Lab 07

Table of Contents

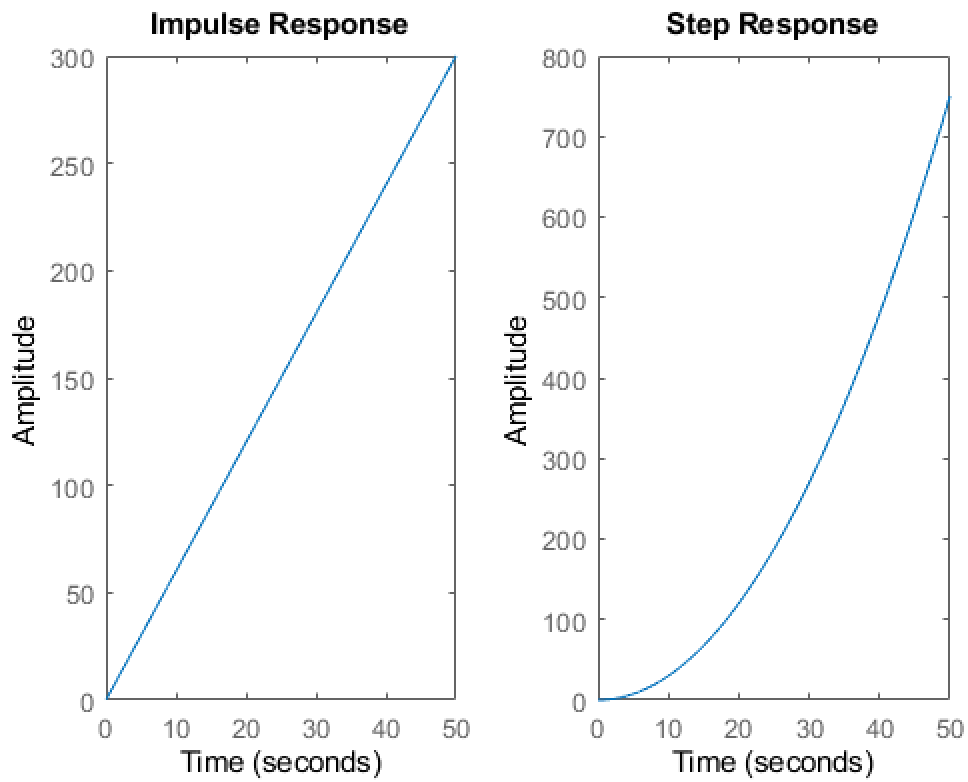
Constantin Gemmingen, Fabian Neumueller, Luke Peppard - Group 4.3

Prelab Q1

create the transfer function of the system (you should use the tf() function):

```
transBallDynamics = tf([30], [5 0 0])

% set step options to have a step-size of 0.1 instead of 1 (standard),
 use
% stepDataOptions()
opt_StepAmplitude01 = stepDataOptions('StepAmplitude', 0.1);

% open loop impulse and step response in one figure (two subplots):
figure(1)
subplot(1,2,1)
impulse(transBallDynamics)
subplot(1,2,2)
step(transBallDynamics, opt_StepAmplitude01)
```

```
transBallDynamics =

   30
  -----
  5 s^2

Continuous-time transfer function.
```



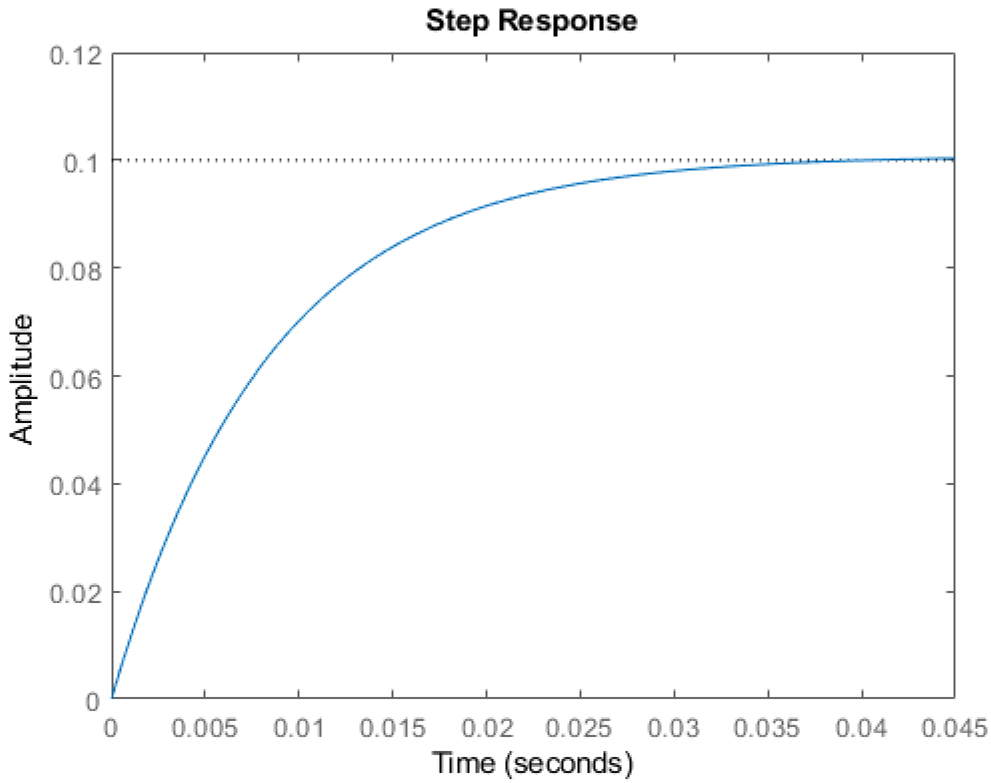
Prelab Q2

Controller Parameters:

```
K_P = 20;
K_I = 1;
K_D = 2;
K_D_ = 20;

% create the transfer function of the Controller:
transPID_Controller = tf([K_D_ K_P K_I], [1 0])
transForward = transPID_Controller * transBallDynamics;

% create the transfer function of the feedback loop
% trans_Q2_feedback = tf([6*K_D 6*K_P 6*K_I], [1 6*K_D 6*K_P 6*K_I]);
trans_Q2_feedback = feedback(transForward, 1)

% Plot the closed loop step response
figure(2)
step(trans_Q2_feedback, opt_StepAmplitude01)
% under damped using K_d = 2, and critically damped using K_d = 20;
```

```
transPID_Controller =

  20 s^2 + 20 s + 1
```

```
  -----------------
          s

Continuous-time transfer function.


trans_Q2_feedback =

      600 s^2 + 600 s + 30
  ----------------------------
  5 s^3 + 600 s^2 + 600 s + 30

Continuous-time transfer function.
```



Prelab Q3

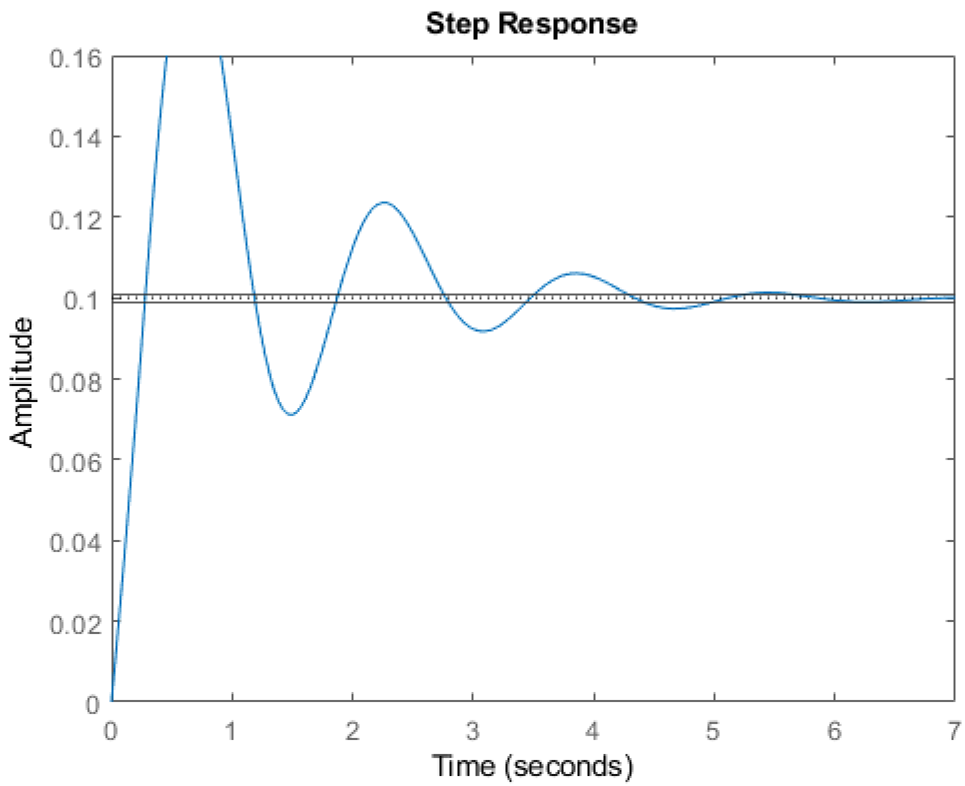
controller constants:

```
Kp = 0.4;
Ki = 0.05;
Kd = 0.54;

% create the transfer function of the Controller
transPID_ControllerDelay = tf([Kd Kp Ki], [1 0])
```

```
% Variables for delay:
DELAY=0.08; % 80ms
DELAY_=0.3; % 300ms

% create the transfer function of the delay in the system
% trans_Q3_delay = exp(-DELAY * s);
transDelay = tf(1, 1, 'InputDelay', DELAY_)


% create the transfer function of the feedback loop with delay
transForwardDelay = transPID_ControllerDelay * transBallDynamics;
transFeedbackDelay = feedback(transForwardDelay, transDelay)

% Plot the closed loop step response of the system with delay
y_top = 0.16;
y_bot = 0;
time_bot = 0;
time_top = 7;

figure(3)
step(transFeedbackDelay, opt_StepAmplitude01)
axis([time_bot time_top y_bot y_top])
yline(0.1+0.001)
yline(0.1-0.001)
% check design criteria for "steady state" error at 5 seconds
```

```
transPID_ControllerDelay =

  0.54 s^2 + 0.4 s + 0.05
  -----------------------
             s

Continuous-time transfer function.


transDelay =

  exp(-0.3*s) * (1)

Continuous-time transfer function.


transFeedbackDelay =

  A =
           x1      x2      x3
   x1   -3.24    -2.4    -0.3
   x2       1       0       0
   x3       0       1       0

  B =
       u1
   x1   2
```

```
   x2   0
   x3   0

  C =
         x1     x2     x3
   y1  1.62    1.2   0.15

  D =
       u1
   y1   0

  (values computed with all internal delays set to zero)

  Internal delays (seconds): 0.3

Continuous-time state-space model.
```



Prelab Q4

```
% Hint:
% - Make use of the linearized differential equations
GRAVITY = 9.81;

syms x(alpha)
eqation = diff(x, alpha) == 3/5 * GRAVITY;
```

```
solution = dsolve(eqation);
% - Filter out the first data points from the time delay to get rid of
 the
% incontinouity at the beginning of the sytem-response (otherwise the
 derivative explodes)
[x_output, t_output] = step(transFeedbackDelay, opt_StepAmplitude01);


% - Make sure to use the right units (rad, degree, ...) --> check if
 your
%  values make sense if you are not sure. E.g. a free-falling ball
% has 9.81 m/s^2 acceleration
```

Published with MATLAB® R2020a