# 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);

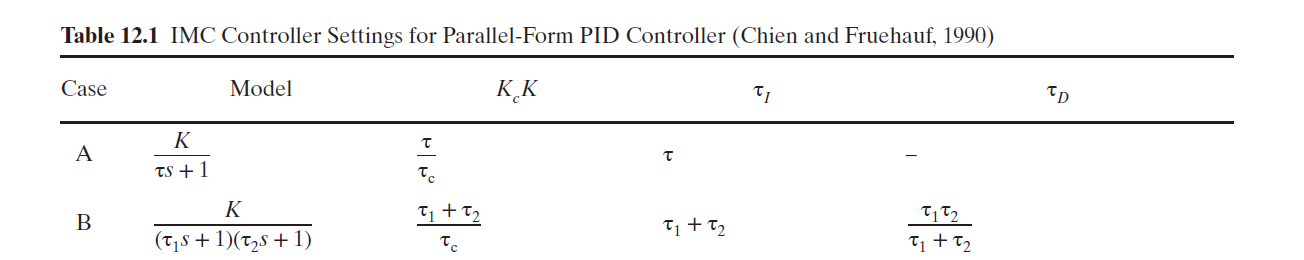
end

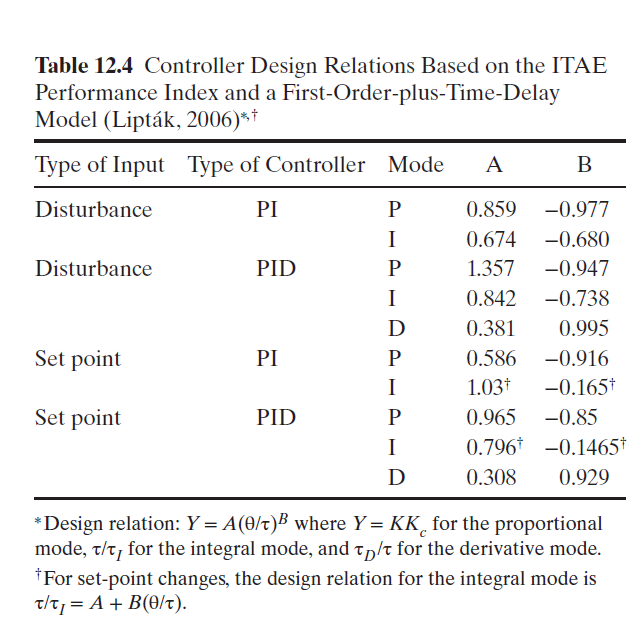
[val,loc] = min(r1);

lambda = lambdavec(loc);

# Question-2

## Tables





## 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;

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);

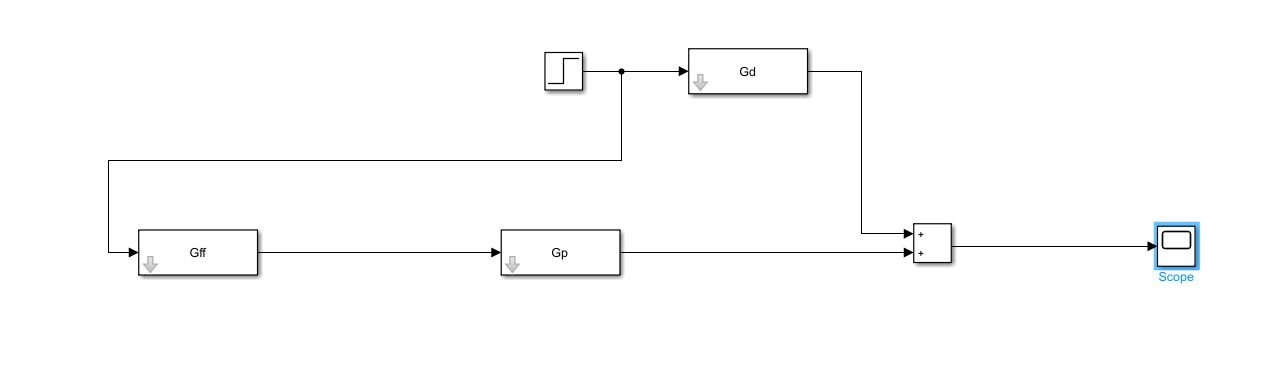
Gp = tf(K,conv([tau 1],[tau2 1]));

Y = step(Gp,tvec);

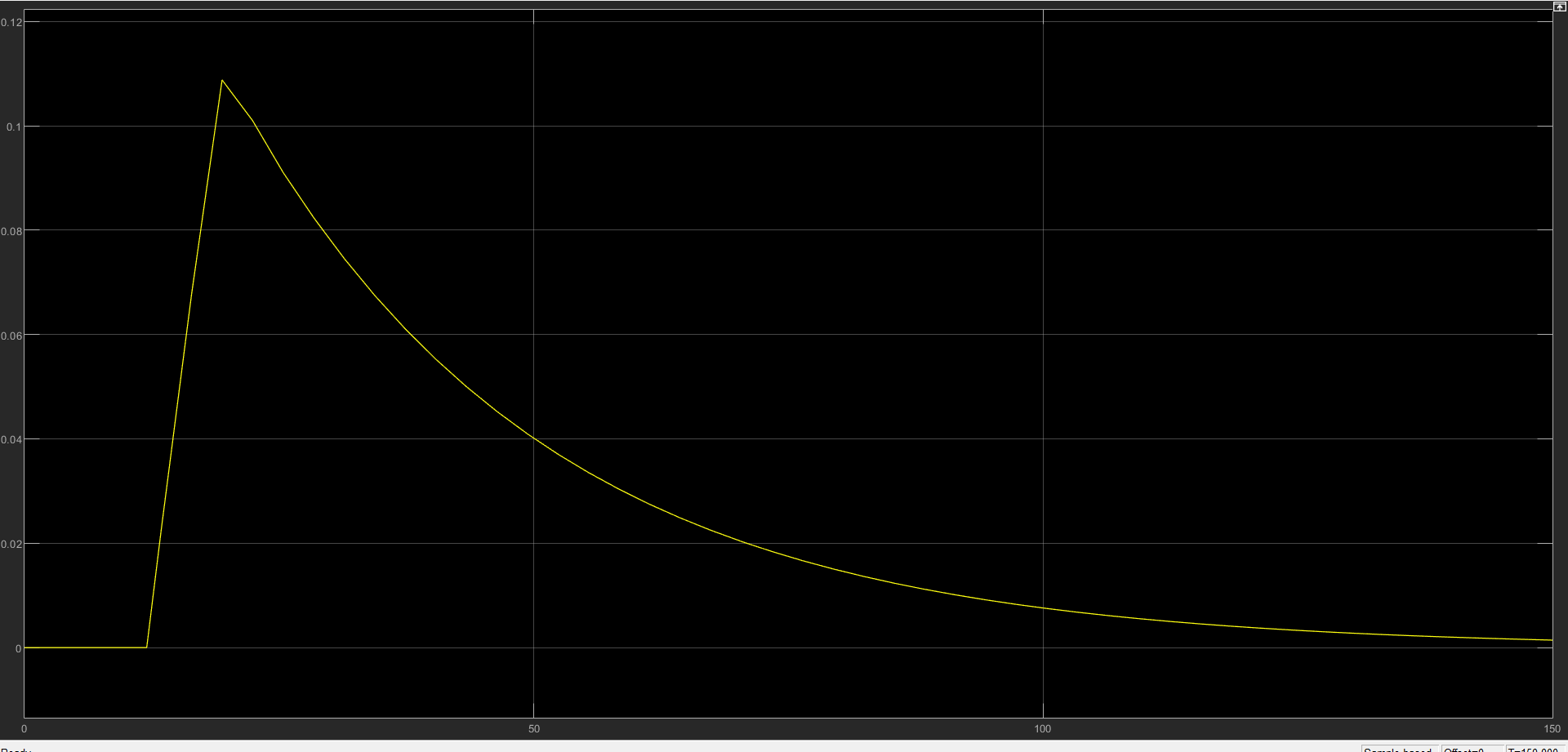
end

# Question-4

## Part a) Feedforward controller

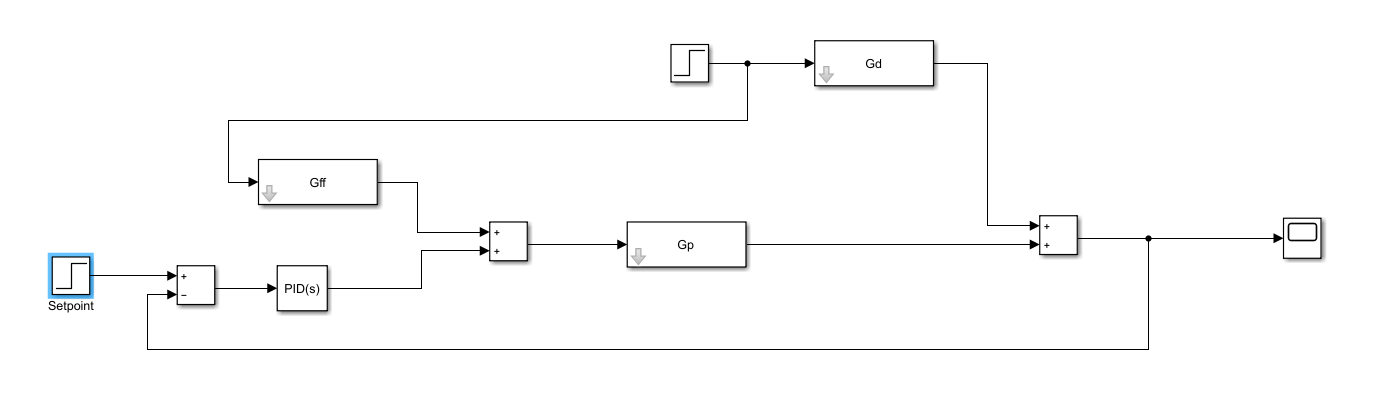


*Figure 1: SIMULINK DIAGRAM of the system with just a feed-forward controller*

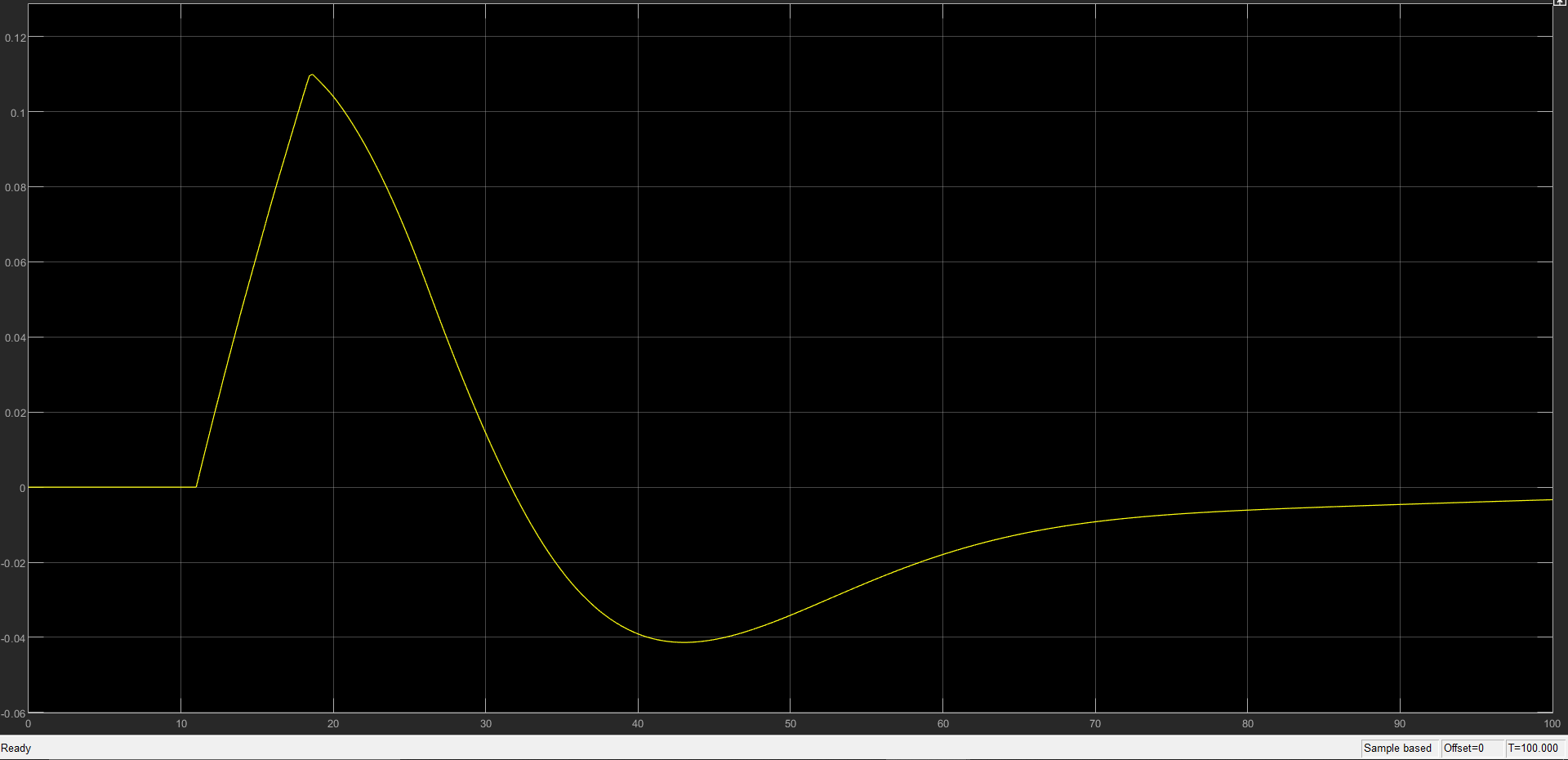


*Figure 2: Disturbance rejection performance*

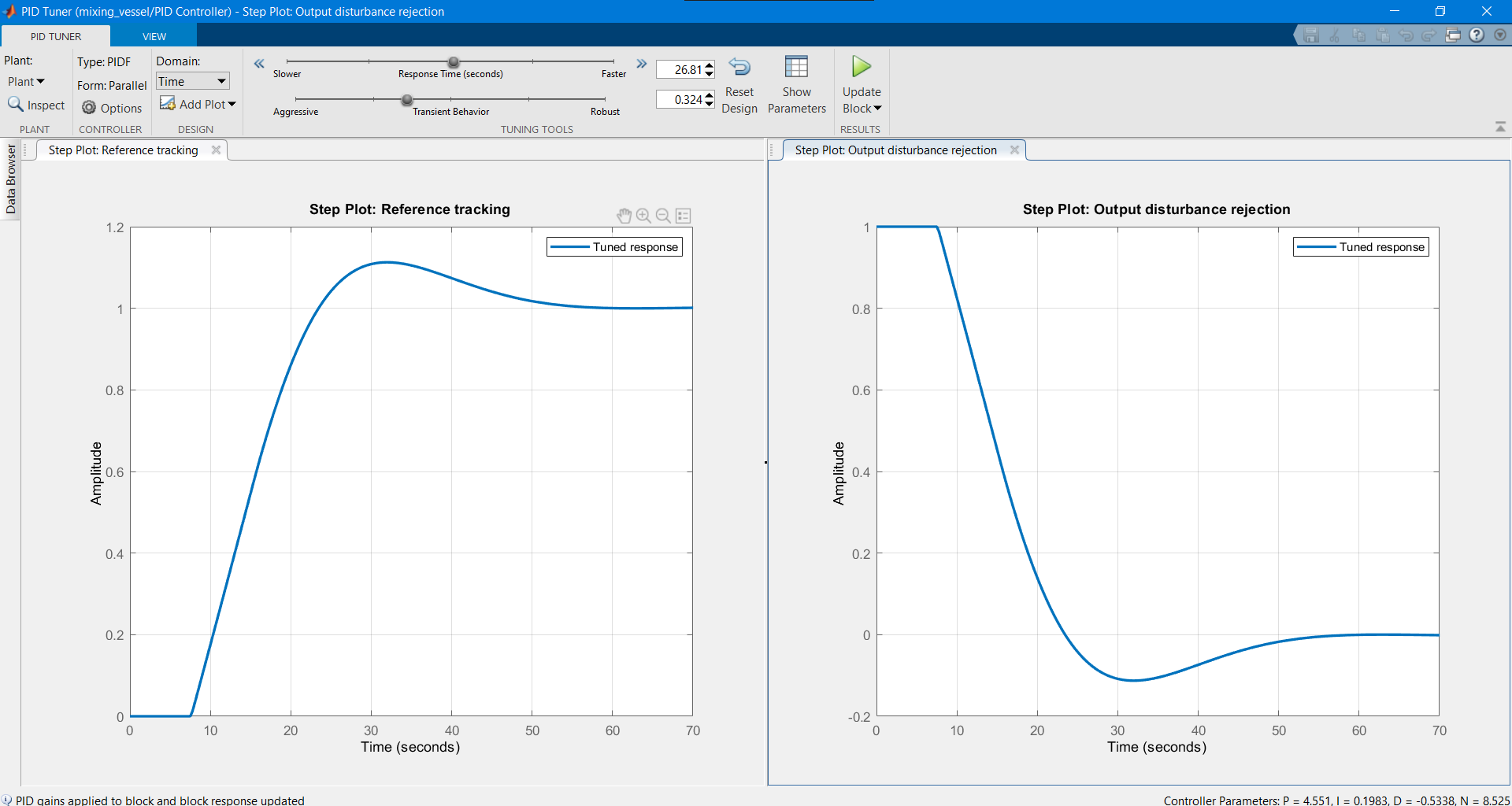
## Part b) Tuned PID Controller



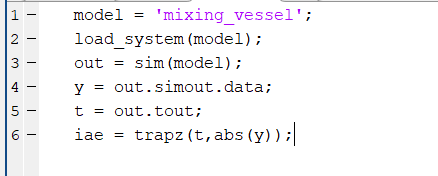
*Figure 3: Feedforward in combination with a PID controller*



*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)*

**

**IAE with FF controller** alonewas obtained to be 3.5326

**IAE with FF + PID controller** was obtained to be 2.3819

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 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.