Control Systems

V1: Rapid Control Prototyping (RCP) with Matlab/Simulink

1 Lab Task 2

2.1 Task description:

The aim of this task was to measure the system behaviour of two unknown systems, System 3 and System 2 and estimate their control parameters by first classifying them as either PT1 elements or PT2 elements and using the equations of the respective types to then optimize the parameters.The system behaviour was observed by inputting a given step-response and then measuring the output of the system, after which the estimated transfer function was compared to the actual output. This was done using the Simulink Desktop Real-Time software, which provides a real-time kernel for executing the application on a Windows desktop, as well as MATLAB.

2.1.1 and 2.1.2 Results:

After finding out that System 3 was a PT1 system, we estimated the the transfer function using the system response and looking at (steady state gain) K= y_∞ / u_∞ to find the gain and the time it took to reach 0.63 y_∞ to find the time constant (τ). Then System 2 was PT2, So we used (steady state gain) K=y_∞ / u_∞, and calculated $\omega 0 = \omega/(\sqrt{1-D^2})$ using $\omega 0 = (2\pi / T)$, where (T) is the time between two peaks and D being calculated from the overshoot Mp = $(y_{max}-y_\infty)/y_\infty$.

$$G_{S3(s)} = \frac{1.5}{1.48(s)+1} \qquad G_{S2(s)} = \frac{0.765}{\left(\frac{s}{4.85}\right)^2+2(0.35)\left(\frac{s}{4.85}\right)+1}$$

2.1.1 and 2.1.2 :

Figure 1: System 3 with Input Signal u, Measured Output Signal y, and Simulated Output Signal ym

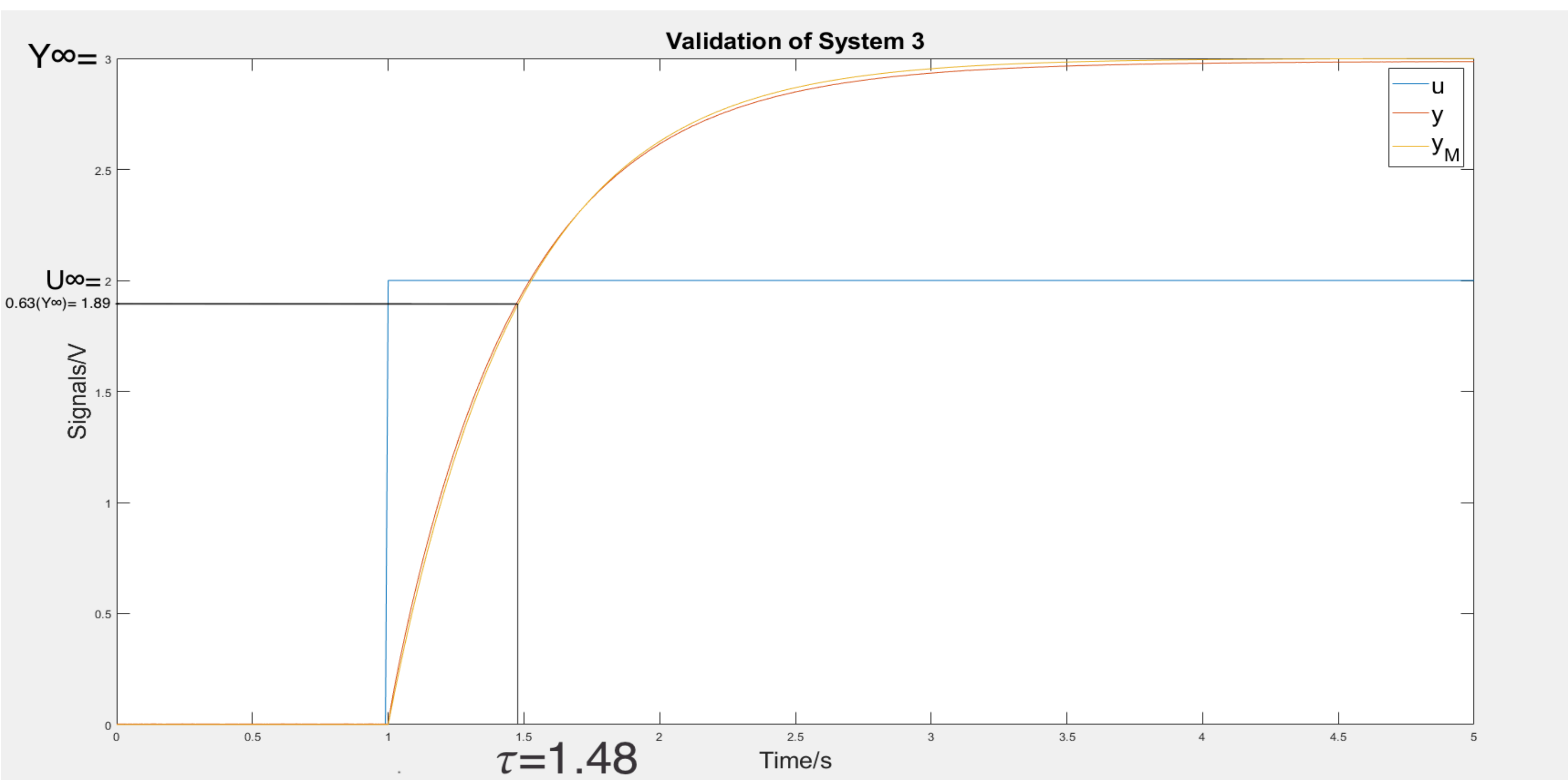
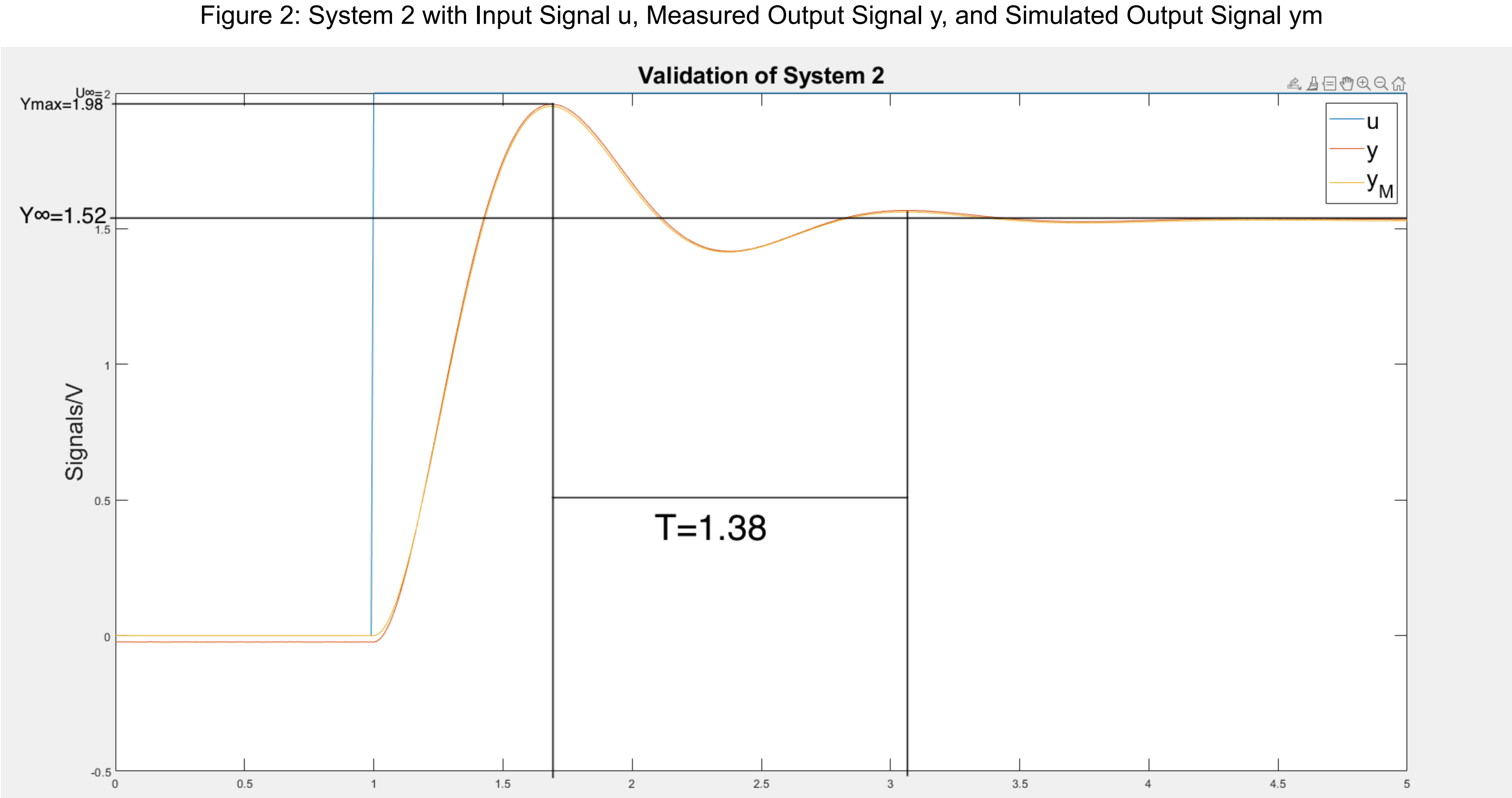


Figure 2: System 2 with Input Signal u, Measured Output Signal y, and Simulated Output Signal ym



2 Lab Task 2

After making a PID Controller as a series connection of System 3 and 2, we had to estimate its control parameters. This was more difficult than the previous tasks and required a lot of guess work. First we switched off the I and the D parts of the controller to get the P value, then switched off the D and assign the I and P lastly switched on all to guess the final values. We changed the values for P and I to adjust for the other parameters.

Parameters of the optimised controllers:

	K_R	T_I	T_D
P controller	1.3	-	-
PI controller	0.9	0.5	-
PID controller	1.3	0.7	1

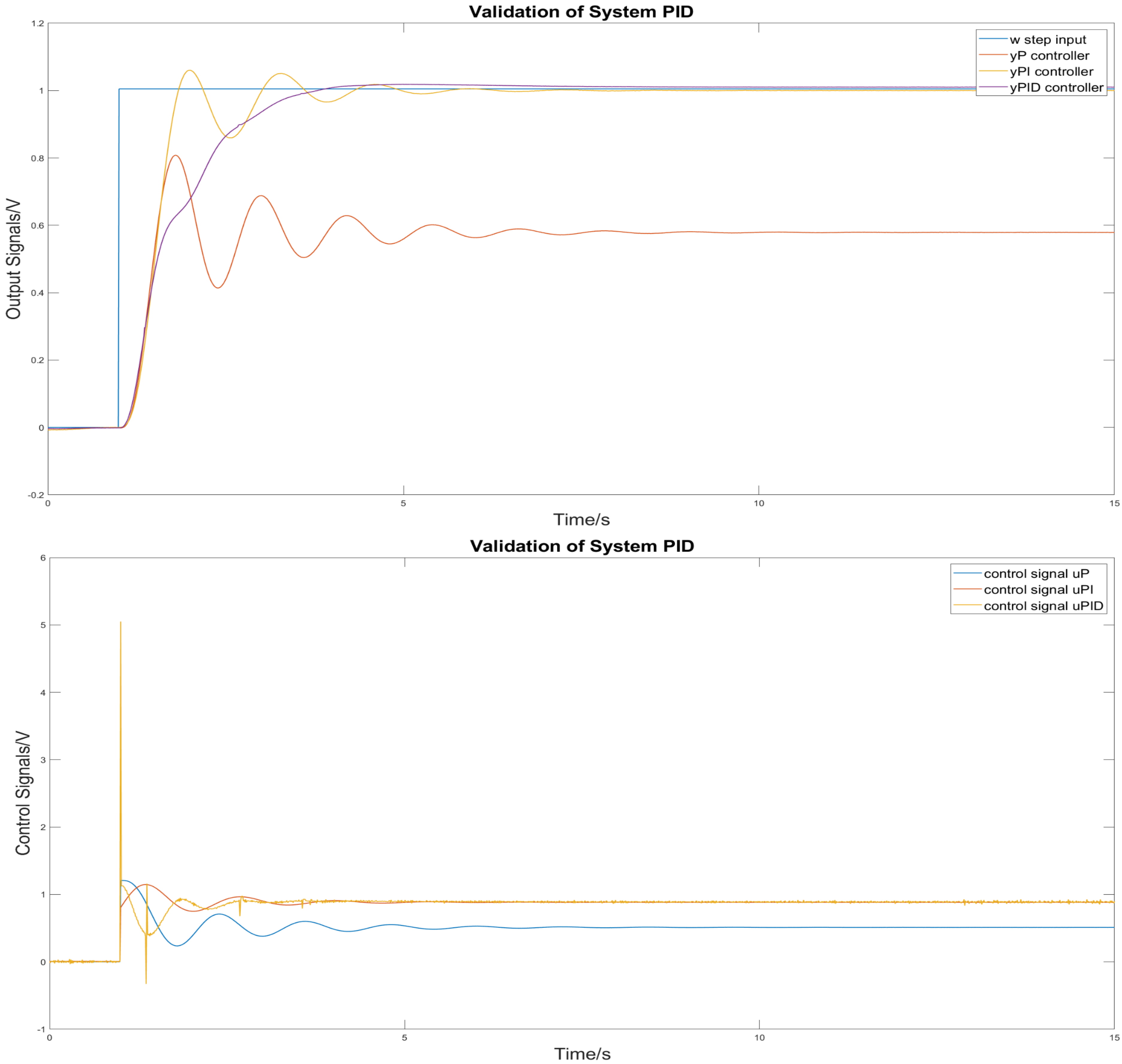


Figure 3:System Outputs and Controller Inputs

Conclusion:

1)when we used the P controller by disabling the I and D controllers, the controller's output is directly proportional to the current error signal which result in steady-state error, where the process variable settles at a value below the desired setpoint.

2)Now, enable the I controller and tune its parameters K_R and T_I we find out that the PI controller still does not consider the rate of change of the error, it lead to slight overshoot, especially in systems with significant delay.

3) By using PID controller we find out that it aims to provide a balanced response that minimizes steady-state error, overshoot, and settling time, leading to improved overall control performance.

1 Lab Task 2