(Intro to) Automatic Control Systems: December 2025 exam, Matlab simulation, Problem 2.d) PID control design

© 2026, Antonio Sala, Universitat Politècnica de València. All rights reserved.

Presentation in video: https://personales.upv.es/asala/YT/p2252dsEN.html

Process and Controller definitions

```
s=tf('s');
G=50/(s+10)/s;
M=15/(s+10)/s;

%question 2.d) ;PID control
Cd1=0.64*(s+10)*(s+32)/s %cancellation-based design
```

```
Cd1 =

  0.64 s^2 + 26.88 s + 204.8
  --------------------------
             s

Continuous-time transfer function.
Model Properties
```

```
Kd=102/50;Kp=3584/50;Ki=81920/50;
Cd2=Kd*s+Kp+Ki/s; %Non-cancellation design
%Three poles were assigned in Cd2
```

Closed-loop dynamics

```
%Filter=1;
```

With Filter=1, control action transfer function is non-realizable, we cannot implement direct "derivative" action, and we forcedly need a "high-frequency noise filter" in practice when derivative action is active.

```
Filter=1/(s/160+1); %10x desired CL poles LowPass filter
CLb1=closetheloop(G,M,Cd1*Filter);
pole(CLb1) %cancellation design
```

```
ans = 4×1 complex
10^2 ×
  -1.3037 + 0.0000i
  -0.1482 + 0.3221i
  -0.1482 - 0.3221i
  -0.1000 + 0.0000i
```

```
minreal(zpk(CLb1))
```

```
ans =

  From input "r" to output...
                5120 (s+32)
   y:  -------------------------------
```

```
        (s+130.4) (s^2 + 29.63s + 1257)

             102.4 s (s+32) (s+10)
    u:  -------------------------------
        (s+130.4) (s^2 + 29.63s + 1257)

                  s^2 (s+160)
    e:  -------------------------------
        (s+130.4) (s^2 + 29.63s + 1257)

  From input "d" to output...
                   15 s (s+160)
    y:  ---------------------------------------
        (s+130.4) (s+10) (s^2 + 29.63s + 1257)

                 -1536 (s+32)
    u:  -------------------------------
        (s+130.4) (s^2 + 29.63s + 1257)

                   -15 s (s+160)
    e:  ---------------------------------------
        (s+130.4) (s+10) (s^2 + 29.63s + 1257)

Continuous-time zero/pole/gain model.
Model Properties
```

*We have a problem when not filtering noise:

```
tf(CLb1(2,1))
```

```
ans =

  From input "r" to output "u":
  102.4 s^4 + 5325 s^3 + 7.578e04 s^2 + 3.277e05 s + 2.532e-08
  ------------------------------------------------------------
         s^4 + 170 s^3 + 6720 s^2 + 215040 s + 1.638e06

Continuous-time transfer function.
Model Properties
```

```
CLb2=closetheloop(G,M,Cd2*Filter);
pole(CLb2) %Three poles were assigned
```

```
ans = 4×1 complex
 -67.2991 +86.8319i
 -67.2991 -86.8319i
 -17.7009 +27.7976i
 -17.7009 -27.7976i
```

```
minreal(zpk(CLb2))
```

```
ans =

  From input "r" to output...
                16320 (s^2 + 35.14s + 803.1)
    y:  ---------------------------------------------
        (s^2 + 35.4s + 1086) (s^2 + 134.6s + 1.207e04)

            326.4 s (s+10) (s^2 + 35.14s + 803.1)
    u:  ---------------------------------------------
        (s^2 + 35.4s + 1086) (s^2 + 134.6s + 1.207e04)

                    s^2 (s+10) (s+160)
    e:  ---------------------------------------------
        (s^2 + 35.4s + 1086) (s^2 + 134.6s + 1.207e04)
```

```
  From input "d" to output...
                      15 s (s+160)
   y:  ---------------------------------------------
       (s^2 + 35.4s + 1086) (s^2 + 134.6s + 1.207e04)

                -4896 (s^2 + 35.14s + 803.1)
   u:  ---------------------------------------------
       (s^2 + 35.4s + 1086) (s^2 + 134.6s + 1.207e04)

                     -15 s (s+160)
   e:  ---------------------------------------------
       (s^2 + 35.4s + 1086) (s^2 + 134.6s + 1.207e04)

Continuous-time zero/pole/gain model.
Model Properties
```

We didn't solve the problem if no filter were applied:

```
tf(CLb2(2,1))
```

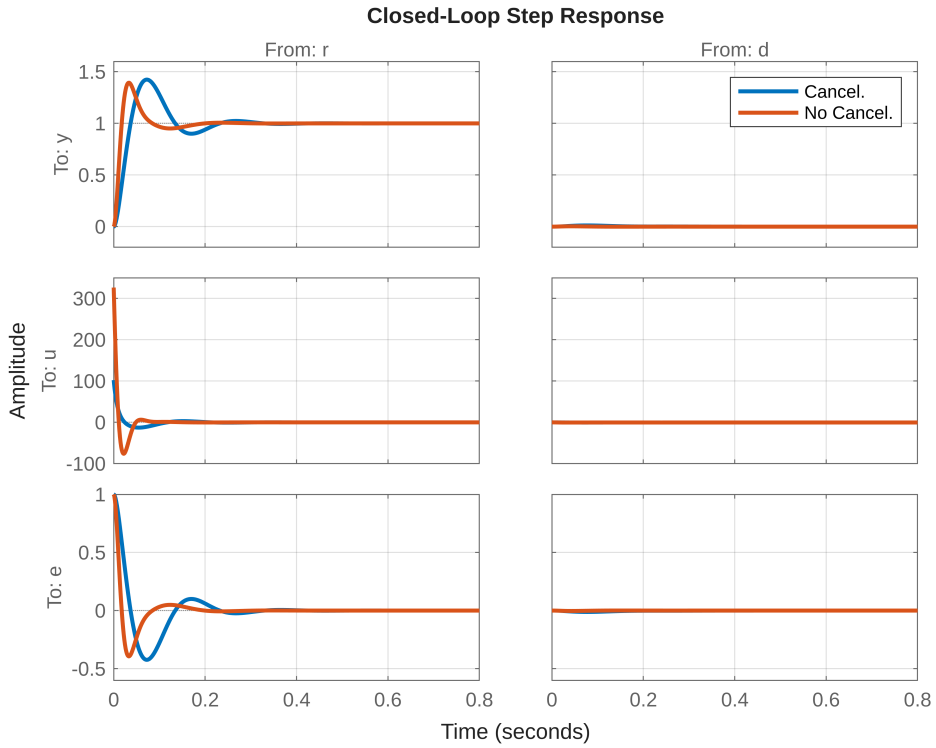
```
ans =

  From input "r" to output "u":
  326.4 s^4 + 1.473e04 s^3 + 3.768e05 s^2 + 2.621e06 s
  ----------------------------------------------------
  s^4 + 170 s^3 + 1.792e04 s^2 + 5.734e05 s + 1.311e07

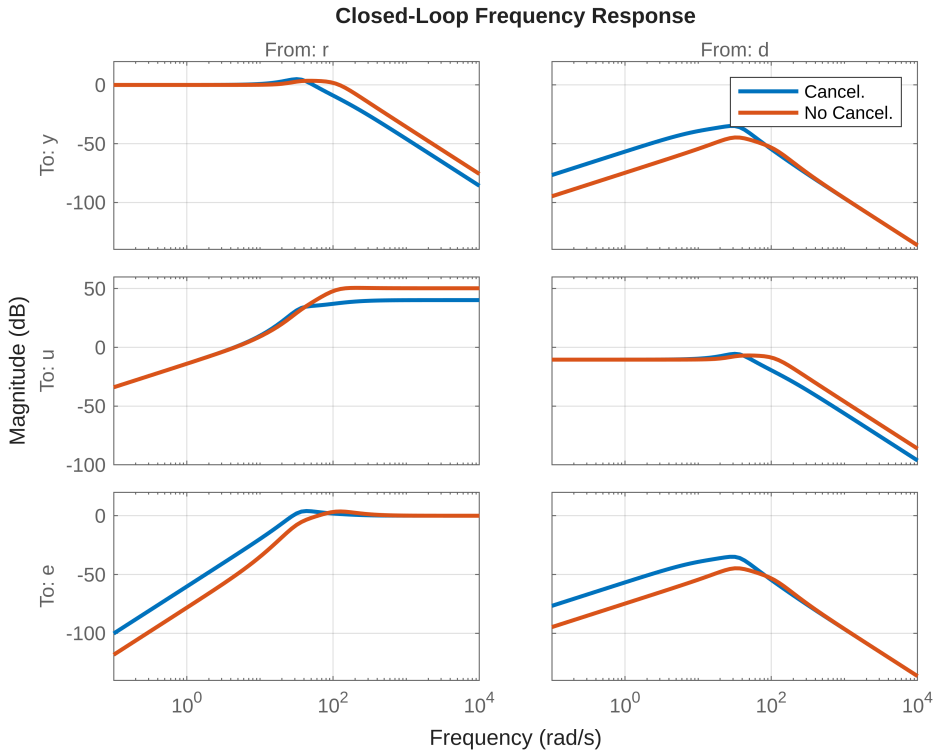
Continuous-time transfer function.
Model Properties
```

Time and Frequency response

```
figure(3)
h=stepplot(CLb1([1 2 3],:),CLb2([1 2 3],:));
h.Responses(1).LineWidth=2;
h.Responses(2).LineWidth=2;
grid on
title('Closed-Loop Step Response');
legend('Cancel.', 'No Cancel.');
```

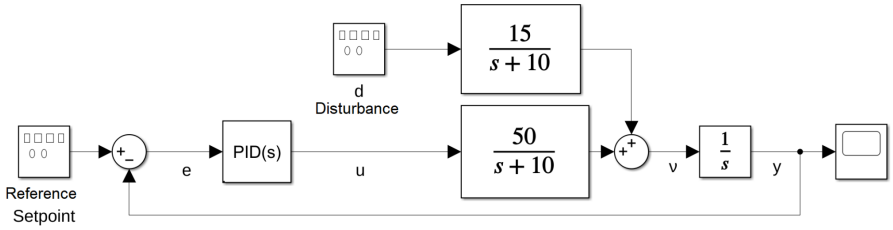



```
figure(4)
bodemag(CLb1,CLb2);
h=gca;
h.Responses(1).LineWidth=2;
h.Responses(2).LineWidth=2;
grid on
title('Closed-Loop Frequency Response');
legend('Cancel.', 'No Cancel.');
```

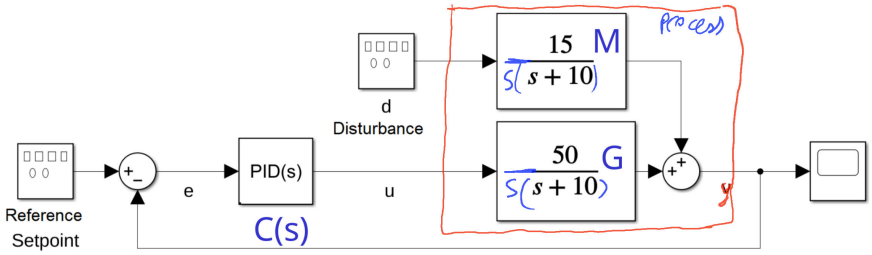


Auxiliary functions

*We will convert



into this one below, to use the exact same code as in the PI speed control case:



*__Note:__ there is something subtly wrong in the above diagram. Indeed, the block diagram duplicates the unstable $1/s$ so the block diagram as such will be "*internally unstable*" (the unnamed output of $M(s)$ may **drift** in an "*uncontrollable*" way as it is an unstable subsystem outside of a control loop). However, *algebraically* in the Laplace domain when solving for (e, y, u) that thing did not matter in the pen-and-paper examination, and "minreal" here will sort it out as well. Duplicating unstable things might produce unbounded internal variables even if the instability is not observed in the outputs. Details omitted for brevity.

```
function CL=closetheloop(G,M,C)
G.InputName="u";
M.InputName="d";
Process=[G M]; % y=Gu+Md, as a 1x2 transfer function matrix
Process.OutputName="y";
C.InputName="e"; C.OutputName="u";
S=sumblk("e=r-y");
CL = connect(Process, C, S, {'r','d'}, {'y','u','e'});
end
```