1. **SIMULATOR OF AVERAING LEVEL CONTROL SYSTEM**

**Model Equations.**

Parameters

A = 2000 m2

hmax = 2 m

Fin,max = 6 m2/s

**PI tuning (Skogestad method)**

The process can be described as the following transfer function

H

y\_mf

u

Sensor

+

Filter

Process

The combined transfer function of the process + sensor + measurement filter can be expressed as the following.

Where τ is the dead time, which is usually present in most of the systems. For a system with a transfer function such as the one described above the value of the PI controller tuning parameters using Skogestad method is as follows.

The value of the process gain K is 1/A. The value can be obtained as the following derivation.

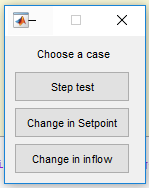
The same value can be obtained from the performing a step test in the simulator and calculating the slope of the response curve. The response is presented in Figure 1.

Since there is no dead time in the system the user has to decide on the value of the Tc

**Matlab Code**

The system explained above is implemented as a matlab code. The code can be found in the filename ‘exercise1.m’. The model is integrated using the forward Euler method. The integration of the model is written as a matlab function with the name ‘model.m’. A random measurement noise is introduced in the system by adding a rand function with a variance of 0.01.This value can be increased by changing the value of noise in the file ‘model.m’ line 21.

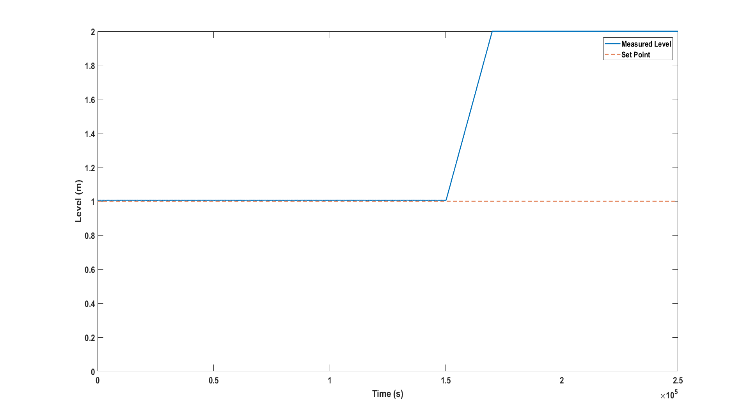
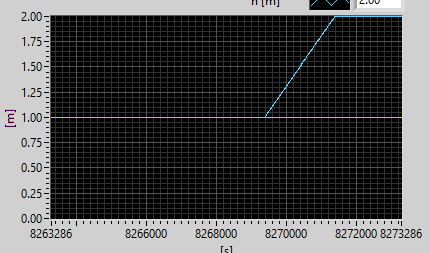
When the script ‘exercise1.m’ is executed, it asks the user to input the measurement filter time constant. Later it asks for the choice for the case. A pop up menu appears asking for options between three cases to choose from.



**Case 1:** Step response test for the manipulated variable

On choosing the case 1. The script ask for the step size of the manipulated variable. It also suggests to provide with a value between -3 and 3. Since the value of the outflow has been expected to vary between 0 m3/s and 6 m3/s and the initial value of the outflow is assumed to be 3 m3/s.

The results of the step response are presented in the figure 1. The plot is compared to the response obtained from the simview level control simulator presented in figure 2.

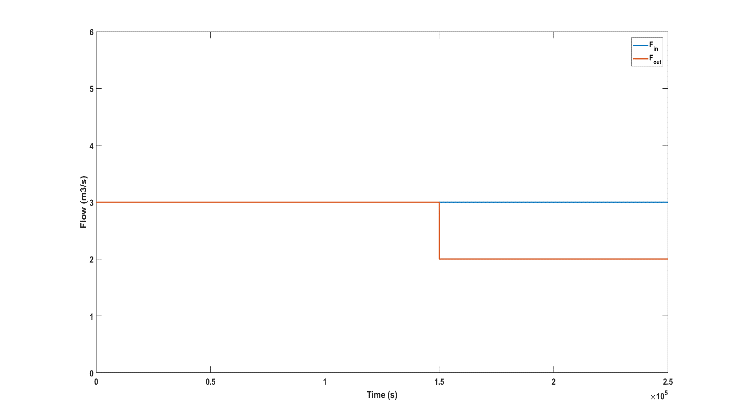
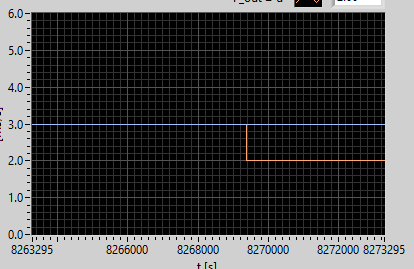
 

Figure 1. Step change in outflow

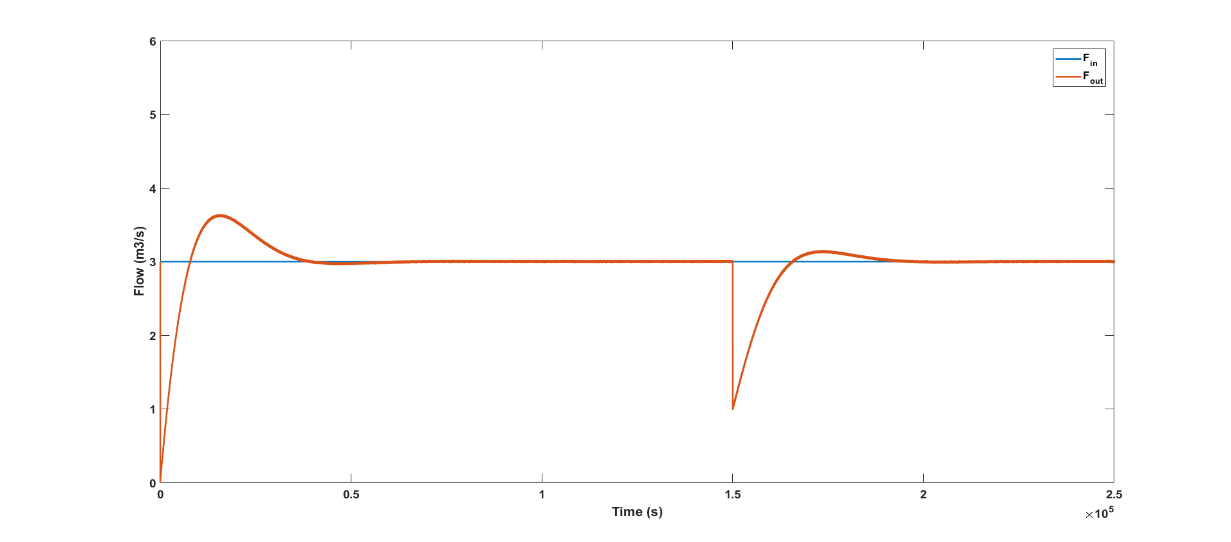
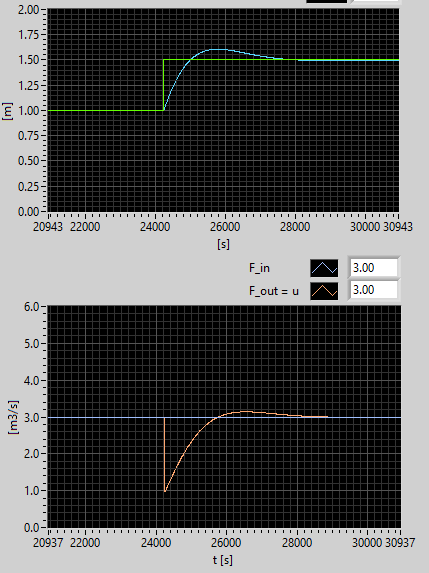
From the figure the slope can be calculated as

**Case 2**: Change in set point

A PI controller is included in the simulator, with the tuning parameters obtained by Skogestad tuning method. The controller performance is tested by providing a step change in the set-point. This can be done by choosing the second option in the choice menu. The simulator then asks the user to input the value of close-loop time constant. The following values are provided for matlab simulator.

Step in set-point = 0.5

Close-loop time constant = 500

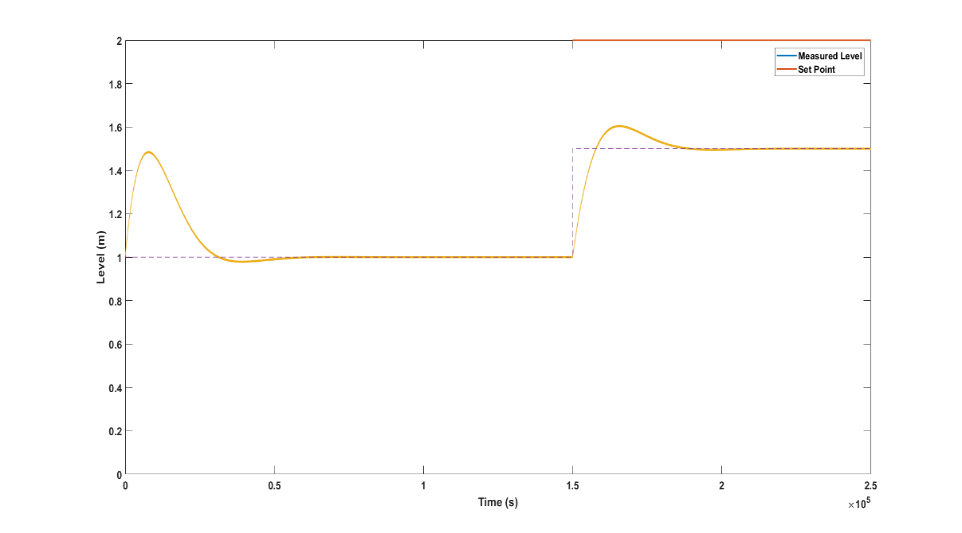
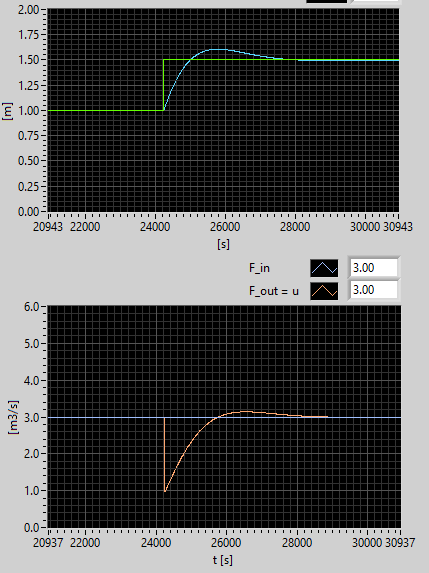
 

Figure 2. Step change in set point

The figure 2 presents a comparison between the result from MATLAB and the result from the simview simulator. The response of the controller to the variation in set-point appears to be the same for both the simulators.

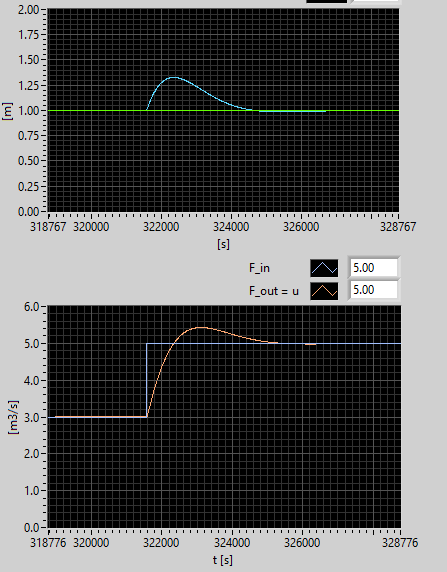
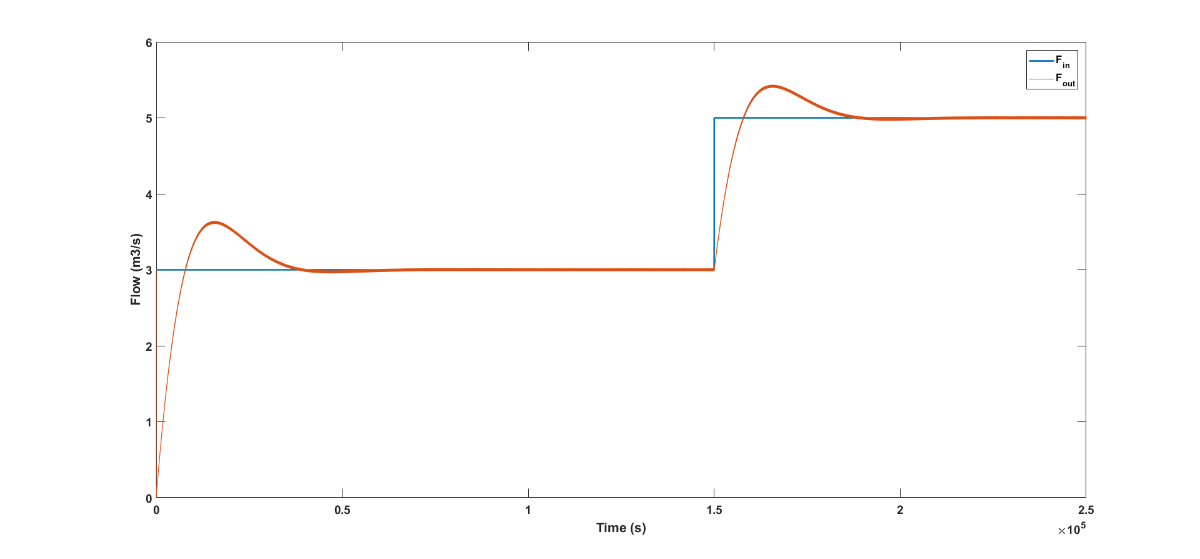
**Case 3:** Change in Inflow (disturbance)

A PI controller is included in the simulator, with the tuning parameters obtained by Skogestad tuning method. The controller performance is tested by providing a step change in the inflow into the tank. The third option is selected from the option menu. The simulator then asks the user to input the value of close-loop time constant. The following values are provided before running the matlab simulator.

Step change in Inflow = 2

close-loop time constant = 500

The figure 3 presents a comparison the MATLAB simulation and the result from the simview simulator. The response of the controller to the variation in set-point appears to be the same for both the simulators.



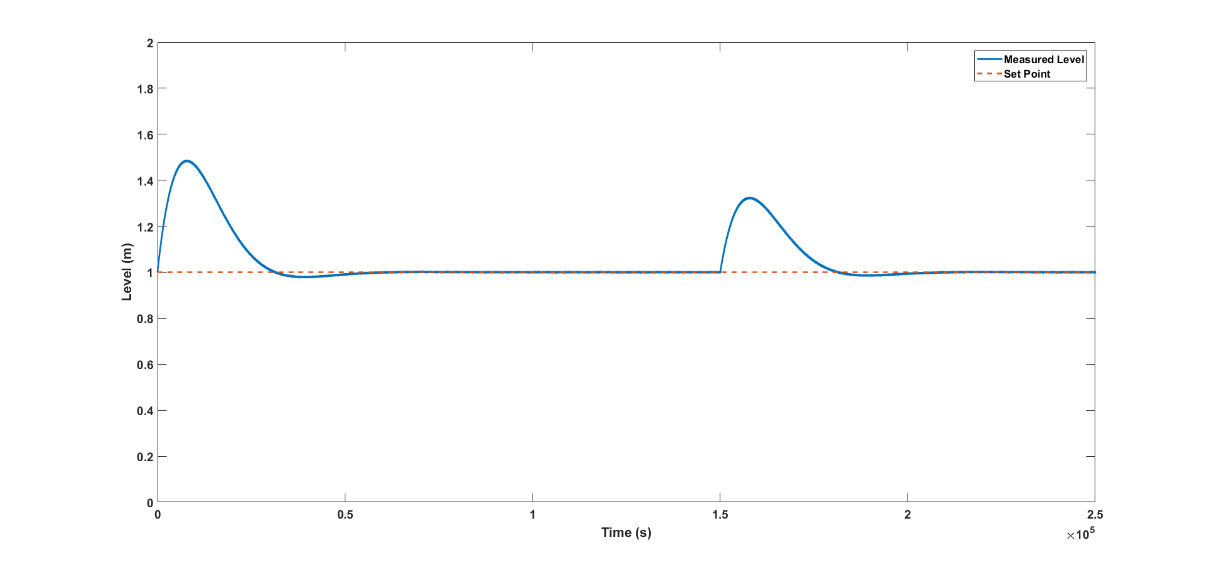
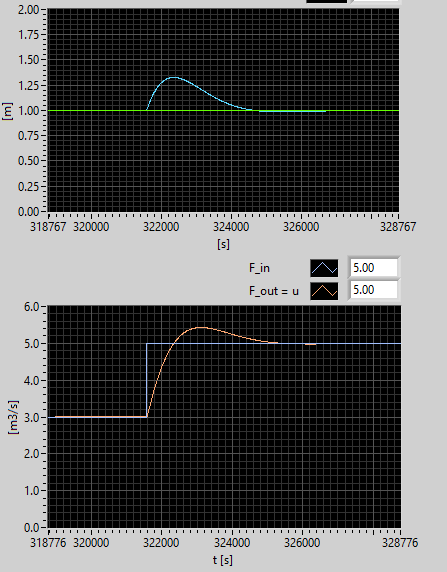
 

Figure 3. Step change in inflow

1. **SIMULATOR OF A TIME DELAY**

The system described below is chosen to demonstrate a simulator with time delay.

A pure dead time can be added to the process by modifying the equation as follows

The simulator ‘exercise 2.m’ is built to present the time delay in the model. The matlab script asks the user to input the dead time which exists in the process. For the results presented below a time delay of 500 s is used.

A step change in the Qout (manipulated variable) is provided and the response of the change in the measured variable is plotted dynamically. The simulation also presents graph of the response of input and output which clearly shows the process time-delay.

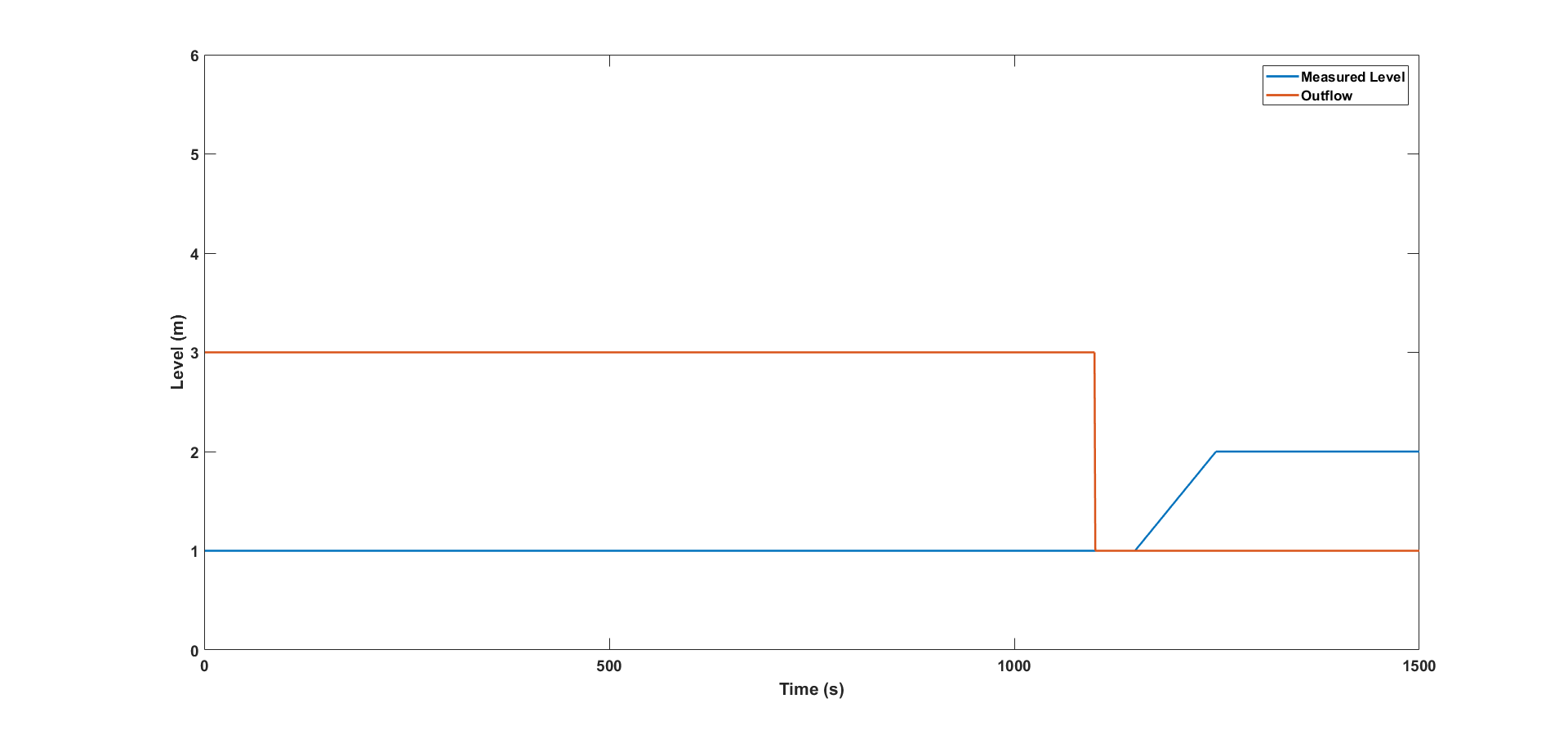


Figure 4. Process time delay between process input and measured output.