%)	T1 (°C)
25.0	50.5053	194.7	208.0	10.0	5.0	40.0

It will be considered that F200 steady state value corresponds approximately to a linear half-opened valve and P100 may vary within a range of more or less its steady state value; thus the constraints for the MV’s are +208/-208 kg/min and +194.7/-194.7 kPa, respectively.

In order to simplify the control problem, it will be assumed that the liquid level inside the evaporator unit is controlled and its value is kept constant through time; as a matter of fact, the transfer functions for this system were obtained with data collected when a level controller (PI-type) was on automatic. Futhermore, the relative gain array for the process indicated that it is more feasible to control Product Composition (X2) by manipulating Steam Pressure (P2), and to control Operating Pressure (P2) by manipulating Cooling Water Flow Rate (F200) in the condenser.

System's dynamics is described by transfer function models below:

$$\frac{X2'(s)}{P100'(s)} = \frac{-0.0001041*s^4-0.005142*s^3-0.1022*s^2+0.0293*s+0.0004008}{s^5+0.6715*s^4+987.1*s^3+149.2*s^2+16.89*s+0.3213}e^{-4*s} \quad \text{(xii)}$$

$$\frac{P2'(s)}{F200'(s)} = \frac{-0.04646}{11.51*s^2+32.38*s+1}e^{-0.5*s} \quad \text{(xiii)}$$

$$\frac{X2'(s)}{F1'(s)} = \frac{0.2111*s-0.04504}{47.73*s^2+13.31*s+1}e^{-3*s} \quad \text{(xiv)}$$

$$\frac{X2'(s)}{X1'(s)} = \frac{213.9*s+3.681}{236*s^2+80.62*s+1}e^{-3.74*s} \quad \text{(xv)}$$

$$\frac{X2'(s)}{T1'(s)} = \frac{0.04059*s+0.0005693}{2014s^3+378.3*s^2+47.09*s+1}e^{-4.78*s} \quad \text{(xvi)}$$

$$\frac{P2'(s)}{F1'(s)} = \frac{1.996}{209.7*s^2+28.97*s+1}e^{-4*s} \quad \text{(xvii)}$$

$$\frac{P2'(s)}{X1'(s)} = \frac{-133.8}{33.81*s+1} \quad \text{(xviii)}$$

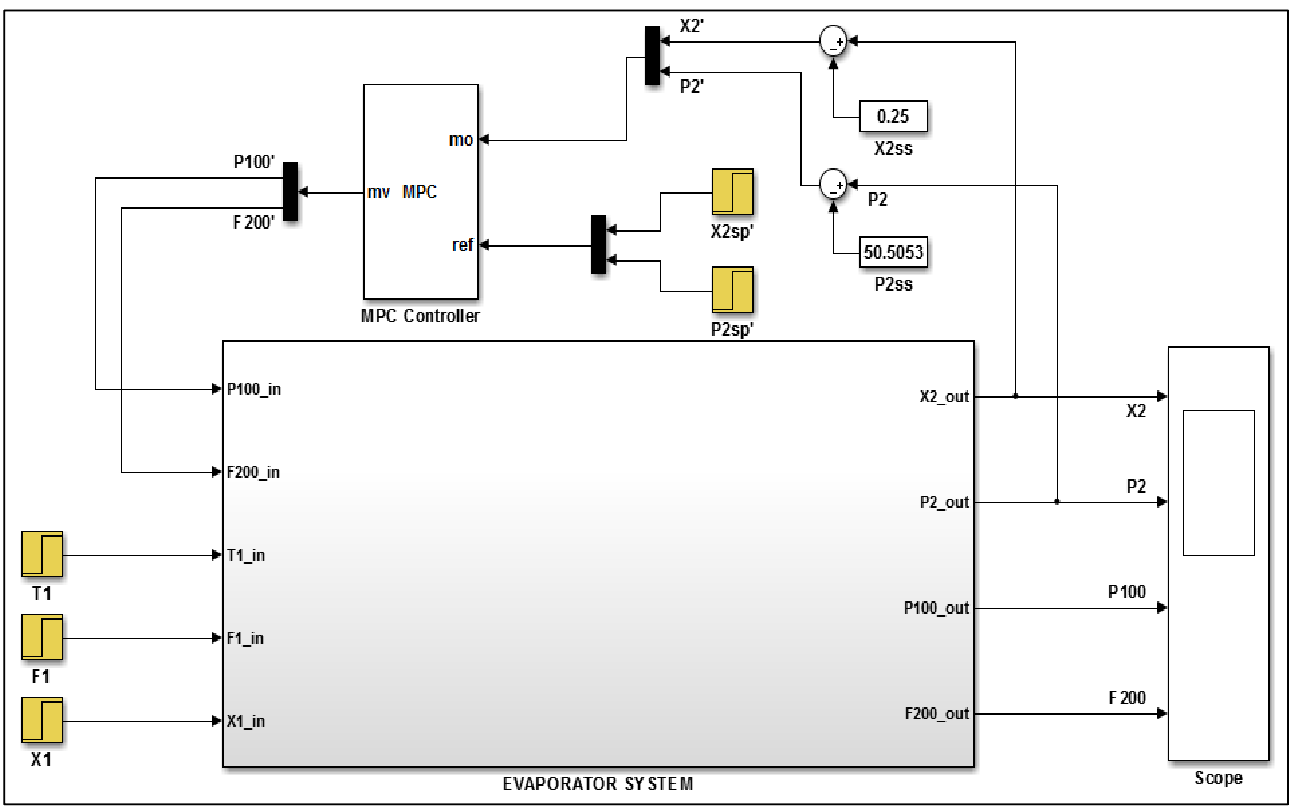
$$\frac{P2'(s)}{T1'(s)} = \frac{0.609*s+0.06062}{552.6*s^3+263.3*s^2+54.92*s+1} \quad \text{(xix)}$$

It is important to emphasize the nonlinear behaviour which is embedded in such equations, or in other words, the gain of each variable is different according to the magnitude of a MV step change. Also, the system may become unstable when distinct step changes are applied in DV's and CV's setpoints; for instance, a controller which has a decent performance when dealing with positive changes over F1 (feed flow rate) may not be suitable for a negative disturbances over the same variable, or even a controller which does a fine job for regulation problems may not be satisfactory when dealing with a servo problem.

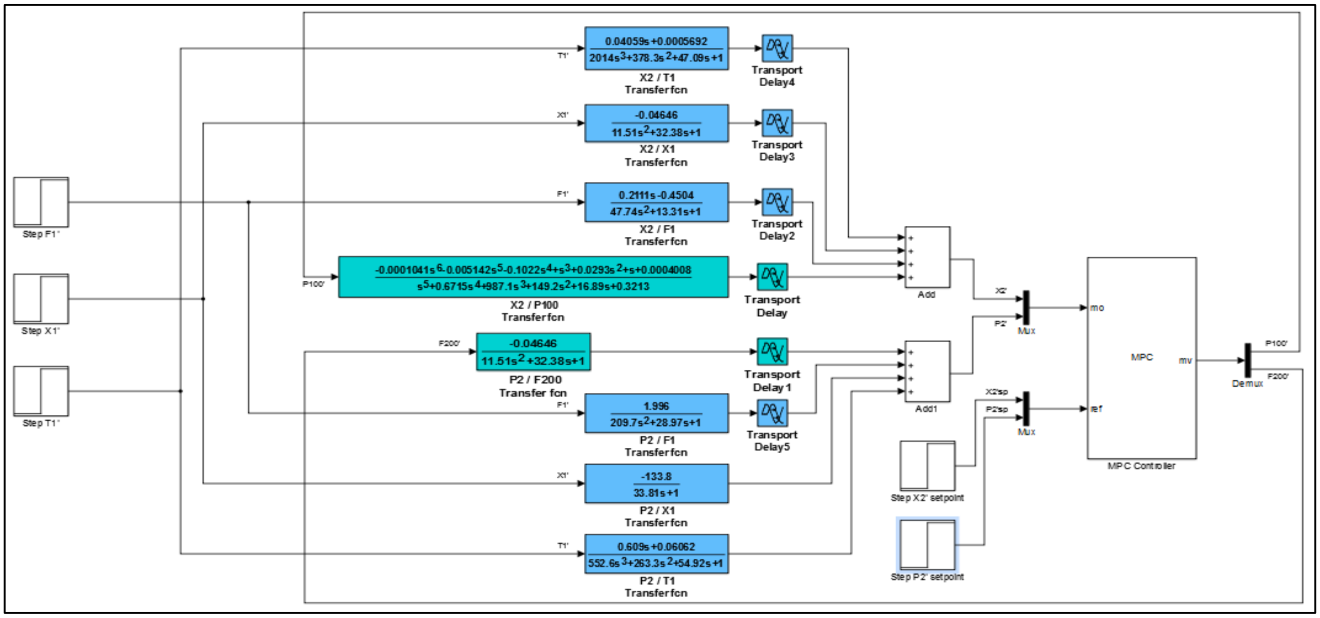
The main design task is to design and deploy a controller which well corresponds to all disturbance types and regulation/servo problems; such demand may show up as a challenging and intriguing work when using linear MPC, which is the type of controller explored in this text.

The focus for this control problem is to keep product composition (X2) as close as possible to its setpoint due to product quality, as well as trying to reduce its settling time (turning the system faster), with operating pressure showing a larger variance yet still being tracked towards its setpoint. Simulation length now is 500 minutes and the fixed step-size for ode4 solver is 0.1 minutes.

A nonlinear mathematical model has been derived in the literature to describe this process and a Simulink model has been developed from this mathematical model. Because such modeling is quite complex to implement in Simulink, a simplified block diagram is demonstrated below in order to show how to connect MPC block to a multivariable plant. The following block diagram will be used for this case study:



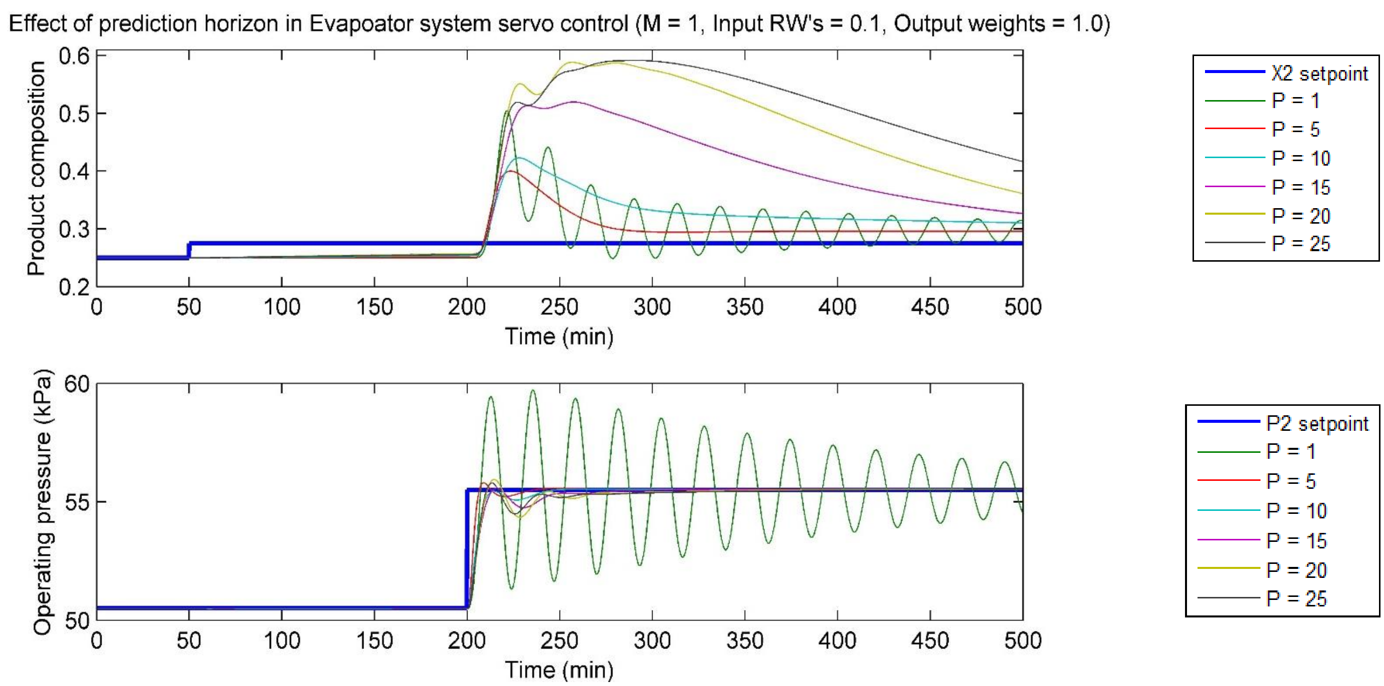
Nevertheless, one is suggested to implement equations (xii) to (xix) as schemed below and run the simulations:



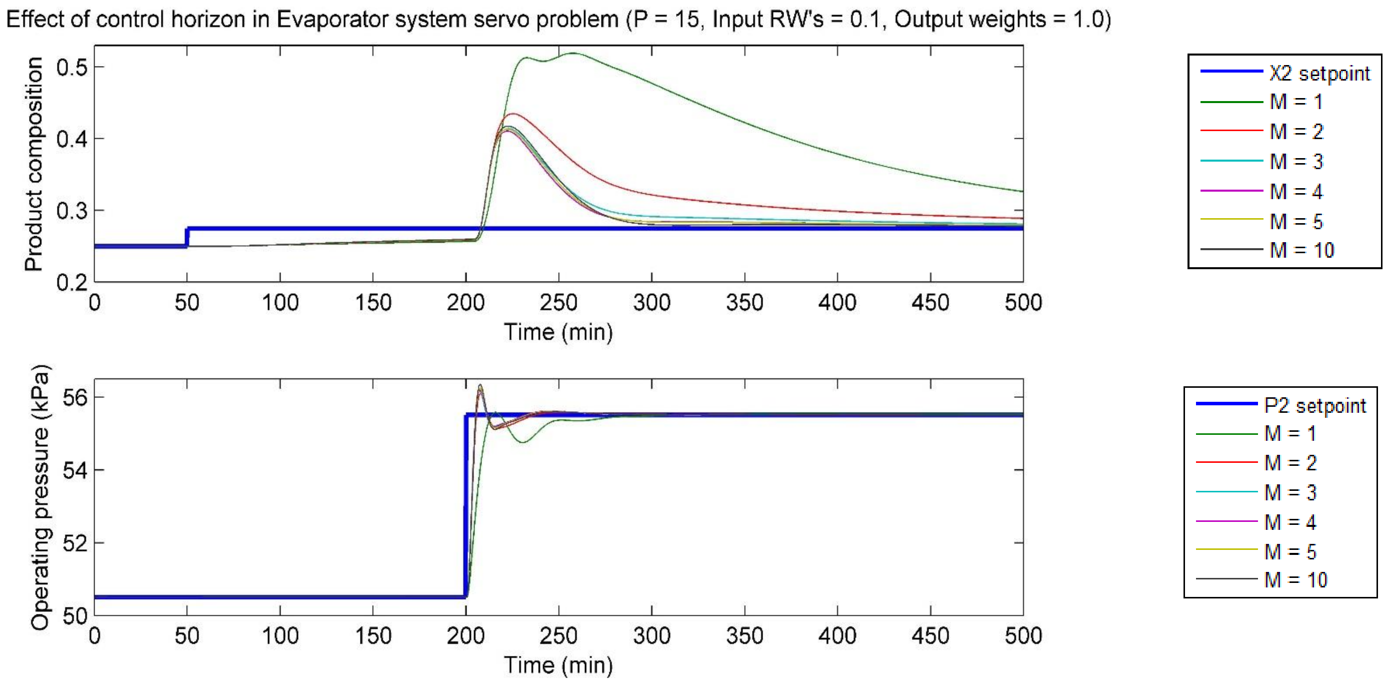
For this configuration, remember to configure all step blocks for both initial and final values as zero before starting to design the controller. Double click on “MPC Controller” block and click “Design”; a small window will show up asking the number of MV and CV’s, as well as the sample time. Set 2 manipulated variables, 2 controlled variables and 0.1 minutes, respectively. Then click “Ok” and wait for the toolbox to load.

4.2.1. Evaporator system: effect of prediction and control horizons on servo problem

Essentially the system traits a slow dynamics for product composition X2 but a faster one for operating pressure P2 when a MPC controller is on automatic, thus initial values of 1.0 for both prediction and control horizons are way from being satisfactory when setpoint changes are applied over these CV's, especially due to a predominant delay or dead time of 4.0 minutes (check equation xii); the next graph provides a sensitivity analysis for prediction horizon when +10% step changes are applied in X2 and P2 setpoints:



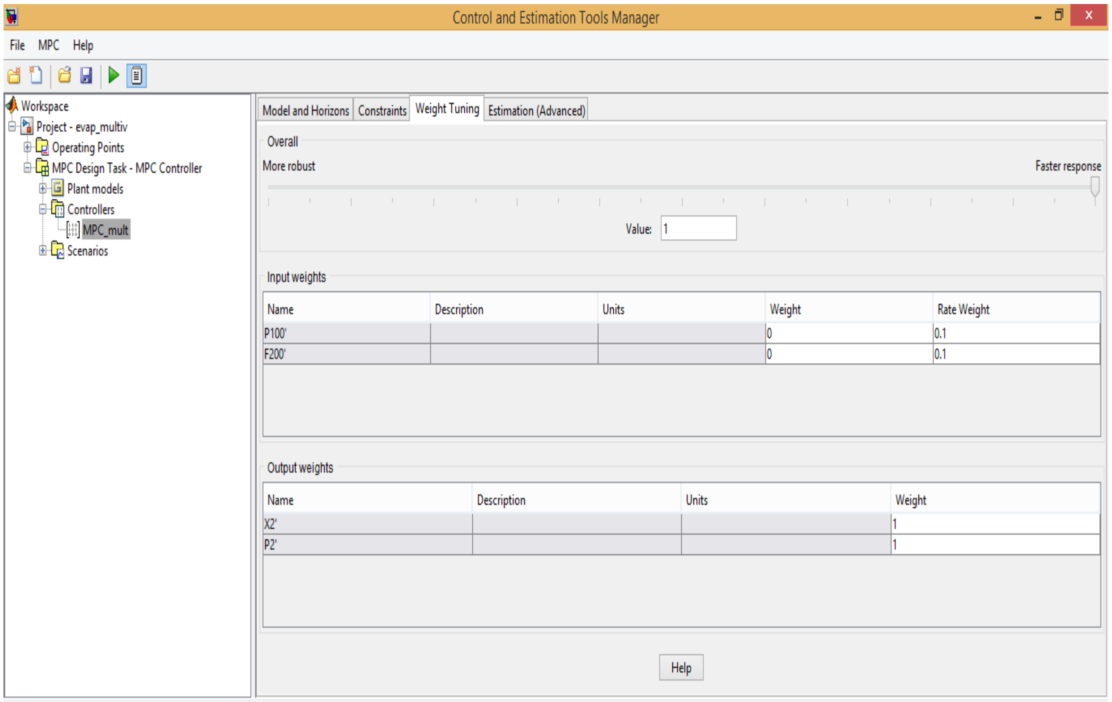
P-values less than the system time delay (4.0 minutes) destabilize both product composition and operating pressure; moreover, it can be noticed that a setpoint change on X2 minimally affected operating pressure. Also, the prediction horizon was unable to provide a decent X2 setpoint tracking (offset is present), however it did a fair job when tracking P2 setpoint for P-values above 5.0 minutes. Let us choose a value of 15 minutes for P and start analyzing control horizon's role in this model.



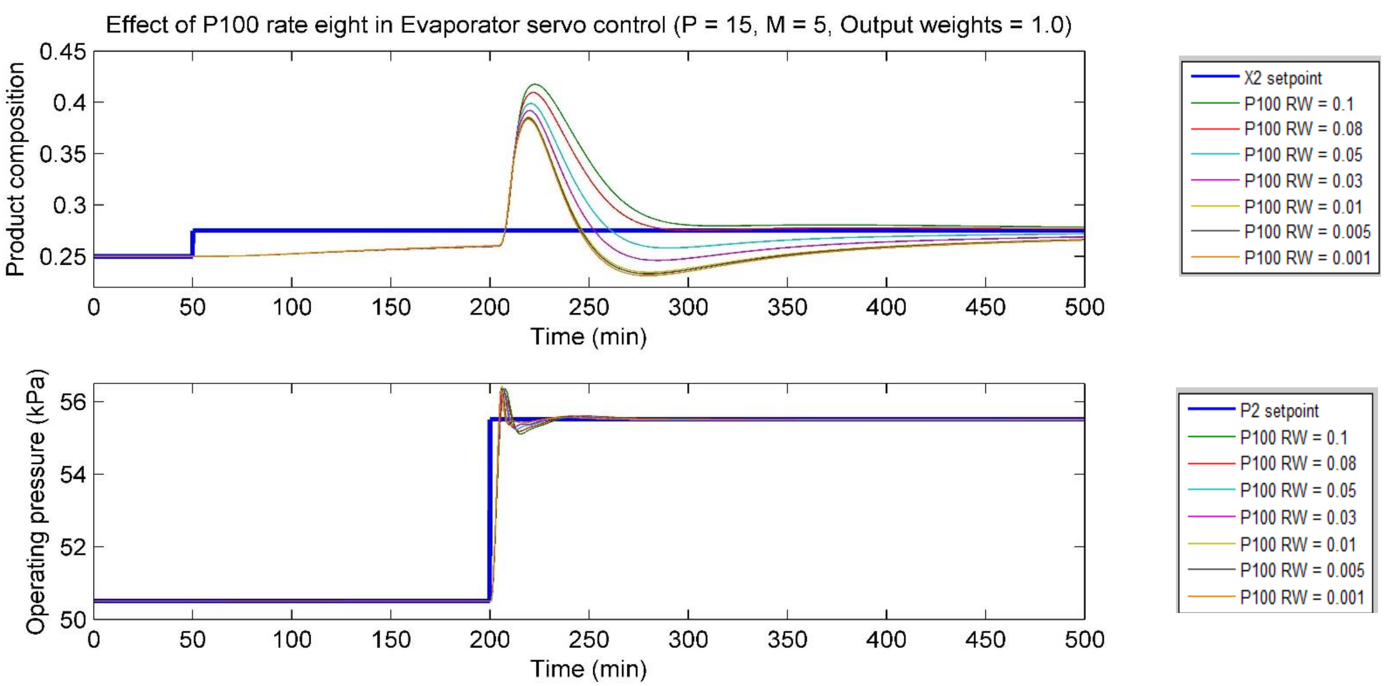
Operating pressure is still stable and well controlled yet product composition still features an offset, huge deviation from its setpoint and a lengthy settling time. An acceptable option would be a value of 5 to the control horizon, so let us choose this value and try to improve control performance of product composition by analyzing how input rate and output weights can boost X2 setpoint tracking.

4.2.2. Evaporator system: effect of input rate and output weights on servo problem

When prediction and control horizons are unable to provide a decent closed-loop response by themselves, one should start to deal with input and output weights. By decreasing or increasing their values, it is possible to accelerate or slow down a CV without substantially affect others; essentially, by decreasing an input rate weight a certain CV can be accelerated, with the opposite being valid, and by increasing output weights the control performance of a single CV can be put in first place, with the opposite again valid. A sensitivity analysis for these parameters can be performed on the following tab:

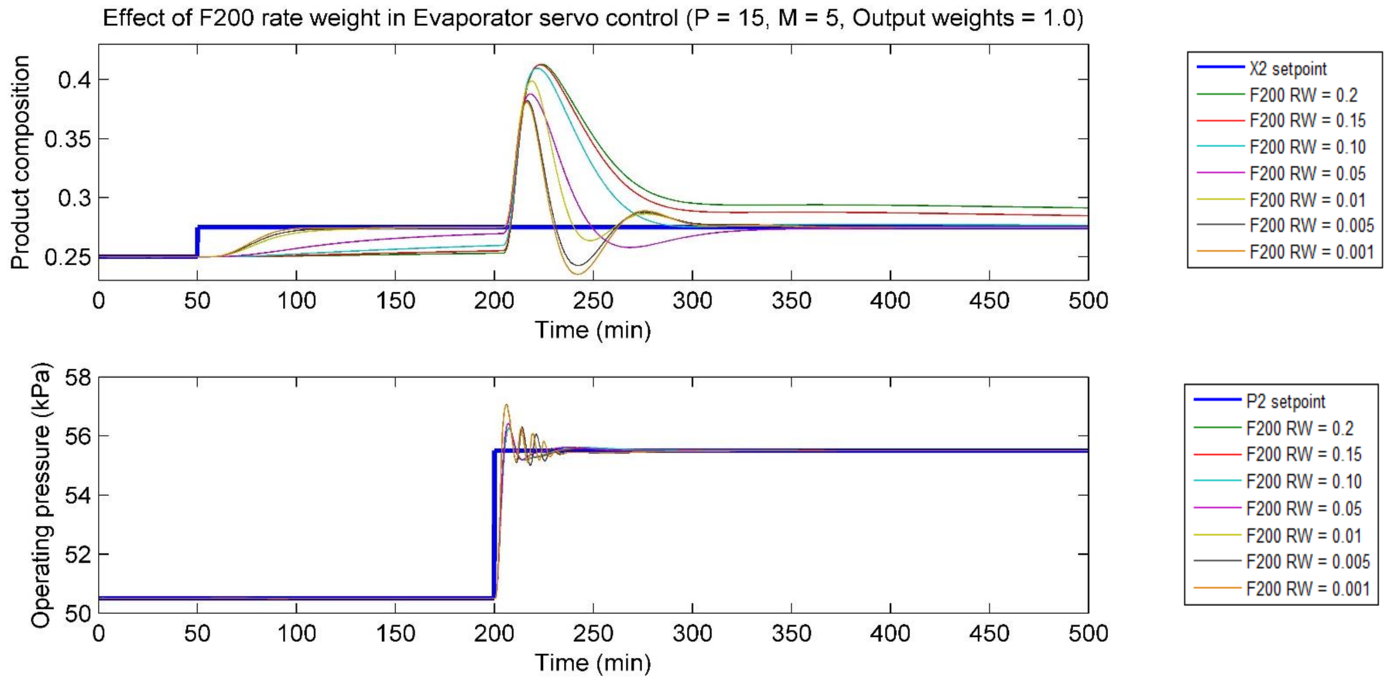


Let us first deal with input rate weights and try to improve product composition control performance by increasing its closed-loop response speed; this task can be performed by decreasing its input rate weight (P100') from the default value of 0.1.

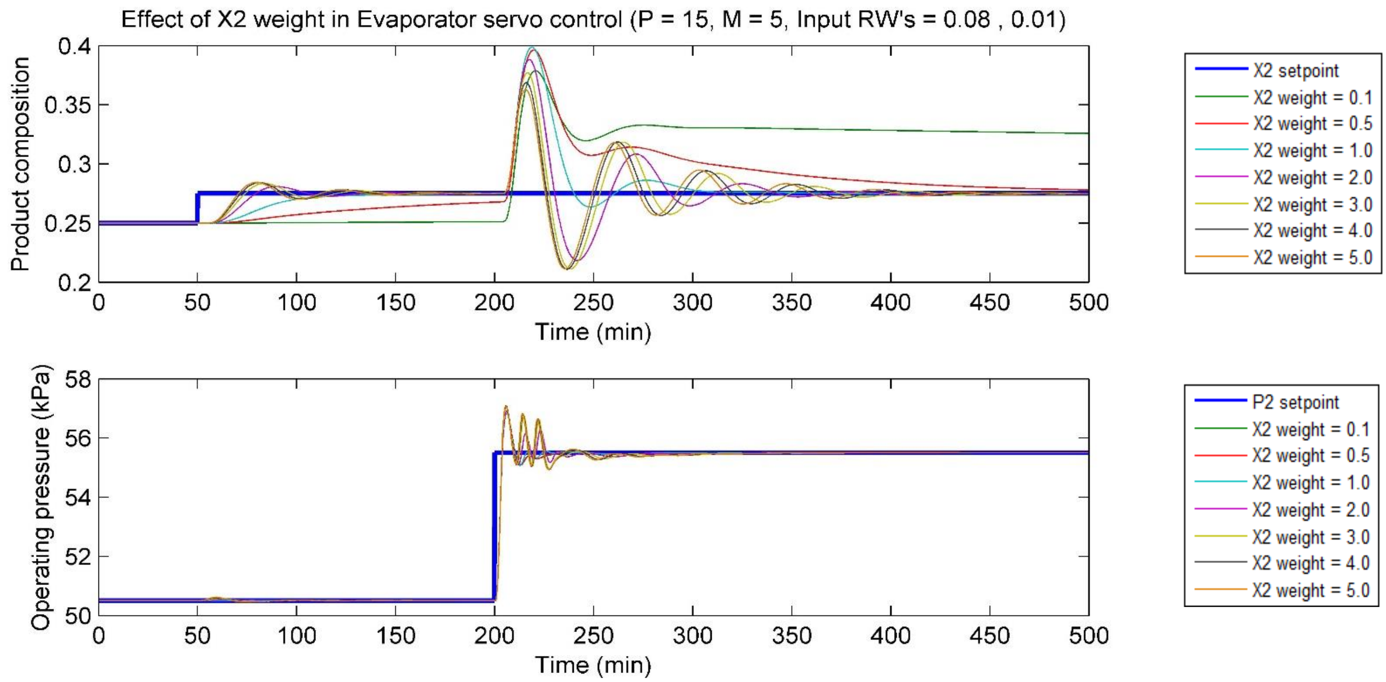


Product composition control was improved by managing P100 rate weight however still presents an offset; such situation can be solved by dealing with the remaining MV rate weight:

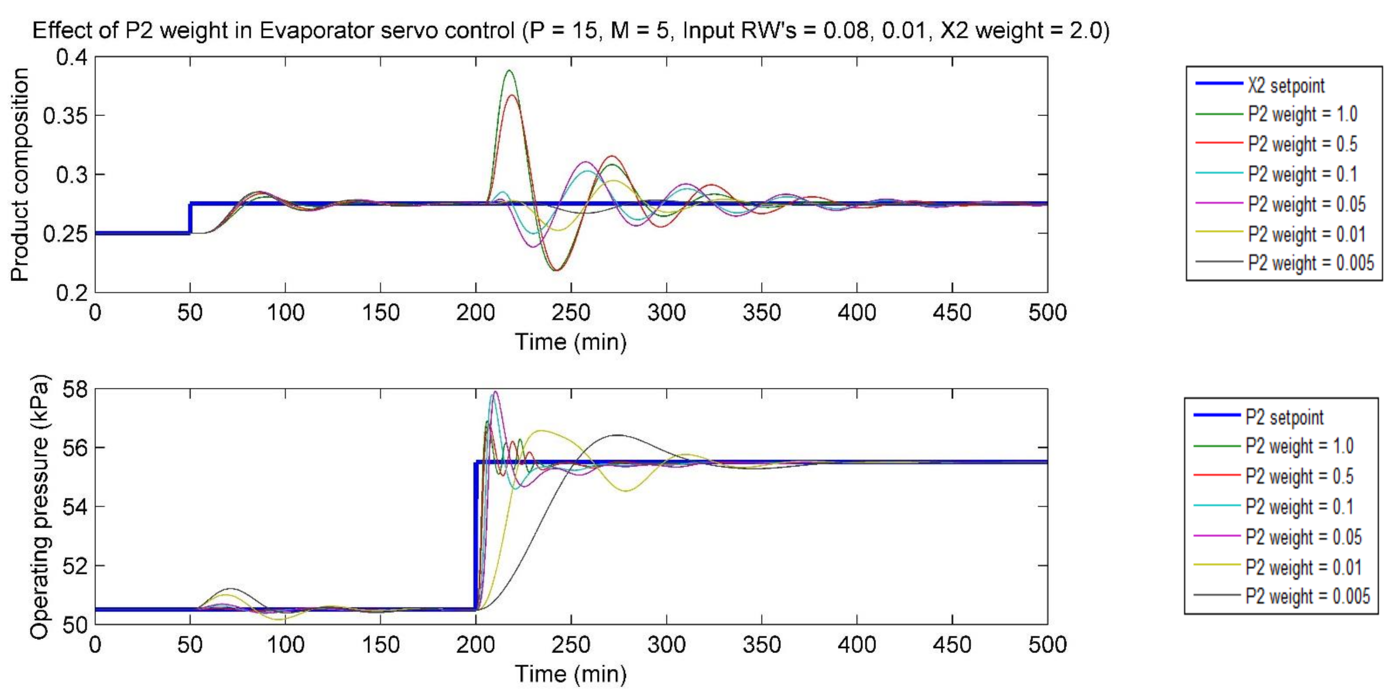
cooling water flow rate F200. For its sensitivity analysis, let us choose a P100 rate weight value of 0.08 and try F200 rate weights over and below the default value of 0.1.



By managing new values for F200 rate weights, it became possible to achieve a stable and fair closed-loop response for both CVs. Nevertheless, operating pressure seems to present an excessively fast response which is not necessary; to solve this situation and improve even more product composition control performance, let us keep a value of 0.01 for F200 rate weight and we can start to deal with output weights over X2 and P2.



Values less than 1.0 for X2 weight led to offsets and poor performance for its closed-loop response; in contrast, values above 1.0 provided a faster setpoint tracking actions even though turning the system oscillatory in face of a P2 setpoint change. Now let us choose an X2 weight value of 2.0 take a quick look on how we can improve composition control by trying different values for P2 weight.



Certainly the optimum value for P2 weight for this case is 0.005 due slight variations in product composition for either a step change on its setpoint or rejecting/minimizing deviances originated from P2 variations, no offsets in both variables and a smooth transition to P2 setpoint.

Finally, the MPC parameters for this controller are listed in Table 3:

Table 3 – MPC parameters for servo problem

Prediction horizon (min)	**Control horizon (# of moves)**	**Sample timing (min)**	**P100 rate weight**	**F200 rate weight**	**X2 weight**	**P2 weight**
15.0	5	0.1	0.08	0.01	2.0	0.005

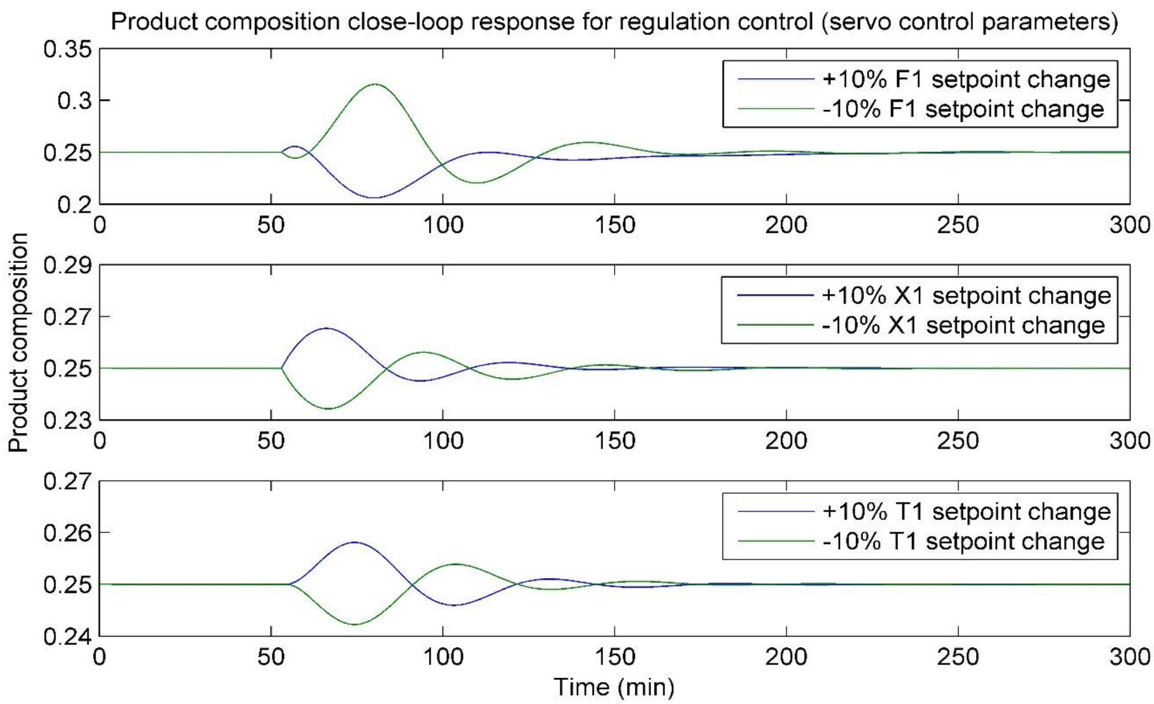
4.2.3. Regulation problem on evaporator system (disturbance rejection)

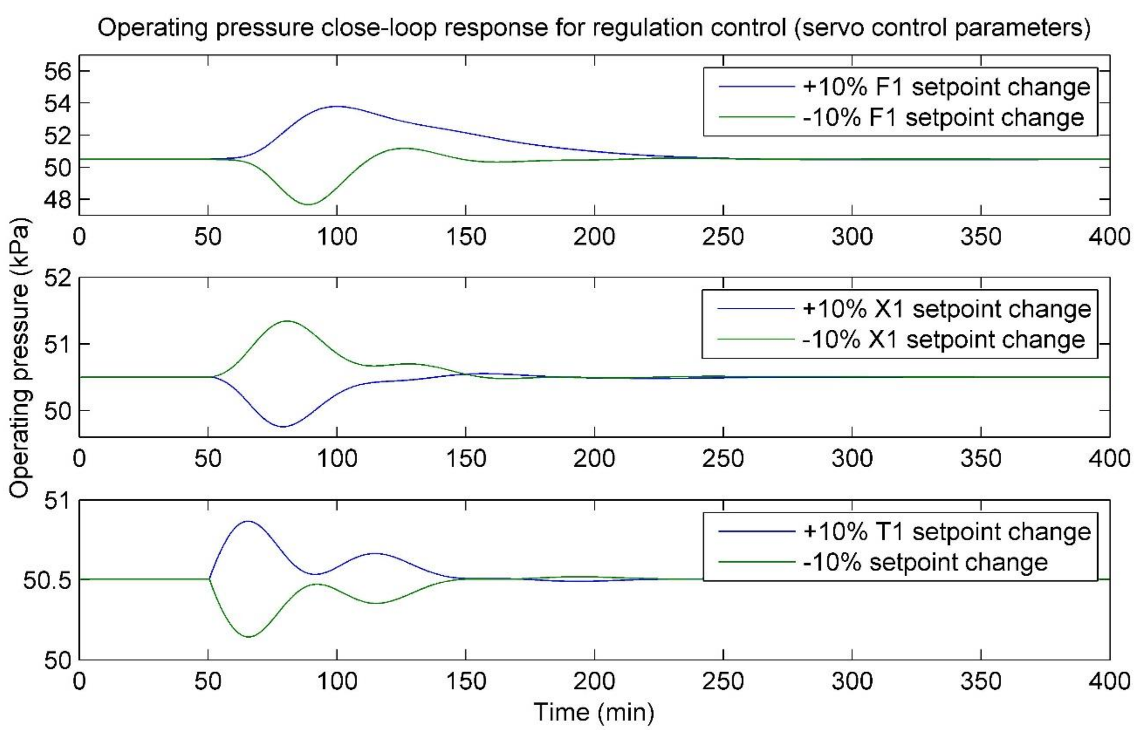
The previously designed controller well succeeded when dealing with setpoints changes over product composition and operating pressure, however this is just half part of the whole control problem. The second half consists basically of testing control performance face a disturbance, which for this case are based on fluctuations in feed flow rate, feed composition and feed temperature, and redesign its parameters if a poor disturbance rejection performance is obtained.

First, we will apply setpoint changes in all DVs and check controller performance, then a sensitivity analysis will be performed among prediction and control horizons to verify whether disturbance rejection can be improved or not. Finally, we will proceed with a sensitivity analysis involving input rate weights and output weights, and testing the new parameters for the servo problem.

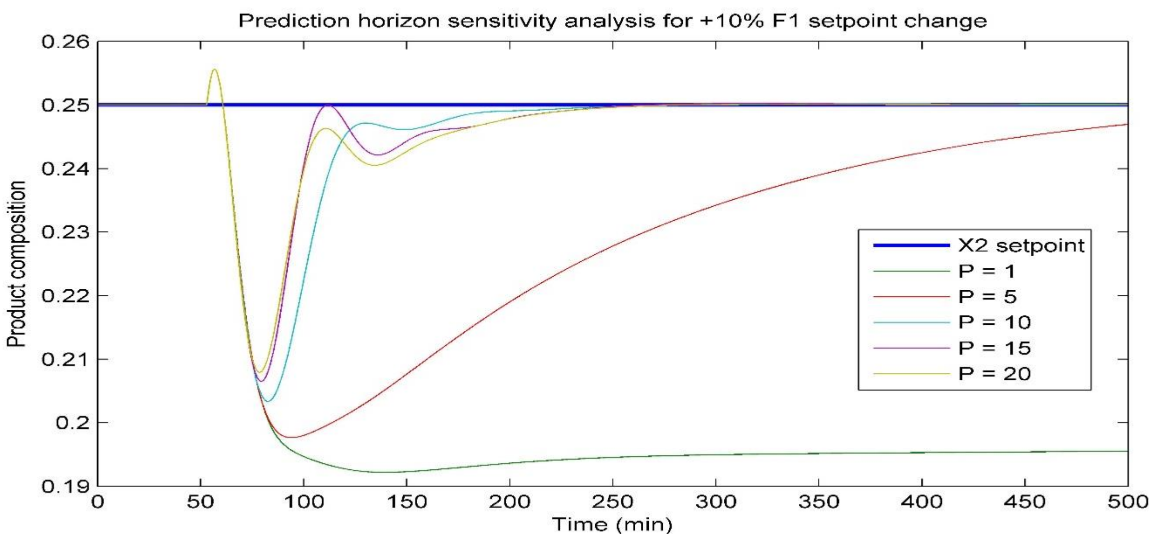
4.2.4. Evaporator system: effect of prediction and control horizon on regulation problem

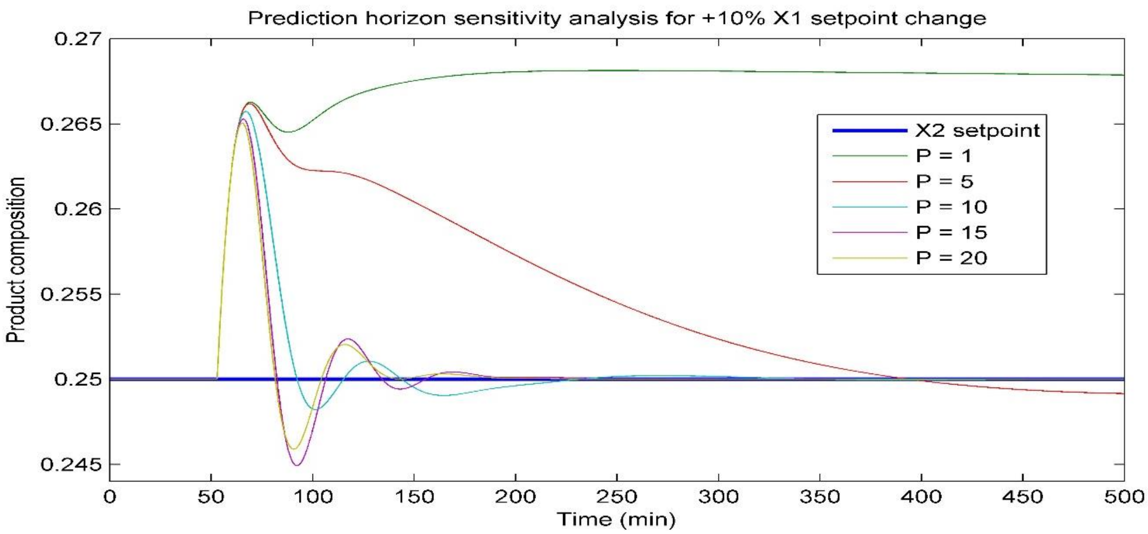
Simulations performed with the previously designed controller (Table 3 values) for +10% setpoint changes over F1, X1 and T1 provided the following results:

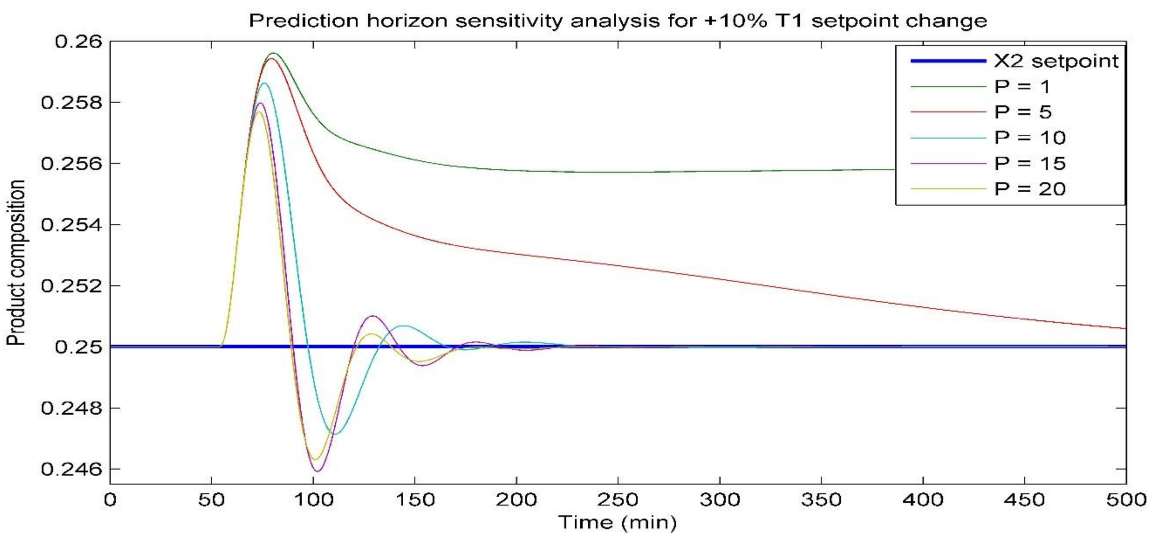




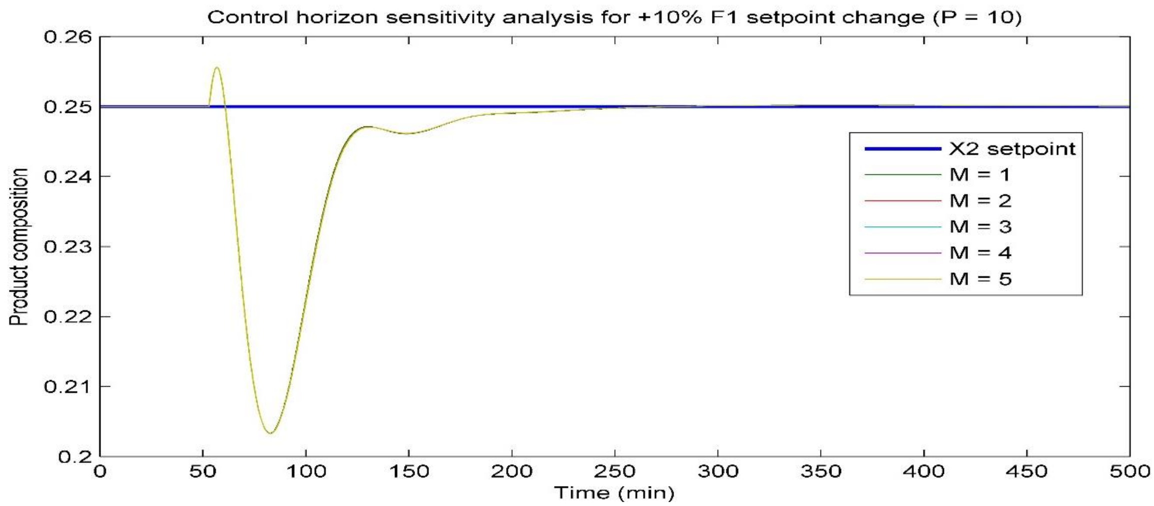
These are fair closed-loop responses for disturbance rejection of all DVs, featuring acceptable over/undershoots and relatively small settling times, still only a sensitivity analysis will enlighten the designer to a sense of whether such responses can be enhanced or not. For the next analysis, we will stick with just product composition control which is the focus for this case study, and explore the role of prediction and control horizon in a regulation problem (all the other parameters will be kept as their previous values except for the control horizon, which must be inferior to the prediction horizon). The following graphs show such sensitivity analysis for P:

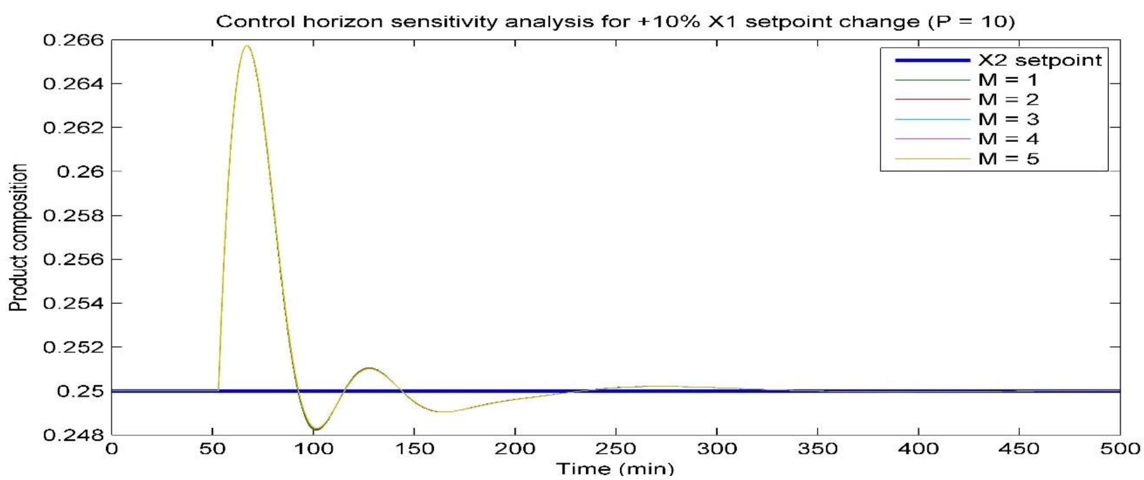


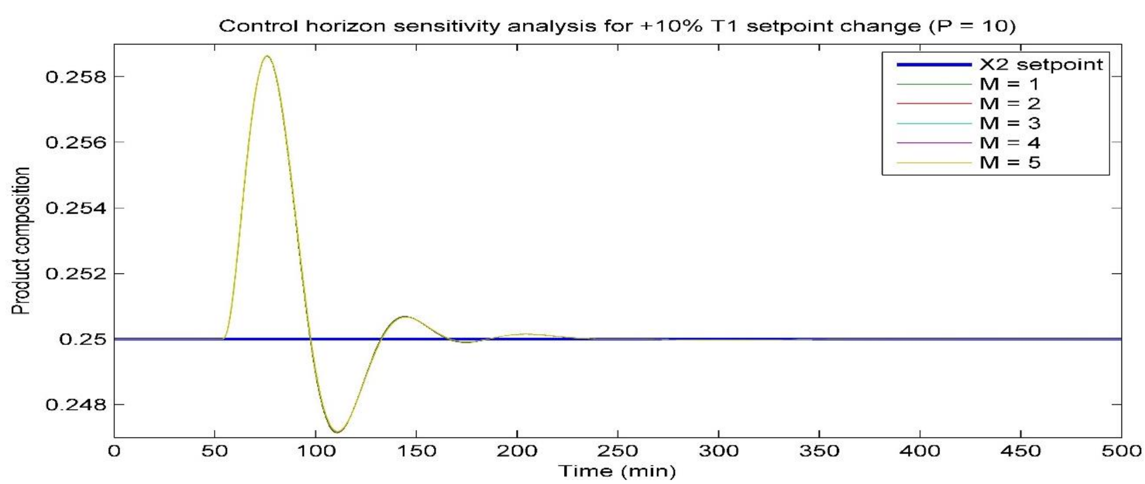




A value of 10 minutes for the prediction horizon seems to be enough to achieve a satisfactory closed-loop response for product composition, thus let us keep this value and check for the control horizon influence when rejecting disturbances:



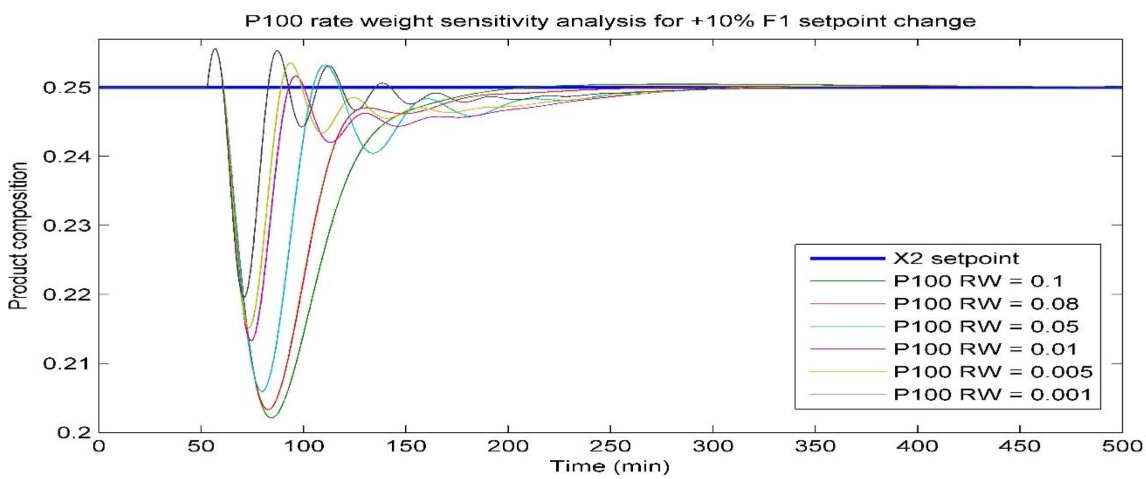


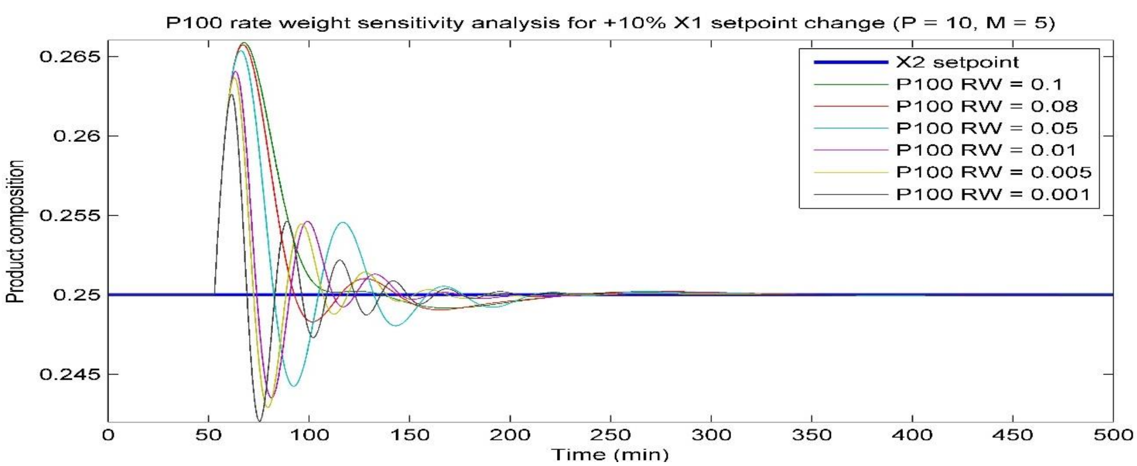


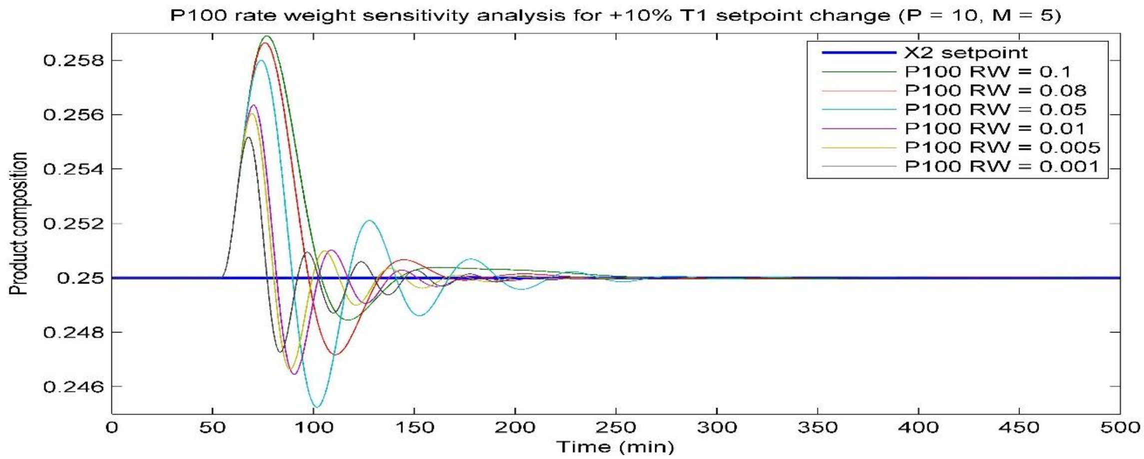
The control horizon did not seem to provide any sort of improvement in product composition control since all the curves are superposed; these results are due to input and output weights being already tuned, therefore the closed-loop response shapes are already defined by such values regardless the control horizon magnitude. One is welcomed to test this hypothesis in the servo problem.

4.2.5. Evaporator system: effect of input rate and output weights on regulation problem

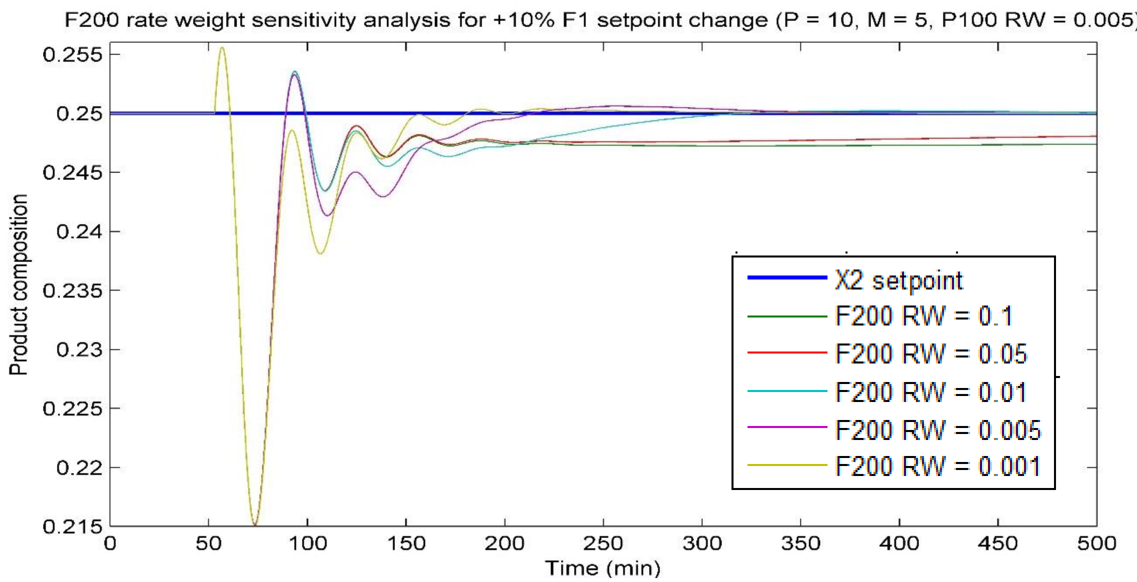
Once the sensitivity analysis for prediction and control horizons are accomplished, we can start to evaluate input and output weights role in a disturbance rejection routine. Let us choose a value of 5 control moves and vary P100 rate weight:

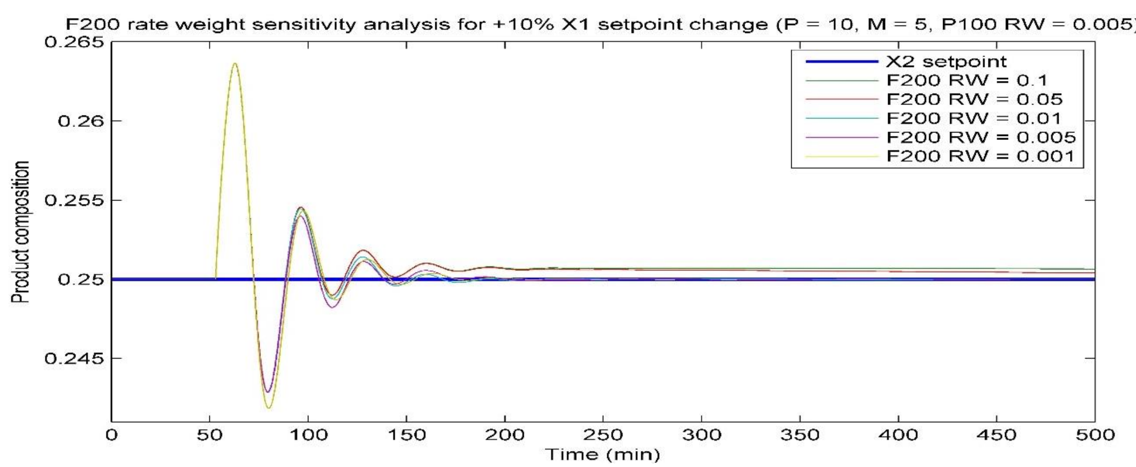


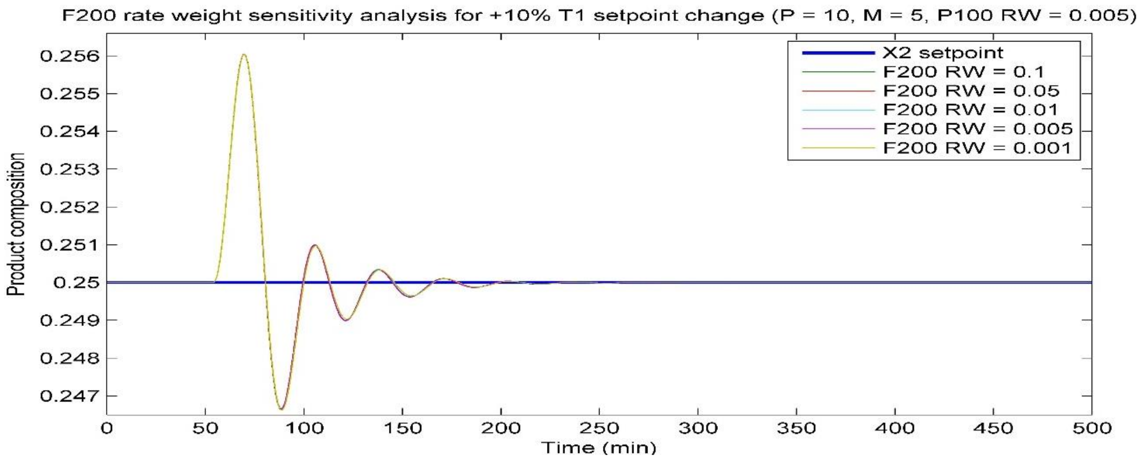




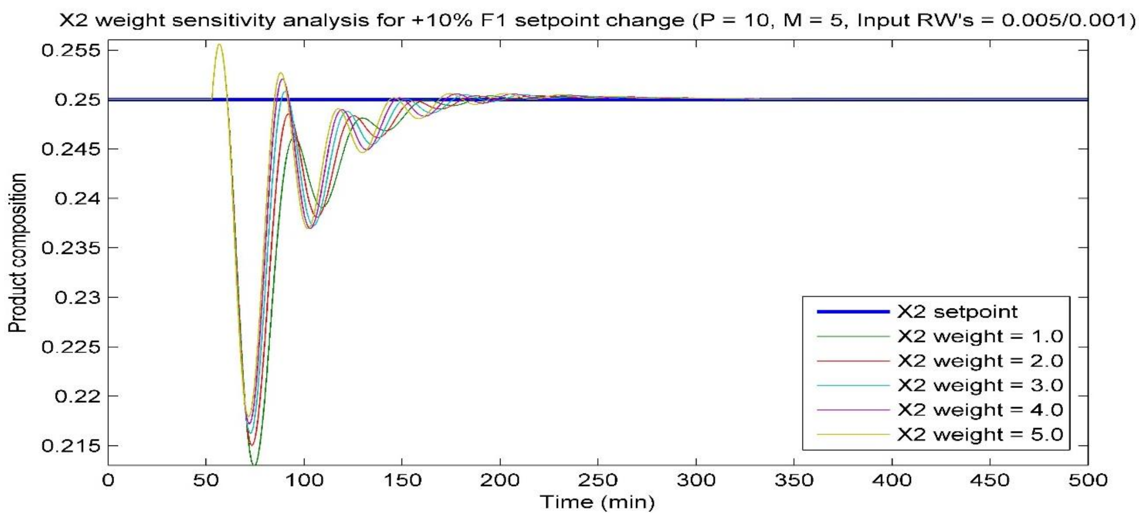
Let us choose a value of 0.005 for P100 rate weight (yellow line) and look for F200 rate weight effect in X2 close-loop response:

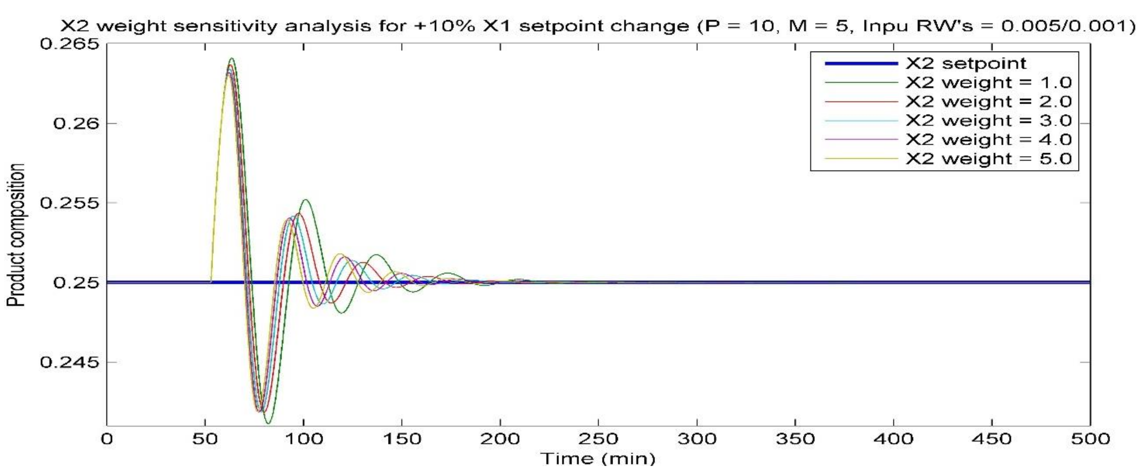


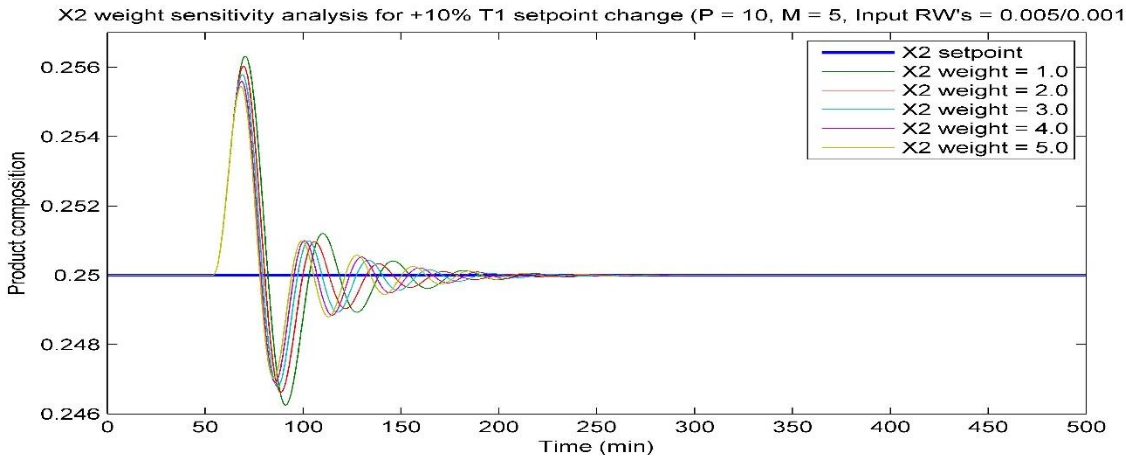




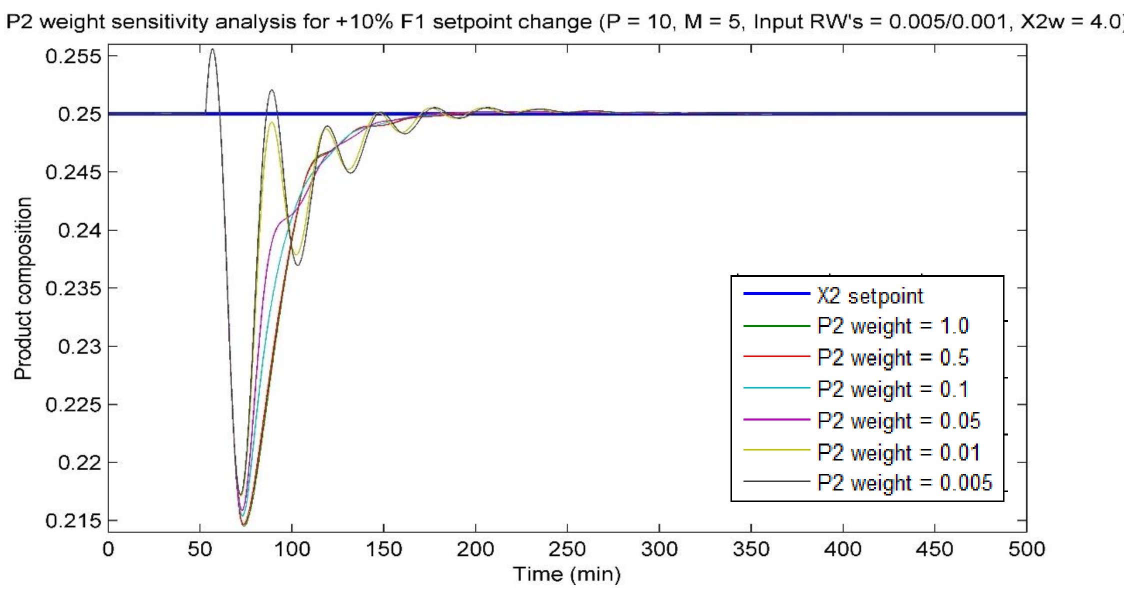
Let us keep a value of 0.001 (yellow line) for F200 rate weight due to its improvement over settling time, turning the system faster yet oscillatory. The last sensitivity analysis will investigate output weights part in the whole controller design; again, keep in mind that P2 weight is 0.005 (servo problem design) and let us start with X2 weight:

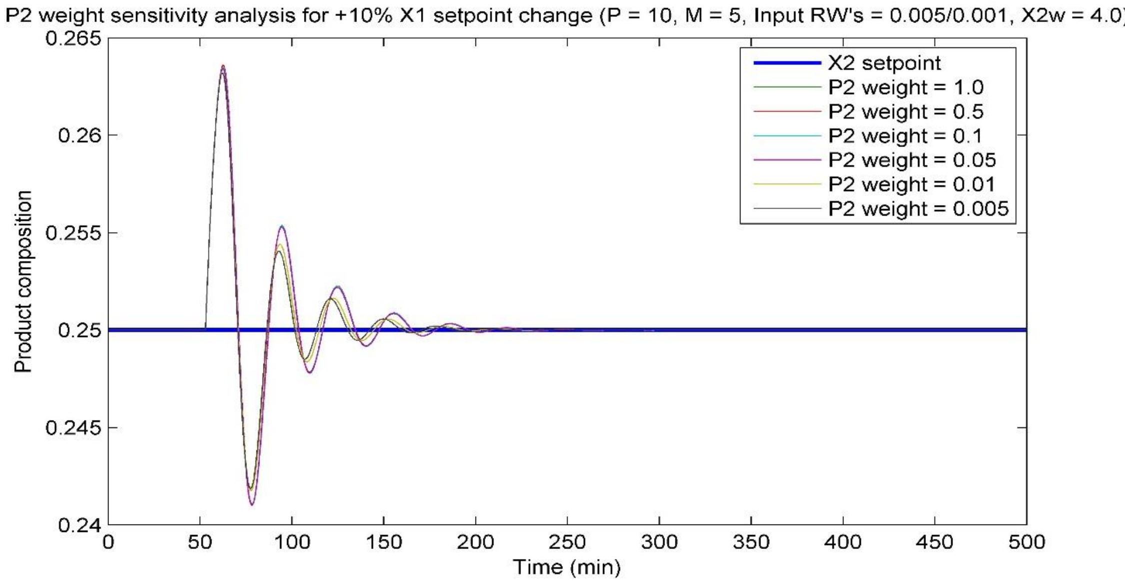


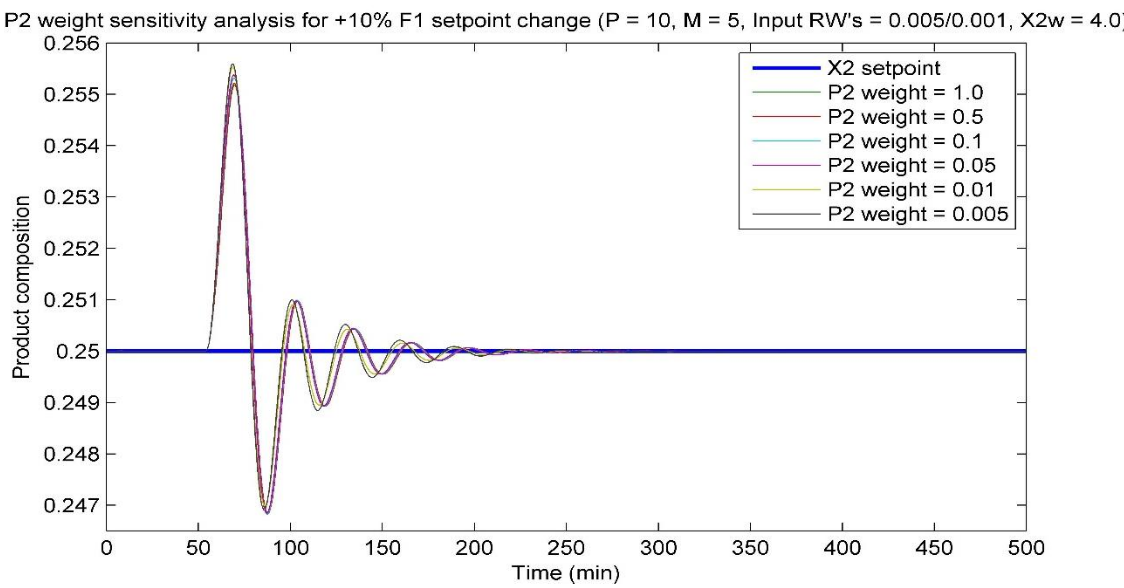




By increasing X2 weight it became possible to reduce over/undershoots when dealing with all disturbance inputs without destabilizing the control loop; let us choose a new value for product composition weight of 4.0 and diagnose operating pressure weight influence:







Both values of 0.1 or 0.05 for P2 weight seemed to reduce oscillations in X2 closed-loop response when rejecting variations in feed flow rate and feed composition, the other values provided a sluggish and oscillatory responses, demonstrating a poor control performance.

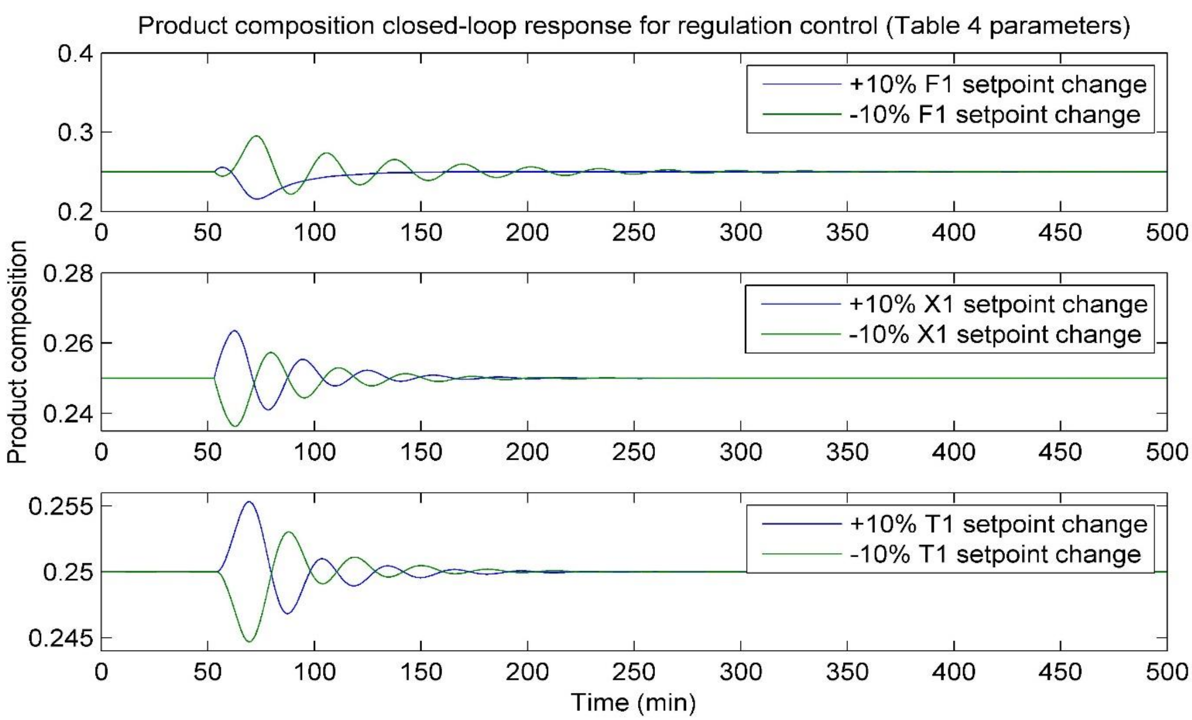
Ultimately, the MPC parameters for this controller are listed in Table 4:

Table 4 – MPC parameters for regulation problem

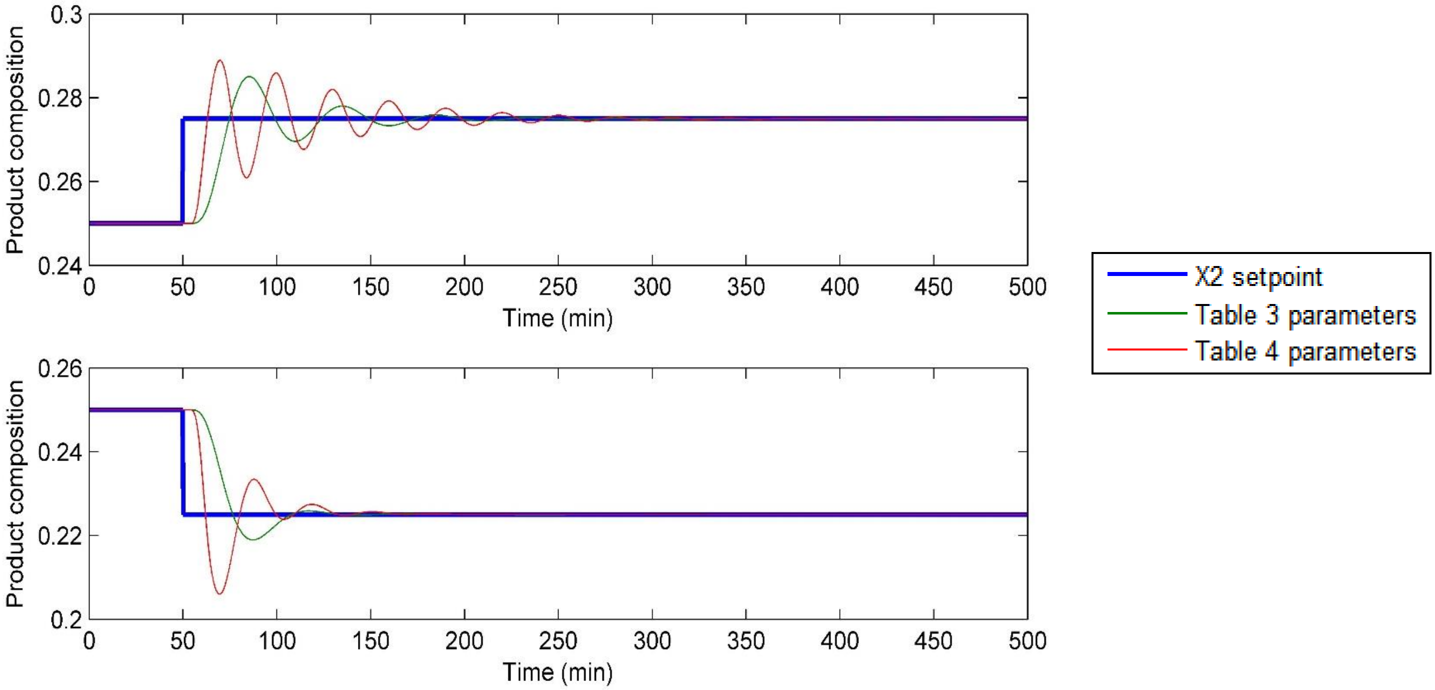
Prediction horizon (min)	**Control horizon (# of moves)**	**Sample timing (min)**	**P100 rate weight**	**F200 rate weight**	**X2 weight**	**P2 weight**
10.0	5	0.1	0.005	0.001	4.0	0.10

4.2.6. Evaporator system: controller validation final test

A final test to properly validate and deploy the controller designed for regulation problem would be to apply positive and negative step changes over all DVs setpoints and check for system stability, as well for X2 and P2 setpoint changes. The next closed-loop responses were obtained using parameter values listed in Table 4:



A quick comparison between these results and the ones depicted in topic's 4.2.4. first graph induces the control designer to conclude that there were no improvements among any disturbance rejection, thus the parameters utilized in servo problem are satisfactory for both control problems. Nevertheless, it is important to check for any enhancements in servo problem performance with regulation problem parameters:



Certainly, the last graphs corroborated the conclusion drown from closed-loop responses in regulation problem using Table 4 parameters: the first tuning action (Table 3 parameters) stated satisfactory results for both servo and regulation problem. This is quite remarkable for such a nonlinear system as an evaporation assembly, demonstrating MPC controller's power and capability to deal with linear and nonlinear constrained multivariable systems.

5. References

Bequette, B. (2003). *Process control: Modeling, design, and simulation*. Upper Saddle River, N.J.: Prentice Hall PTR.

Seborg, D. (2011). *Process dynamics and control: International student version* (3.rd ed.). Hoboken, NJ: John Wiley and Sons.

Morari, M., & Zafiriou, E. (1989). *Robust process control*. Englewood Cliffs, N.J.: Prentice Hall.

Prett, D., & A, C. (1988). *Fundamental process control*. Boston: Butterworths.