Question-1

Code

```
clear; close all;
tc = 5;
s = tf('s');
Gp = tf(1.15,[50 15 1]);
Gm = tf(1,[50 15 1]);
% PID controller
Gc  = tf([50 15 1],[tc 0]);
% Disturbance tf
Gd = tf(1,[5 1]);
lambdavec = 0.001:0.001:0.5;
r1 = ones(length(lambdavec),1);
r2 = r1;
for k = 1:length(lambdavec)
    % Feedforward controller
    Gff = -Gd*1/(lambdavec(k)*s+1)/Gp;
    % sys is Y/Do
    sys = (Gff*Gp+Gd)/(1+Gp*Gc);
    S = stepinfo(sys);
    % get settling time as close as possible to 15
    r2(k) = S.SettlingTime;
    r1(k) = abs(S.SettlingTime-15);
end
[val,loc] = min(r1);
lambda = lambdavec(loc);
```

Question-2

Tables

Table 12.1 IMC Controller Settings for Parallel-Form PID Controller (Chien and Fruehauf, 1990)

Case	Model	K_cK	τ_I	τ_D
A	$\dfrac{K}{\tau s+1}$	$\dfrac{\tau}{\tau_c}$	τ	–
B	$\dfrac{K}{(\tau_1 s+1)(\tau_2 s+1)}$	$\dfrac{\tau_1+\tau_2}{\tau_c}$	$\tau_1+\tau_2$	$\dfrac{\tau_1\tau_2}{\tau_1+\tau_2}$

Table 12.4 Controller Design Relations Based on the ITAE Performance Index and a First-Order-plus-Time-Delay Model (Lipták, 2006)*,†

Type of Input	Type of Controller	Mode	A	B
Disturbance	PI	P	0.859	−0.977
		I	0.674	−0.680
Disturbance	PID	P	1.357	−0.947
		I	0.842	−0.738
		D	0.381	0.995
Set point	PI	P	0.586	−0.916
		I	1.03†	−0.165†
Set point	PID	P	0.965	−0.85
		I	0.796†	−0.1465†
		D	0.308	0.929

*Design relation: $Y = A(\theta/\tau)^B$ where $Y = KK_c$ for the proportional mode, τ/τ_I for the integral mode, and τ_D/τ for the derivative mode.
†For set-point changes, the design relation for the integral mode is $\tau/\tau_I = A + B(\theta/\tau)$.

Code

```
clear;close all;
s = tf('s');
%% Given Data
Kv = 0.9; Kip = 0.75;
t = (0:1:11)';
T =
([12,12.5,13.4,14,14.8,15.4,16.1,16.4,16.8,16.9,17,16.9]'
-12)/2;
plot(t,T);
% Can't see any inverse response, so mostly no zero
assume first order plus
% time delay.
%% Model Estimation
[X,RESNORM,RESIDUAL,EXITFLAG] = lsqcurvefit(@resp,[5 2
1],t,T);
K = X(1)*Kv*Kip;
tau1  = X(2);
tau2 = X(3);
Gp = tf(K,conv([tau1 1],[tau2 1]));
%% Part a) IMC
tauc = max(tau1 ,tau2)/2;
Kc = (tau1 +tau2)/(K*tauc);
tauI = tau1 + tau2;
tauD = (tau1*tau2)/(tau1 + tau2);
Gc_imc = Kc*(1+1/(tauI*s)+tauD*s);
%% Part b) ITAE (setpoint)
% FOPTD approximation
D = tau2/2;
```

```matlab
tau = tau1 + tau2/2;
% Use tables
AP = 0.965;
BP = -0.85;
Kc_b = AP*(D/tau)^BP/K;
AI = 0.796;
BI = -0.1465;
tauI_b = tau/(AI + BI*(D/tau));
AD = 0.308;
BD = 0.929;
tauD_b = AD*(D/tau)^BD*tau;
Gc_b = Kc_b*(1+1/(tauI_b*s)+tauD_b*s);
%% Part c) ITAE (disturbance)
AP = 1.357;
BP = -0.947;
Kc_c = AP*(D/tau)^BP/K;
AI = 0.842;
BI = -0.738;
tauI_c = tauI/(AI*(D/tau)^BI);
AD = 0.381;
BD = 0.995;
tauD_c = AD*(D/tau)^BD*tau;
Gc_c = Kc_c*(1+1/(tauI_c*s)+tauD_c*s);
%% Function to give step response for lsqcurvefit
function Y = resp(params,tvec)
    K = params(1);
    tau = params(2);
    tau2 = params(3);
    Gp = tf(K,conv([tau 1],[tau2 1]));
    Y = step(Gp,tvec);
end
```

Question-4

Part a) Feedforward controller

Model Workspace

Workspace data

Data source: MATLAB Code

MATLAB Code:

```
s = tf('s');
Kd = 0.25/0.5;
Dd = 10;
taud = 30;
Gd = tf(Kd,[taud 1],'iodelay',Dd);
Kp = 0.4;
Dp = 7.5;
taup = 25;
Gp = tf(Kp,[taup 1],'iodelay',Dp);
Gff = -1*Gd*(1-Dd*s/2)*(taup*s+1)/(1-Dd*s/2)/Kp;
```

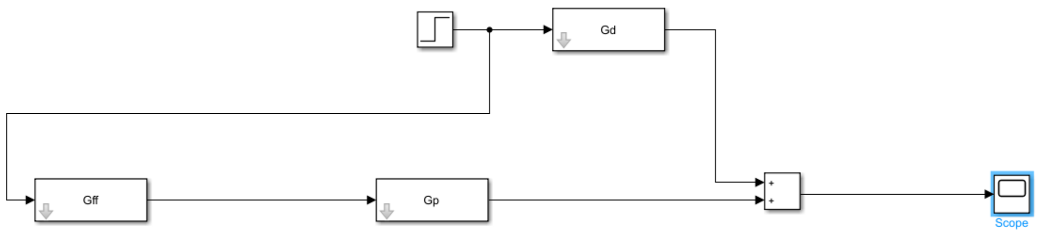

Figure 1: SIMULINK DIAGRAM of the system with just a feed-forward controller

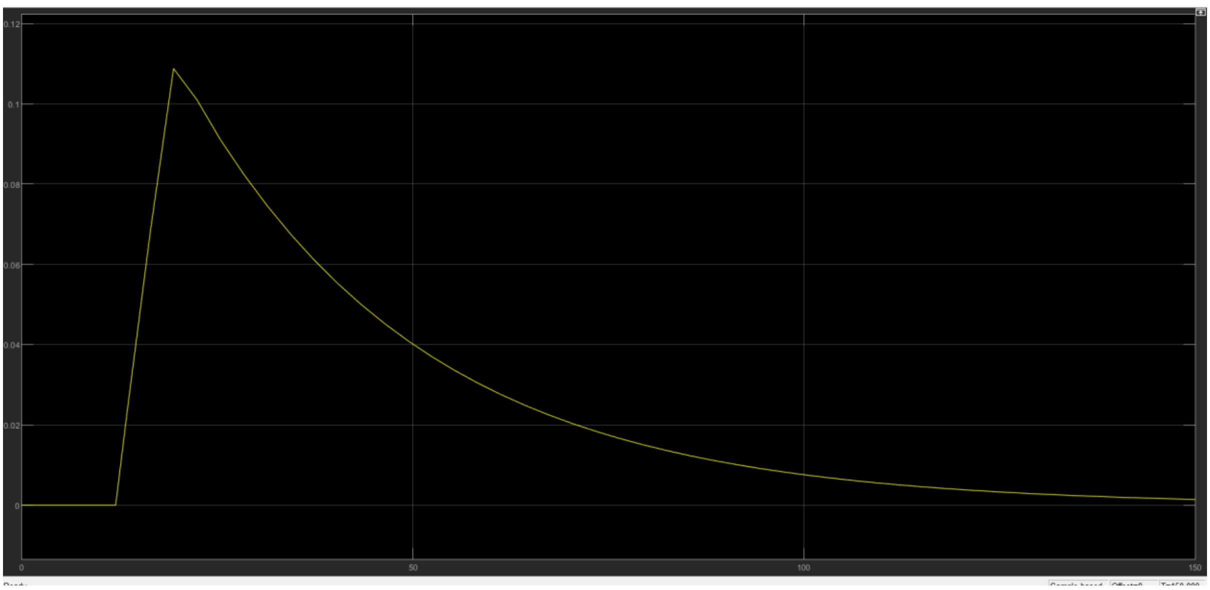

Figure 2: Disturbance rejection performance

Part b) Tuned PID Controller

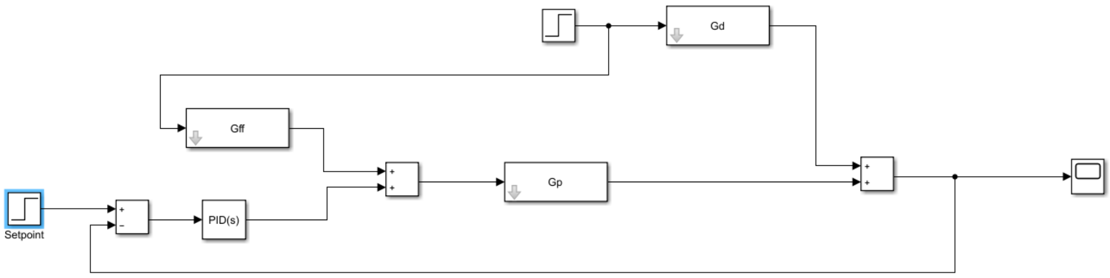

Figure 3: Feedforward in combination with a PID controller

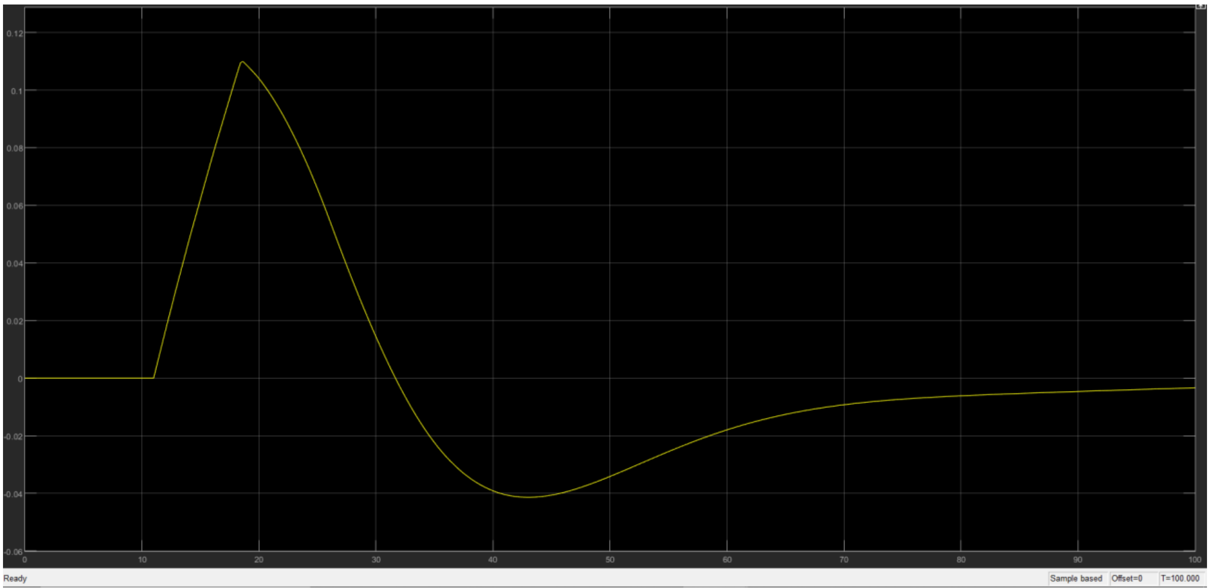

Figure 4: Response for combined efforts of feedforward and PID controller

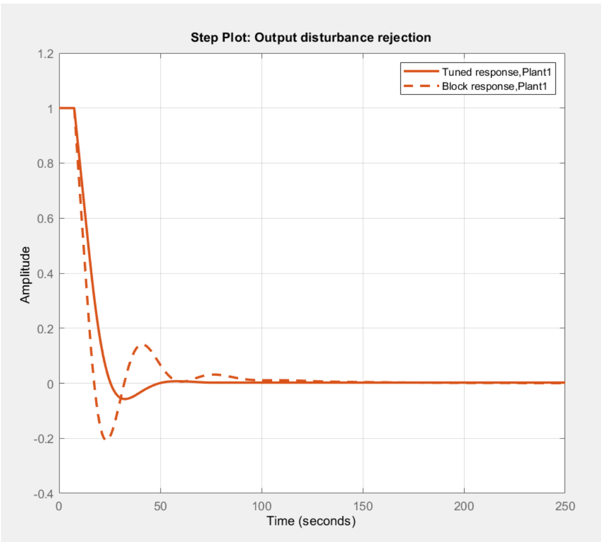

Figure 5: Tuning of the PID Controller (It linearizes the closed loop, and then we manually tune it on the basis of the responses/settling time requirements)

```
model = 'mixing_vessel';
load_system(model);
out = sim(model);
y = out.simout.data;
t = out.tout;
iae = trapz(t,abs(y));
```

Code for computing IAE

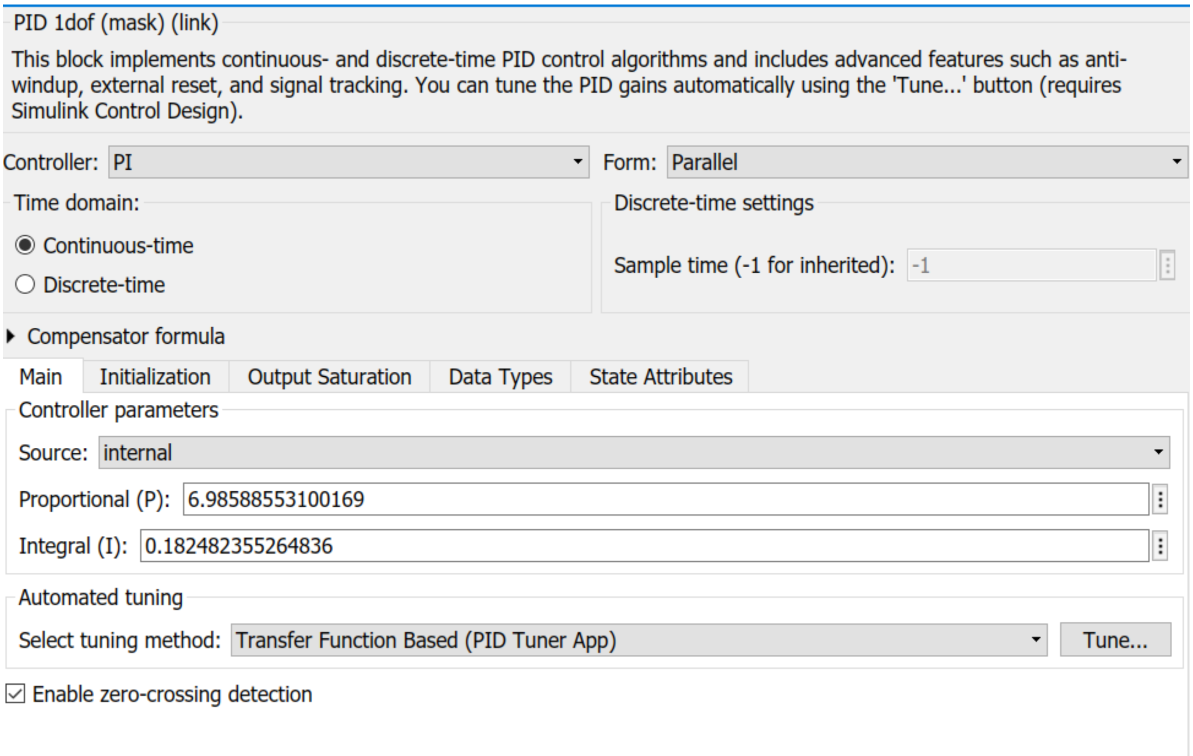

Figure 6: Tuned parameters

IAE with FF controller alone was obtained to be 3.5326

IAE with FF + PID controller was obtained to be 2.3346

As expected, we see an improvement when we use a PID controller in addition.

Part c) MPC

- Firstly, I will obtain the step response models of the process and disturbance using the corresponding transfer functions we have.
- Given the time delay and time constants, I will choose a sampling interval of about 2.5 minutes.
- In this way I can obtain step response model length as about 5*25/2.5 = 50.
- Given this n, and multiple delays involved I would want to have larger prediction horizon, **p = 25**. Roughly half of what we have for n. (Note that obviously if we go for p > n, the system might exhibit instability)

- It is always safe to have the control horizon to be smaller than the predictive horizon. We can probably have **m = 10-15.** And tune as per the response we get.
- Higher m is more aggressive but we need more computational power (because we need to optimize more variables).
- Input constraints can be decided based on the expected disturbance inputs that might occur. Let's say if 0.5 is the maximum expected disturbance (this causes a change of 0.25 in output) we can constrain the valve to have absolute value of input moves within 0.25/0.4 = 0.625 psig.