# Control System Design

Jeremiah Mahler <jmmahler@gmail.com>
December 21, 2013

# Contents

# 1 Introduction

This document is a collection of notes on control system design. The author collected these while taking several control systems classes at California State University Chico taught by Dr. Adel Ghandakly. The first was an Introduction to Control Systems (EECE 482) and the second was Computer Control of Dynamic Systems (EECE 682). The former focused on continuous systems whereas the latter focused on digital systems suitable for implentation in a computer.

Matlab code is included for most examples. They have been tested with Octave,[1] an open source Matlab equivalent.

[1] Octave community. *GNU/Octave*. 2012. URL: `www.gnu.org/software/octave/`.

## 2 Common Plants/Systems

Plants denoted by $G(s)$, systems by $H(s)$ and controllers by $D(s)$.

| | |
|---|---|
| $G(s) = \frac{1}{s(s+1)}$ | Pole Placement (Section 6), ZOH (Section 15), Direct Design ($K$) (Section 20) |
| $G(s) = \frac{1}{s(s+7)}$ | |
| $G(s) = \frac{a}{s(s+a)}$ | Mapping (Section 16.1), ZOH (Section 15) |
| $G(s) = \frac{1}{(s+1)^2}$ | |
| $G(s) = \frac{1}{s(s+0.4)}$ | PID (Section 9), ZOH (Section 15) |
| $G(s) = \frac{1}{s^2}$ | ZOH (Section 15), Direct Design (Section 20), double integrator |
| $G(s) = \frac{10}{(s+0.1)(s+0.2)}$ | |
| $G(z) = 0.0484\frac{z+0.9672}{(z-1)(z-0.9048)}$ | antenna model[2] |
| $H(s) = \frac{10s+1}{1+s}$ | |
| $H(s) = \frac{20}{s^2+4s+20}$ | System From Specs (Section 4), Pole Placement (6) |
| $H(s) = \frac{10}{s^2+s+1}$ | |
| $\frac{B_m}{A_m} = \frac{0.62z-0.3}{z^2-1.2z+0.52}$ | Model Matching (Section 21)[3] |
| $D(s) = \frac{149s+880}{s+47}$ | Pole Placement (Section 6) |
| $D(s) = \frac{(s+7)}{1.5+2.5+s^2}$ | |
| $D(s) = \frac{20s^2+20s}{s^2+4s}$ | Pole Placement (Section 6) |

[2] G.F. Franklin, J.D. Powell, and M.L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley world student series. Addison-Wesley Longman, Incorporated, 1998. ISBN: 9780201331530, Pg. 261.

[3] K. Ogata. *Discrete-Time Control Systems*. Prentice Hall International editions. Prentice-Hall International, 1995. ISBN: 9780133286427, Pg. 532.

# 3 Laplace Transform

9/8/13

**Example 1**

Laplace of unit step

(e.g.)

$$\mathcal{L}\{u(t)\} = \int_0^\infty e^{-st} \cdot u(t)\, dt$$

$$= \int_0^\infty e^{-st}\, dt$$

$$= -\frac{1}{s}\int_0^\infty e^{-st} \cdot (-s)\, dt$$

$$= -\frac{1}{s} \cdot e^{-s \cdot t}\Big|_0^\infty$$

$$= -\frac{1}{s} \cdot \left[e^{-\infty} - e^{0}\right]$$

$$= -\frac{1}{s}\left[0 - 1\right]$$

$$\boxed{\mathcal{L}\{u(t)\} = \frac{1}{s}} \quad \checkmark$$

$\int e^u\, du = e^u$

$u = -st$

$du = (-s)\, dt$

## 4 Systems From Specifications

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1}$$

There are different ways to find $\omega_n$ and $\zeta$. The following specs for the rise time ($T_r$), over shoot ($\%OS$), and setting time ($T_s$) are one example.

$$T_r = \frac{2.22}{\omega_n}$$
$$\%OS = \left(1 - \frac{\zeta}{0.6}\right) \cdot 100$$
$$T_s = \frac{4}{\zeta\omega_n}$$

**Example 1**

9/4/13

System From Specs

$T_r = 0.5$

$\%OS = 25$

---

$$T_r = \frac{2.22}{\omega_n} = 0.5$$

$$\omega_n = 4.44$$

$$\%OS = \left(1 - \frac{\zeta}{0.6}\right) \cdot 100 = 25$$

$$-\frac{\zeta}{0.6} = \frac{25}{100} - 1$$

$$\zeta = -0.6\left[\frac{25}{100} - 1\right]$$

$$\zeta = 0.450$$

$$H(s) = \frac{\omega_n^2}{s^2 + 2 \cdot \zeta \cdot \omega_n + \omega_n^2}$$

$$\boxed{H(s) = \frac{19.714}{s^2 + 3.996 \cdot s + 19.714}}$$

The step response of this system is shown in Figure 1. It can be seen that the specs are met when a unit step is applied. Interestingly, these specs are satisfied even if the input signal is scaled and shifted as shown by Figure 2.

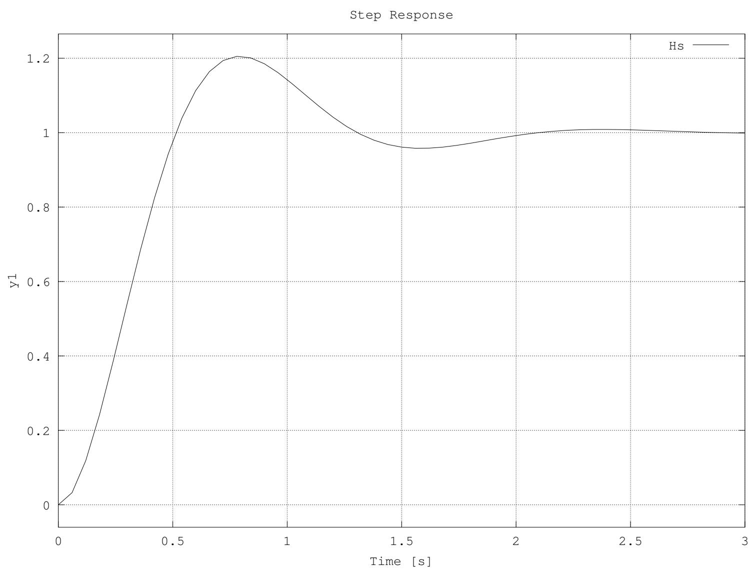



Figure 1: Step response for system with $Tr = 0.5$ and $\%OS = 25$.

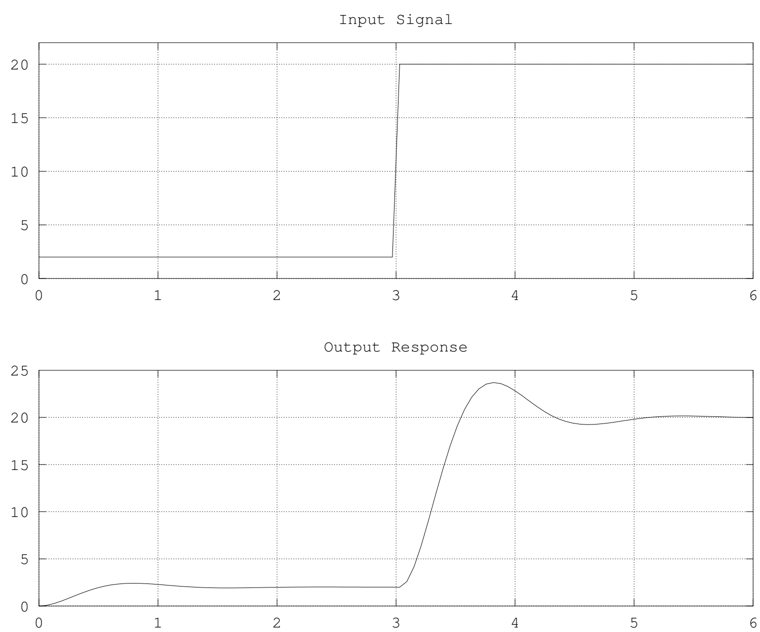



Figure 2: Shifted and scaled step response with $Tr = 0.5$ and $\%OS = 25$ (same as previous).

# 5 Final Value Theorem

<u>Final Value Theorem</u>

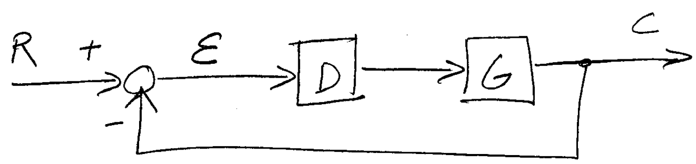
10/2/12

[Ogata 199]
[Franklin 225]

$$\lim_{t \to \infty} f(t) = \lim_{s \to 0} s \cdot F(s)$$

$$= \lim_{z \to 1} (1 - z^{-1}) \cdot F(z)$$

| R | s | z |
|---|---|---|
| step | $\frac{1}{s}$ | $\frac{1}{1-z^{-1}}$ |
| ramp | $\frac{1}{s^2}$ | $\frac{T \cdot z^{-1}}{(1-z^{-1})^2}$ |
| accel | $\frac{1}{s^3}$ | |




$$\mathcal{E} = R \cdot \frac{1}{1+DG}$$

<u>s-domain</u>

$$\mathcal{E}_{ss} = \lim_{s \to 0} s \cdot R \cdot \frac{1}{1+DG}$$

<u>z-domain</u>

$$\mathcal{E}_{ss} = \lim_{z \to 1} (1 - z^{-1}) R \cdot \frac{1}{1+DG}$$

$$\mathcal{E}_{ss_{step}} = \lim_{z \to 1} \cancel{(1 - z^{-1})} \frac{1}{\cancel{(1-z^{-1})}} \cdot \frac{1}{1+DG}$$

$$= \frac{1}{1+K_p}$$

$$\mathcal{E}_{ss_{ramp}} = \lim_{z \to 1} (1 - z^{-1}) \frac{T \cdot z^{-1}}{(1-z^{-1})^2} \cdot \frac{1}{1+DG}$$

$$= \lim_{z \to 1} \frac{T \cdot z^{-1}}{(1-z^{-1})(1+DG)} = \frac{1}{K_v}$$

# 6 Pole Placement (S-Domain)

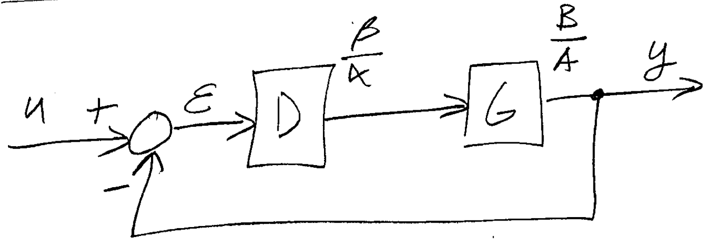
**Example 1**

Pole Placement

9/17/12

$$G(s) = \frac{1}{s(s+1)} \qquad H(s) = \frac{20}{s^2+4s+20}$$

$\omega_n = 4.44$




$$\frac{y}{u} = H = \frac{DG}{1+DG}$$

$$= \frac{\beta B}{\alpha A + \beta B}$$

place the poles

$$\alpha A + \beta B = s^2 + 4s + 20$$

$$(\alpha_1 s + \alpha_0)\cdot s(s+1) + (\beta_1 s + \beta_0) = s^2 + 4s + 20$$

3rd order

2nd order

Add root $(\omega_n \cdot 10)$ to correct order

$$(\alpha_1 s + \alpha_0) s(s+1) + \beta_1 s + \beta_0 = (s^2+4s+20)(s+44.4)$$

$$\alpha_1 s^3 + (\alpha_0 + \alpha_1) s^2 + (\alpha_0 + \beta_1) s + \beta_0$$

$$= s^3 + 48 s^2 + 196 s + 880$$

(cont)

(cont)

$$\alpha_1 s^3 + (\alpha_0 + \alpha_1)s^2 + (\alpha_0 + \beta_1)s + \beta_0 = s^3 + 48s^2 + 196s + 880$$

(match polynomials)

$\alpha_1 = 1$

$\alpha_0 + \alpha_1 = 48$

$\alpha_0 = 47$

$\beta_0 = 880$

$\alpha_0 + \beta_1 = 196$

$\beta_1 = 196 - 47$

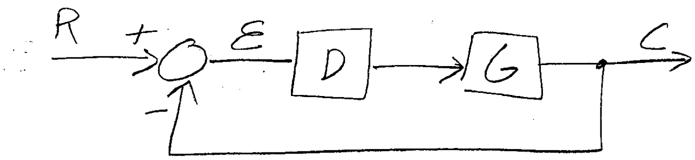
$\beta_1 = 149$

$$D(s) = \frac{\beta}{\alpha} = \frac{\beta_1 s + \beta_0}{\alpha_1 s + \alpha_0}$$

$$\boxed{D(s) = \frac{149s + 880}{s + 47}}$$

## Steady State Error, step

$$G(s) = \frac{1}{s(s+1)} \qquad D(s) = \frac{149s+880}{s+47}$$




$$H = \frac{C}{R} = \frac{DG}{1+DG} \qquad \varepsilon = \frac{1}{1+DG} = 1 - H$$

$$\varepsilon_{ss_{step}} = \lim_{s \to 0} \cdot s \cdot R \cdot \frac{1}{1+DG} \qquad R = \frac{1}{s} \quad (step)$$

$$= \lim_{s\to 0} s \cdot \frac{1}{s} \cdot \frac{1}{1+DG}$$

$$= \lim_{s \to 0} \frac{1}{1+DG}$$

$$= \lim_{s\to 0} \frac{1}{1 + \frac{149s+880}{(s+47)\,s(s+1)}}$$

numerator → 0

→ 0

$$= \lim_{s\to 0} \frac{(s+47)\cdot s\,(s+1)}{(s+47)\,s\,(s+1) + 149s + 880}$$

$$\boxed{\varepsilon_{ss_{step}} = 0}$$

The step response of this system is show in Figure 3. The Matlab code used to produce this plot is shown in Listing 1. It can be seen that it agrees with calculated steady state error.

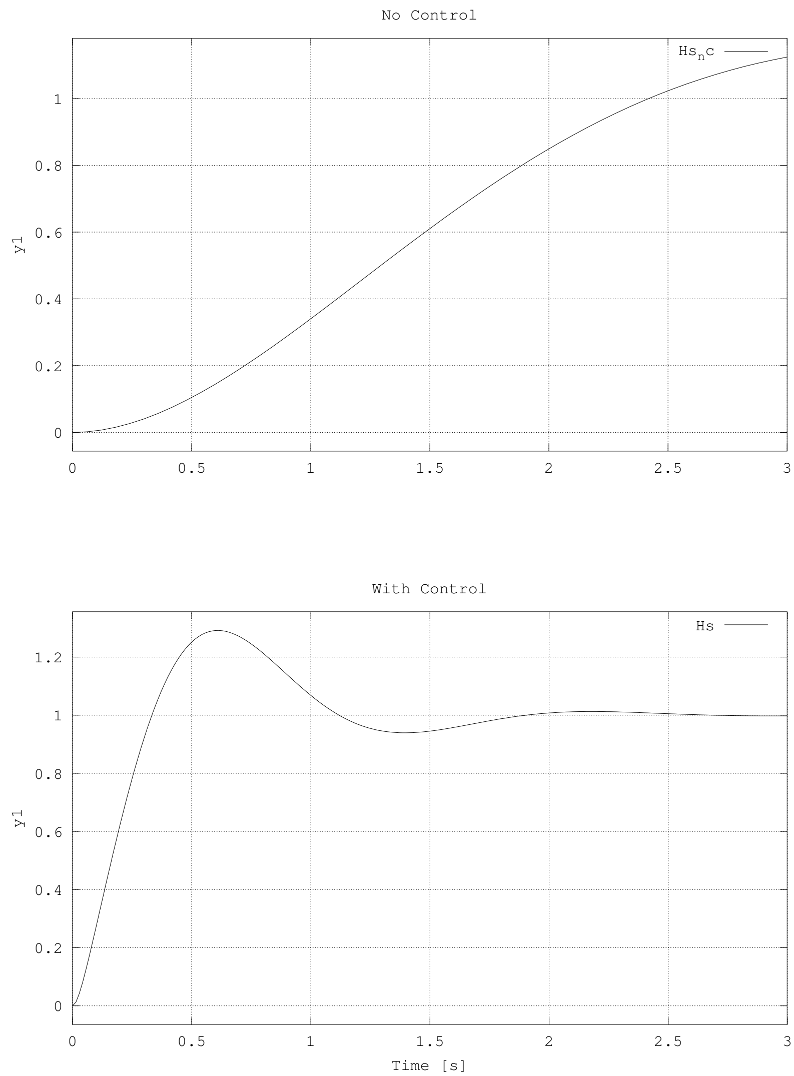


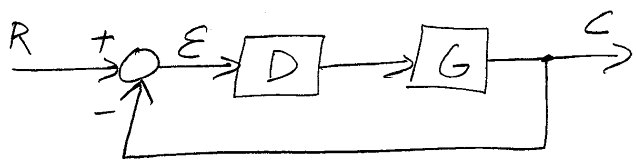

Figure 3: Step response with no control and with controller built using Pole Placement.

```
%
% pps2s1_plot.m
%
% Pole Placement
%

T_end = 3;  % time final (sec)

Gs = tf([1], [1 1 0]);
Ds = tf([149 880], [1 47]);

Hs_nc = (Gs)/(1 + Gs);  % no control
Hs        = (Ds*Gs)/(1 + Ds*Gs);

figure;
orient tall;

subplot(2,1,1);
step(Hs_nc, T_end);
title('No Control');
xlabel('');

subplot(2,1,2);
step(Hs, T_end);
title('With Control');

print('pps2s1_plot.eps', '-deps');
```

Listing 1: Matlab script to plot step response of Pole Place controller.

Steady State Error, ramp

$$G(s) = \frac{1}{s(s+1)} \qquad D(s) = \frac{149s + 880}{s + 47}$$




$$H = \frac{C}{R} = \frac{DG}{1+DG} \qquad \mathcal{E} = \frac{1}{1+DG} = 1 - H$$

$$\mathcal{E}_{ss} = \lim_{s\to 0} s \cdot R \cdot \frac{1}{1+DG} \qquad R = \frac{1}{s^2} \quad \text{(ramp)}$$

$$= \lim_{s\to 0} s \cdot \frac{1}{s^2} \cdot \frac{1}{1+DG}$$

$$= \lim_{s\to 0} \frac{1}{s} \cdot \frac{1}{1+DG}$$

$$= \lim_{s\to 0} \frac{1}{s} \cdot \frac{1}{1 + \frac{149s+880}{(s+47)\,s(s+1)}}$$

$$= \lim_{s\to 0} \frac{1}{\cancel{s}} \cdot \frac{(s+47)\cancel{s}(s+1)}{(s+47)s(s+1) + 149s + 880}$$

$$= \lim_{s\to 0} \frac{(s+47)(s+1)}{s^3 + 48s^2 + 47s + 149s + 880}$$

$$= \frac{47}{880}$$

$$\boxed{\mathcal{E}_{ss\,\text{ramp}} = 0.053}$$

The ramp response of this system is show in Figure 4. The Matlab code used to produce this plot is shown in Listing 2. It can be seen that it agrees with calculated steady state error.

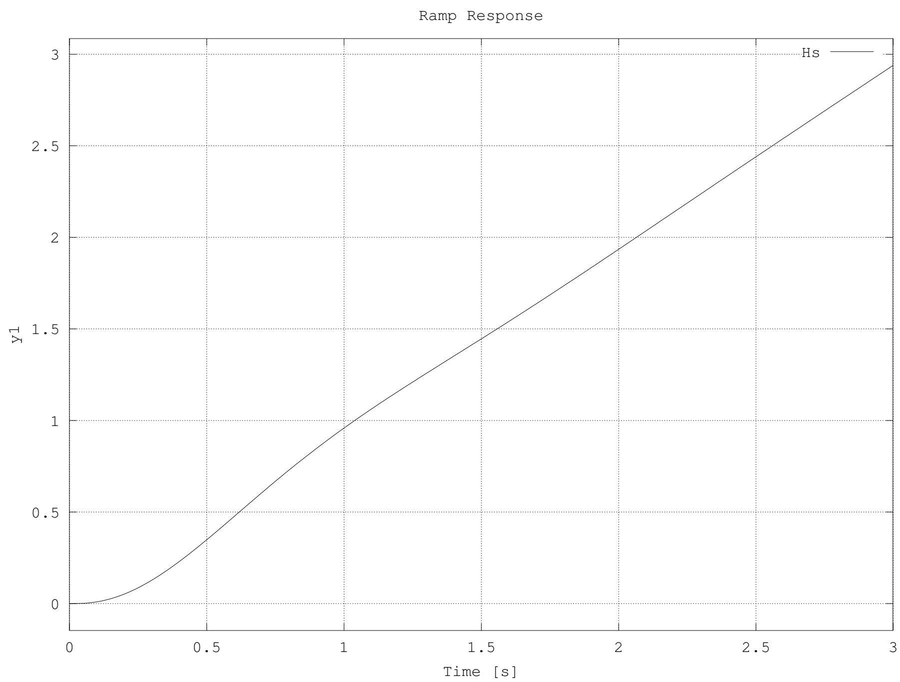



Figure 4: Ramp response of controller built using Pole Placement.

```
%
% pps2s1r_plot.m
%
% ramp response
%

Tf = 3;  % time final (sec)

Gs = tf([1], [1 1 0]);
Ds = tf([149 880], [1 47]);

Hs = (Ds*Gs)/(1 + Ds*Gs);

ramp(Hs, Tf);

print('pps2s1r_plot.eps', '-deps');
```

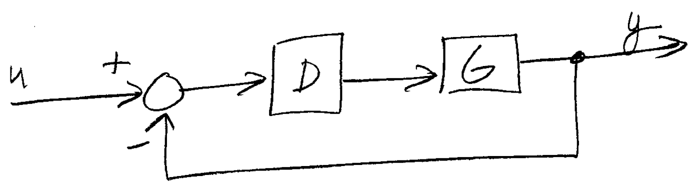
Listing 2: Matlab script to plot ramp response of Pole Place controller.

**Example 2**

Algebraic

11/22/13

$$G(s) = \frac{1}{s(s+1)} \qquad H(s) = \frac{20}{s^2+4s+20}$$




$$\frac{y}{u} = H(s) = \frac{DG}{1+DG}$$

(solve for D)

$$D = \frac{H}{G - HG}$$

$$= \frac{H}{G} \cdot \frac{1}{1-H}$$

$$= \frac{20 \cdot s(s+1)}{(s^2+4s+20)} \cdot \frac{1}{\left(1 - \frac{20}{s^2+4s+20}\right)}$$

$$= \frac{(20s^2+20s)\cancel{(s^2+4s+20)}}{\cancel{(s^2+4s+20)}(s^2+4s+\cancel{20}-\cancel{20})}$$

$$\boxed{D(s) = \frac{20s^2+20s}{s^2+4s}}$$

The step response of this system is show in Figure 5. The Matlab code used to produce this plot is shown in Listing 3.

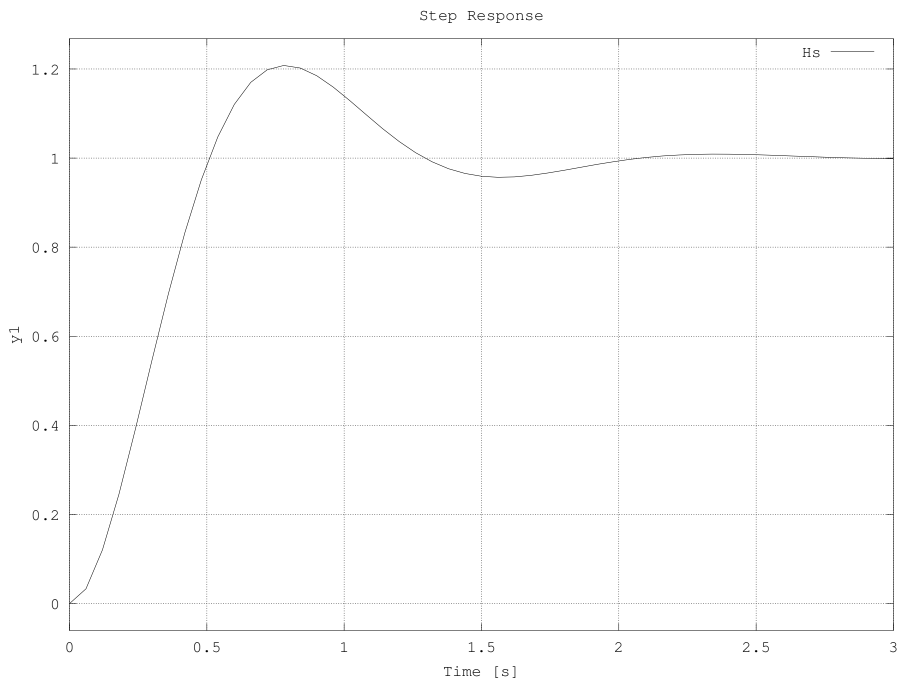


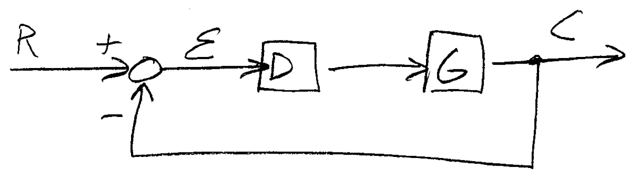

Figure 5: Step response of controller found using algebraic methods.

```
%
% pps2s1a_plot.m
%
% Pole Placement
%

Tf = 3;  % time final (sec)

Gs = tf([1], [1 1 0]);
Ds = tf([20 20 0], [1 4 0]);

Hs = (Ds*Gs)/(1 + Ds*Gs);

step(Hs, Tf);

print('pps2s1a_plot.eps', '-deps');
```

Listing 3: Matlab script to plot step response of controller found using algebraic methods.

steady state error, ramp

$$G(s) = \frac{1}{s(s+1)} \qquad D(s) = \frac{20s^2+20s}{s^2+4s}$$




$$\frac{C}{R} = H = \frac{DG}{1+DG} \qquad \mathcal{E} = \frac{1}{1+DG} = 1-H$$

$$\mathcal{E}_{ss_{ramp}} = \lim_{s\to 0} s \cdot R \cdot \frac{1}{1+DG} \qquad \frac{1}{s^2} \quad \text{(ramp)}$$

$$= \lim_{s\to 0} s \cdot \frac{1}{s^2} \cdot \frac{1}{1+DG}$$

$$= \lim_{s\to 0} \frac{1}{s} \frac{1}{1+\frac{20s^2+20s}{(s^2+4s)\,s\,(s+1)}}$$

$$= \lim_{s\to 0} \frac{1}{\cancel{s}} \frac{(s^2+4s)\cancel{s}(s+1)}{(s^2+4s)s(s+1) + 20s^2+20s}$$

$$= \lim_{s\to 0} \frac{(s^2+4s)(s+1)}{(s^2+4s)\cdot s(s+1) + 20s^2+20s}$$

$$= \lim_{s\to 0} \frac{\cancel{s}(s+4)(s+1)}{\cancel{s}\left[(s^2+4s)(s+1)+20s+20\right]}$$

$$= \frac{4}{20}$$

$$\boxed{\mathcal{E}_{ss_{ramp}} = 0.200}$$

The ramp response of this system is show in Figure 6. The Matlab code used to produce this plot is shown in Listing 4. It can be seen that the calculated steady state error for a ramp response agrees with the plot.

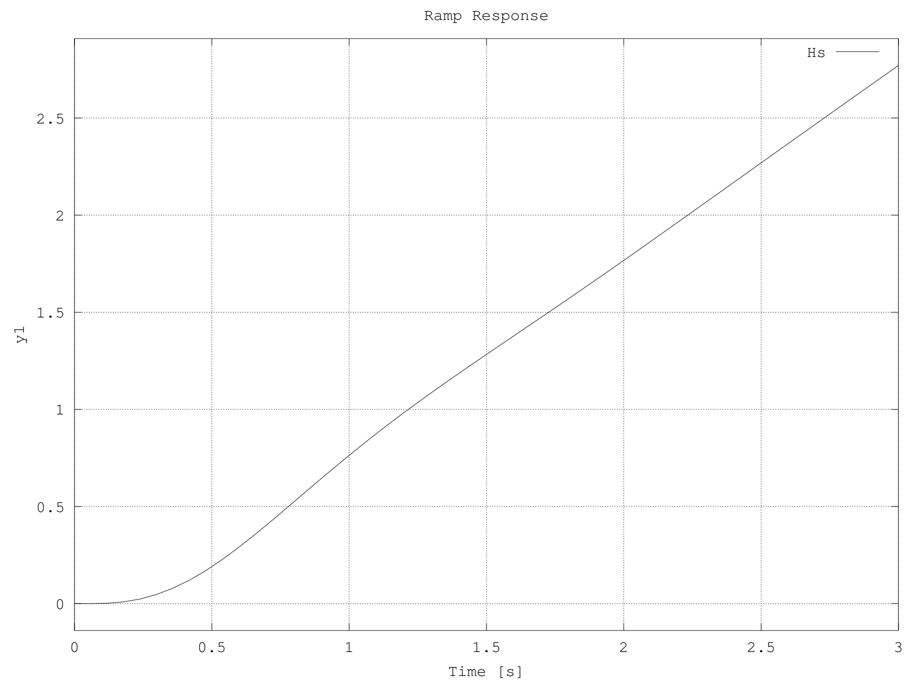



Figure 6: Ramp response of controller found using algebraic methods.

```
%
% pps2s1ar_plot.m
%
% ramp response
%

Tf = 3;  % time final (sec)

Gs = tf([1], [1 1 0]);
Ds = tf([20 20 0], [1 4 0]);

Hs = (Ds*Gs)/(1 + Ds*Gs);

ramp(Hs, Tf);

print('pps2s1ar_plot.eps', '-deps');
```

Listing 4: Matlab script to plot ramp response of controller found using algebraic methods.

# 7 Pole Placement, Diophantine

# 8 Steady State Performance/Error

See Section 6 for examples with steady state error ($\varepsilon_{ss}$).

# 9 PID Controller Design

**Example 1**

PID

9/4/13

$$G(s) = \frac{B}{A} = \frac{\cancel{b_1 \cdot s} + b_o}{s^2 + a_1 \cdot s + a_o} = \frac{1}{s^2 + s(0.4)}$$

(roots of $H(s)$)

$$(s^2 + 3.996 \cdot s + 19.714)(s + 44.4) = d_3 s^3 + d_2 s^2 + d_1 s + d_o$$

$$s^3 + 48.396 s^2 + 197.14 s + 875 = d_3 s^3 + d_2 s^2 + d_1 s + d_o$$

(special PID case, $b_1 = 0$)

$$k_i = \frac{d_o}{b_o} = \frac{875}{1}$$

$$k_i = 875$$

$$k_p = \frac{d_1 - a_o}{b_o}$$

$$= \frac{197.14 - 0}{1}$$

$$K_p = 197.14$$

$$k_d = \frac{d_2 - a_1}{b_o}$$

$$= \frac{48.396 - 0.4}{1}$$

$$k_d = 47.996$$

$$D(s) = \frac{k_d s^2 + k_p \cdot s + k_i}{s} \quad \checkmark$$

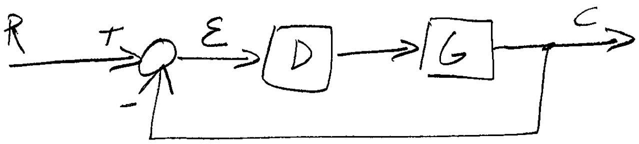
$$\boxed{D(s) = \frac{47.996 s^2 + 197.14 \cdot s + 875}{s}}$$

11/24/13

## steady state response, step

PID

$$G(s) = \frac{1}{s^2+0.4s} \qquad D(s) = \frac{48s^2+197s+875}{s}$$




$$H = \frac{C}{R} = \frac{DG}{1+DG} \qquad \mathcal{E} = \frac{1}{1+DG} = 1-H$$

$$\mathcal{E}_{ss\,step} = \lim_{s\to 0} s \cdot R \cdot \frac{1}{1+DG} \qquad \frac{1}{s} \quad \text{(step)}$$

$$= \lim_{s\to 0} \cancel{s} \cdot \frac{1}{\cancel{s}} \cdot \frac{1}{1+\dfrac{48s^2+197s+875}{s(s^2+0.4s)}}$$

$$= \lim_{s\to 0} \frac{s(s^2+0.4s)}{s(s^2+0.4s)+48s^2+197s+875}$$

$$\boxed{\mathcal{E}_{ss\,step} = 0 \qquad \text{agrees with plot}}$$

The step response of this system is show in Figure 7. The Matlab code used to produce this plot is shown in Listing 5.

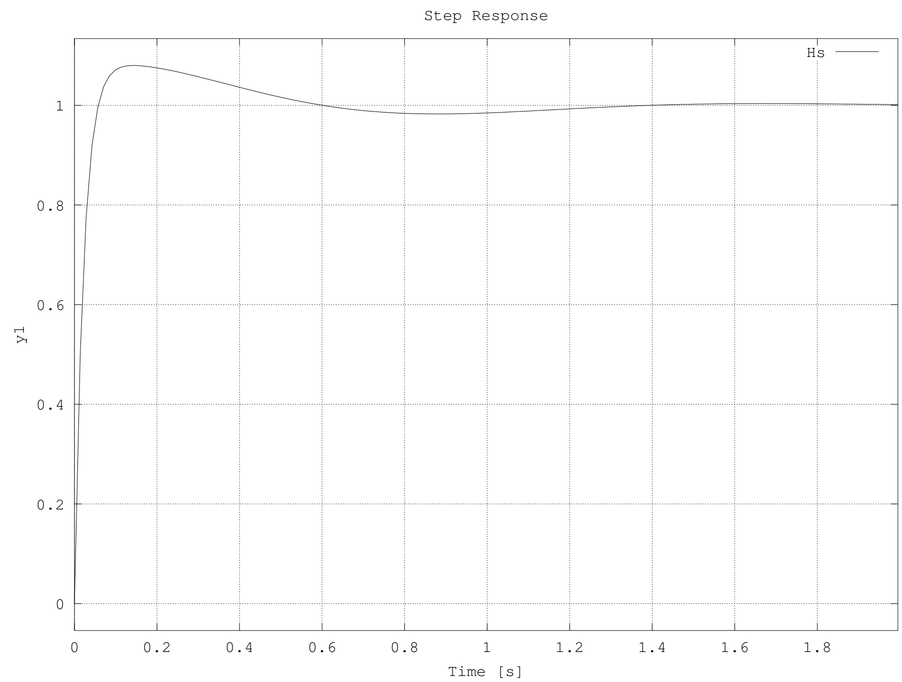



Figure 7: Step response of PID controller.

```
%
% pids2s4_plot.m
%
% PID, step response
%

Tf = 2;  % time final (sec)

Gs = tf([1], [1 0.4 0]);
Ds = tf([47.996 197.14 875], [1 0]);

Hs = (Ds*Gs)/(1 + Ds*Gs);

step(Hs, Tf);

print('pids2s4_plot.eps', '-deps');
```

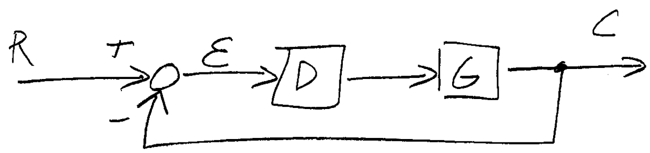
Listing 5: Matlab script to plot response of PID controller.

11/24/13

steady state response, ramp

PID

$$G(s) = \frac{1}{s^2 + 0.4 s} \qquad D(s) = \frac{48 s^2 + 197 s + 875}{s}$$




$$\frac{C}{R} = H = \frac{DG}{1 + DG} \qquad E = \frac{1}{1 + DG}$$

$$\frac{1}{s^2} \quad \text{(ramp)}$$

$$E_{ss,\,\text{ramp}} = \lim_{s \to 0} s \cdot R \cdot \frac{1}{1 + DG}$$

$$= \lim_{s \to 0} s \cdot \frac{1}{s^2} \cdot \frac{1}{1 + \dfrac{48 s^2 + 197 s + 875}{s(s^2 + 0.4 s)}}$$

$$= \lim_{s \to 0} \frac{1}{\cancel{s}} \cdot \frac{\cancel{s}(s^2 + 0.4 s)}{s(s^2 + 0.4 s) + 48 s^2 + 197 s + 875}$$

$$= \lim_{s \to 0} \frac{s^2 + 0.4 s}{s^3 + 48.4 s^2 + 197 s + 875}$$

$= \boxed{0}$

But from the plot of the ramp response it is clearly not zero!

A shortcoming of the final value theorem?

The ramp response of this system is show in Figure 8. The Matlab code used to produce this plot is shown in Listing 6. Notice that the response does not agree with the calculated error.

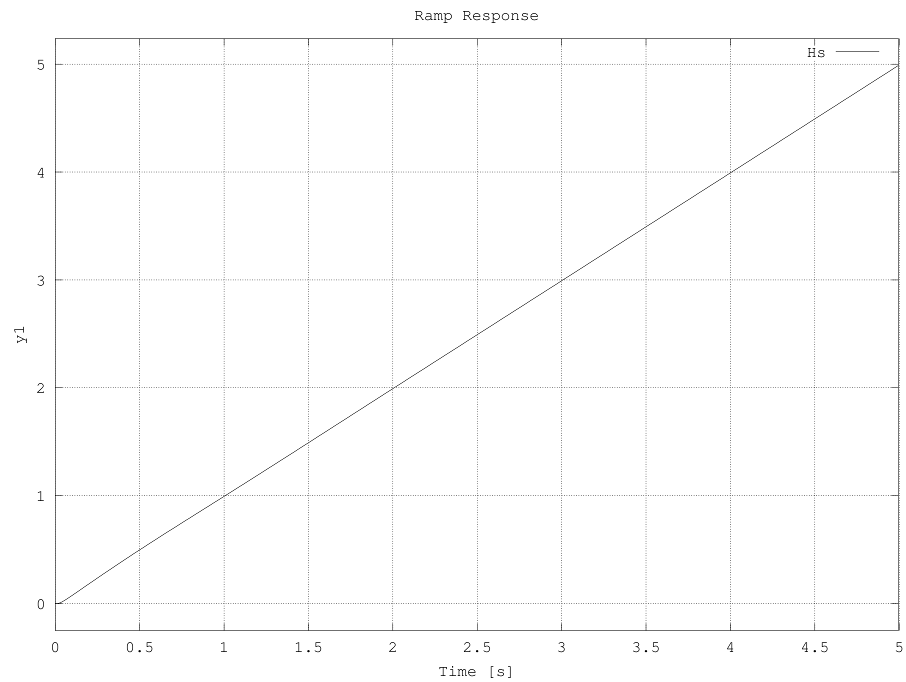



Figure 8: Ramp response of PID controller.

```
%
% pids2s4_plot.m
%
% PID, ramp response
%

Tf = 5;  % time final (sec)

Gs = tf([1], [1 0.4 0]);
Ds = tf([47.996 197.14 875], [1 0]);

Hs = (Ds*Gs)/(1 + Ds*Gs);

ramp(Hs, Tf);

print('pids2s4r_plot.eps', '-deps');
```

Listing 6: Matlab script to plot ramp response of PID controller.

## 10 State Space to Transfer Function (S-Domain)

**Example 1**

State Space → Transfer function

(e.g.)

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

$$\underset{2\times1}{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}} = \underset{2\times2}{\begin{bmatrix} -10 & 1 \\ -0.02 & -2 \end{bmatrix}} \underset{2\times1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} + \underset{2\times1}{\begin{bmatrix} 0 \\ 2 \end{bmatrix}} \cdot \underset{1\times1}{u}$$

$$y = \underbrace{\underset{1\times2}{\begin{bmatrix} 1 & 0 \end{bmatrix}} \underset{2\times1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}}_{1\times1} + \cancel{0 \cdot u}$$

Recall Laplace

$$\mathcal{L}\{\dot{x}\} = s \cdot \mathcal{L}\{x\} - x(0)$$

$$\mathcal{L}\{x(t)\} = X(s)$$

$$s \cdot X(s) = A \cdot X(s) + B u(s)$$

$$s \cdot X - AX = Bu$$

$$s \cdot I \cdot X - Ax = Bu$$

$$(sI - A)X = Bu$$

$$\boxed{X = (sI - A)^{-1} \cdot Bu}$$

(cont)

12/12/12

(cont)

$$u \longrightarrow \boxed{G(s)} \longrightarrow y$$

$$y = C \cdot X$$

$$y = C \cdot \left[ (sI - A)^{-1} \cdot Bu \right]$$

$$\frac{y}{u} = C \cdot (sI - A)^{-1} \cdot B$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \left( \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -10 & 1 \\ -0.02 & -2 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \left( \begin{bmatrix} s+10 & -1 \\ 0.02 & s+2 \end{bmatrix}^{-1} \right) \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$\left. \begin{array}{l} \det = (s+10)(s+2) + 0.02 \\ \dfrac{1}{\det} \begin{bmatrix} s+2 & 1 \\ -0.02 & s+10 \end{bmatrix} \end{array} \right\} \text{inverse}$$

$$= \frac{\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s+2 & 1 \\ -0.02 & s+10 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix}}{s^2 + 12s + 20.02}$$

$$= \frac{\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 2s+20 \end{bmatrix}}{s^2 + 12s + 20.02}$$

$$\boxed{G(s) = \frac{2}{s^2 + 12s + 20.02}}$$

Matlab

ss2tf(A,B,C,D);

= [num, den]

* solution is unique

12/12/12

## 11 Transfer Function (S-Domain) to State Space

**Example 1**

Transfer Function to State Space

(e.g.) $$G(s) = \frac{10}{s^3+6s^2+11s+6}$$

$$= \frac{10}{(s+3)(s+2)(s+1)}$$

$$\frac{10}{(s+3)(s+2)(s+1)} = \frac{A}{s+3} + \frac{B}{s+2} + \frac{C}{s+1}$$

$$A = \frac{10}{(-1)(-2)} \qquad B = \frac{10}{-1}$$

$$A = 5 \qquad B = -10$$

$$C = \frac{10}{2}$$

$$C = 5$$

$$G(s) = \frac{5}{s+3} - \frac{10}{s+2} + \frac{5}{s+1}$$

$$\frac{y}{u} = \frac{5}{s+3} - \frac{10}{s+2} + \frac{5}{s+1}$$

$$y = \frac{5}{s+3}\cdot u - \frac{10}{s+2}\cdot u + \frac{5}{s+1}\cdot u$$

(cont)

12/12/12

(cont)

substitute with $x$

setup as passthru

$$y = \underbrace{\frac{5}{s+3}\cdot u}_{x_1} \underbrace{- \frac{10}{s+2}\cdot u}_{x_2} + \underbrace{\frac{5}{s+1}\cdot u}_{x_3}$$

$$x_1 = \frac{5}{s+3}\cdot u \qquad x_2 = -\frac{10}{s+2}\cdot u \qquad x_3 = \frac{5}{s+1}\cdot u$$

$$x_1(s+3) = 5\cdot u \qquad x_2(s+2) = -10u \qquad x_3(s+1) = 5\cdot u$$

$$s\cdot x_1 = 5\cdot u - 3x_1 \qquad s\cdot x_2 = -10u - 2\cdot x_2 \qquad s\cdot x_3 = 5\cdot u - x_3$$

$$\left( \mathcal{L}\{\dot{x}\} = s\cdot X(s) - x(0) \right)$$ assume $x(0)=0$

$$\dot{x}_1 = -3\cdot x_1 + 5u \qquad \dot{x}_2 = -2x_2 - 10u \qquad \dot{x}_3 = -x_3 + 5u$$

$$\dot{x} = Ax + Bu$$

$$y = Cx + \underbrace{Du}_{0}$$

$$\dot{x} = \begin{bmatrix} -3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 5 \\ -10 \\ 5 \end{bmatrix} u$$

Solution is not unique, assumptions were made

$$y = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

pass thru

12/12/12

## 12 Z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{2}$$

**Example 1**

<u>z - Transform</u>

$$X(z) = Z\{x[k]\} = \sum_{n=-\infty}^{\infty} x[k]\, z^{-k}$$

e.g.) <u>delta</u>

$$Z\{\delta[k]\} = \sum_{k=-\infty}^{\infty} \delta[k] \cdot z^{-k}$$

$$= \sum_{k=0}^{0} \delta[0] \cdot z^{-0}$$

$$= \boxed{1}$$

e.g.) <u>unit step</u>

$$Z\{u[k]\} = \sum_{k=-\infty}^{\infty} u[k] \cdot z^{-k}$$

$$= \sum_{k=0}^{\infty} z^{-k} = 1 + z^{-1} + z^{-2} + z^{-3} + \ldots$$

(Geometric series)

$$= \frac{1}{1 - z^{-1}} = \boxed{\frac{z}{z - 1}}$$

10/22/12

## 13 Geometric Series

Geometric Series

$$S_n = \sum_{k=0}^{n} r^k = 1 + r + r^2 + r^3 + \dots r^n$$

$$r \cdot S_n = r + r^2 + r^3 + \dots + r^{n+1}$$

$$r \cdot S_n - S_n = -1 + r^{n+1}$$

$$S_n (r-1) = -1 + r^{n+1}$$

$$S_n = \frac{-1 + r^{n+1}}{r-1}$$

$$\boxed{S_n = \frac{1 - r^{n+1}}{1-r} = \sum_{k=0}^{n} r^k}$$

$$S_\infty = \sum_{k=0}^{\infty} r^k = \frac{1 - r^{\infty+1}}{1-r}$$

$$\boxed{\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}}$$

(infinite geometric series)

$$\boxed{\sum_{k=0}^{\infty} r^{-k} = \frac{1}{1-r^{-1}} = \frac{r}{r-1}}$$

# 14 Algorithm (Z-Domain) to Difference Equation

**Example 1**

Algorithm (z-domain) to Difference Equation

(e.g.)

$$\frac{z^2 - 0.95z + 0.7}{z^2 - 0.85z + 0.98}$$

$$\frac{u(z)}{\mathcal{E}(z)} = \frac{1 - 0.95 \cdot z^{-1} + 0.7z^{-2}}{1 - 0.85z^{-1} + 0.98z^{-2}}$$

$$u(z) - u(z)0.85z^{-1} + u(z)0.98z^{-2}$$
$$= \mathcal{E}(z) - \mathcal{E}(z)0.95z^{-1} + \mathcal{E}(z)0.7z^{-2}$$

$$u(z) = u(z)0.85z^{-1} - u(z)0.98z^{-2} + \mathcal{E}(z) - \mathcal{E}(z)0.95z^{-1}$$
$$+ \mathcal{E}(z)0.7z^{-2}$$

$$\boxed{u(k) = u(k-1)0.85 - u(k-2)0.98 + \mathcal{E}(k) - \mathcal{E}(k-1)0.95 + \mathcal{E}(k-2)0.7}$$

10/22/12

**Example 2**

Algorithm ($z$-domain) to Difference Equation

(e.g.) $$D(z) = 10 - 9z^{-1}$$

$$\frac{u(z)}{\mathcal{E}(z)} = 10 - 9 \cdot z^{-1}$$

$$u(z) = 10 \cdot \mathcal{E}(z) - 9 \cdot \mathcal{E}(z)\, z^{-1}$$

$$u(k) = 10 \cdot \mathcal{E}(k) - 9 \cdot \mathcal{E}(k-1)$$ (inverse $z$-transform)

10/23/12

**Example 3**

(e.g.)

$$D(z) = \frac{11 - 10z^{-1}}{2 - z^{-1}}$$

$$\frac{u(z)}{\mathcal{E}(z)} = \frac{11 - 10z^{-1}}{2 - z^{-1}}$$

$$u(z)\left[2 - z^{-1}\right] = \mathcal{E}(z)\left[11 - 10z^{-1}\right]$$

$$2 \cdot u(z) - u(z)z^{-1} = 11 \cdot \mathcal{E}(z) - 10 \cdot \mathcal{E}(z)z^{-1}$$

$$2 \cdot u(z) = u(z)z^{-1} + 11 \cdot \mathcal{E}(z) - 10 \cdot \mathcal{E}(z)z^{-1}$$

$$u(z) = \frac{1}{2}\left[u(z)z^{-1} + 11\,\mathcal{E}(z) - 10\,\mathcal{E}(z)z^{-1}\right]$$

$$\boxed{u(k) = \frac{1}{2}\left[u(k-1) + 11 \cdot \mathcal{E}(k) - 10 \cdot \mathcal{E}(k-1)\right]}$$

10/23/12

# 15 Zero Order Hold

A Zero Order Hold converts a S-domain system to the Z-domain. It is effectively the same as putting an D/A converter before the continuous system. In fact this is exactly what is done in Simulink (Figure 9).

The Zero Order Hold is typically used for $G(s)$, the plant/model. Mapping operations such as Forward, Backward, etc (Section ) are not typically used.

$$G(z) = (1 - z^{-1})Z\left[\frac{G(s)}{s}\right] \tag{3}$$

All of these examples can be verified in Matlab using code as shown below. There may be small differences due to round off errors.

```
% Matlab
Bs = [1];
As = [1 0];
Gs = tf(Bs, As);
Gz = c2d(Gs, T, 'ZOH');
```

**Example 1**

s-domain $\Rightarrow$ z-domain

Zero Order Hold

(e.g.) $$G(s) = \frac{1}{s^2}$$

$$G(z) = (1 - z^{-1})\, \mathcal{Z}\left[\frac{G(s)}{s}\right]$$

$$= (1 - z^{-1})\, \mathcal{Z}\left[\frac{1}{s^3}\right]$$

| $X(s)$ | $X(z)$ |
|---|---|
| $\frac{2}{s^3}$ | $\frac{T^2 z^{-1}(1+z^{-1})}{(1-z^{-1})^3}$ |

$$= (1 - z^{-1}) \cdot \frac{1}{2} \cdot \mathcal{Z}\left[\frac{2}{s^3}\right]$$

$$= \cancel{(1 - z^{-1})}\, \frac{1}{2} \cdot \frac{T^2 \cdot z^{-1}(1+z^{-1})}{(1-z^{-1})^{\cancel{3}\,2}}$$

$$G(z) = \frac{T^2 \cdot z^{-1}(1+z^{-1})}{2(1-z^{-1})^2}$$

$$= \frac{T^2}{2} \cdot \frac{(z^{-1}+z^{-2})}{(1-z^{-1})^2}$$

$$\boxed{G(z) = \frac{T^2}{2} \cdot \frac{z^{-1}+z^{-2}}{1-2z^{-1}+z^{-2}}}$$

**Example 2**

$G(s) \to G(z)$ discrete

$$G(s) = \frac{1}{s(s+0.4)}$$

$$G(z) = (1-z^{-1})\, \mathcal{Z}\left[\frac{G(s)}{s}\right] \qquad (ZOH)$$

$$= (1-z^{-1})\, \mathcal{Z}\left[\frac{1}{s^2(s+0.4)}\right]$$

$$\frac{1}{s^2(s+0.4)} = \frac{A}{s} + \frac{B}{s^2} + \frac{C}{s+0.4} \qquad \left(\text{partial fractions}\right)$$

...

$$= \frac{-6.25}{s} + \frac{2.5}{s^2} + \frac{6.25}{s+0.4}$$

$$= \frac{-6.25}{1-z^{-1}} + \frac{2.5 \cdot T z^{-1}}{(1-z^{-1})^2} + \frac{6.25}{1-e^{-0.4 \cdot T} \cdot z^{-1}}$$

$$G(z) = (1-z^{-1})\left[\frac{-6.25}{1-z^{-1}} + \frac{2.5\, T \cdot z^{-1}}{(1-z^{-1})^2} + \frac{6.25}{1-e^{-0.4 \cdot T} z^{-1}}\right]$$

$$= -6.25 + \frac{2.5 \cdot T \cdot z^{-1}}{(1-z^{-1})} + \frac{6.25(1-z^{-1})}{1-e^{-0.4 \cdot T} \cdot z^{-1}}$$

(cont)

(cont)

$T = 0.1$

$$G(z) = -6.25 + \frac{0.25 \cdot z^{-1}}{1 - z^{-1}} + \frac{6.25\,(1 - z^{-1})}{1 - 0.961 \cdot z^{-1}}$$

$$= \frac{-6.25(1-z^{-1})(1-0.961z^{-1}) + 0.25z^{-1}(1-0.961z^{-1}) + 6.25(1-z^{-1})^2}{(1-z^{-1})(1-0.961z^{-1})}$$

$$= \frac{-6.25\left[\cancel{1} - 1.961z^{-1} + 0.961z^{-2}\right] + 0.25z^{-1} - 0.240z^{-2} + 6.25\left[\cancel{1} - 2z^{-1} + z^{-2}\right.}{(1-z^{-1})(1-0.961z^{-1})}$$

$$\boxed{G(z) = \frac{0.006\,z^{-1} + 0.004\,z^{-2}}{1 - 1.961\,z^{-1} + 0.961\,z^{-2}}}\ \checkmark$$

**Example 3**

11/22/13

$$\underline{G(s) \rightarrow G(z)}$$

(e.g.) $G(s) = \dfrac{0.1}{s(s+0.1)}$ $T = 1$

$$G(z) = (1 - z^{-1})\, Z\left[\frac{G(s)}{s}\right] \quad \text{(ZOH)}$$

$$= (1 - z^{-1})\, Z\left[\frac{0.1}{s^2(s+0.1)}\right]$$

$$\frac{0.1}{s^2(s+0.1)} = \frac{A}{s} + \frac{B}{s^2} + \frac{C}{s+0.1} \quad \text{(partial fractions)}$$

$$C = \frac{0.1}{(-0.1)^2}$$

$$C = 10$$

Let $s = 1$

$$\frac{0.1}{1.1} = A + B + \frac{10}{1.1}$$

$$A + B = -\frac{9.9}{1.1}$$

$$A + B = -9$$

$$A = -9 - B$$

Let $s = 2$

$$\frac{0.1}{4(2.1)} = \frac{A}{2} + \frac{B}{4} + \frac{10}{2.1}$$

$$\frac{A}{2} + \frac{B}{4} = -4.75$$

$$2A + B = -19$$

$$2(-9-B) + B = -19$$

$$-18 - 2B + B = -19$$

$$-B = -1$$

$$B = 1$$

$$A + B = -9$$

$$A = -10$$

(cont)

9/22/13

(cont)

$$G(z) = (1-z^{-1})\, \mathcal{Z}\left[ -\frac{10}{s} + \frac{1}{s^2} + \frac{10}{s+0.1} \right]$$

$$= (1-z^{-1})\left[ \frac{-10}{1-z^{-1}} + \frac{T\cdot z^{-1}}{(1-z^{-1})^2} + \frac{10}{1-e^{-0.1T}z^{-1}} \right]$$

$$= -10 + \frac{T\cdot z^{-1}}{(1-z^{-1})} + \frac{10(1-z^{-1})}{1-e^{-0.1T}z^{-1}}$$

Let $T=1$

$$= -10 + \frac{z^{-1}}{1-z^{-1}} + \frac{10(1-z^{-1})}{1-0.905\cdot z^{-1}}$$

$$-10(1-z^{-1})(1-0.905z^{-1}) + z^{-1}(1-0.905z^{-1}) + 10(1-z^{-1})(1-z^{-1})$$

$$(-10+10z^{-1})(1-0.905z^{-1}) + z^{-1} - 0.905z^{-2} + (10-10z^{-1})(1-z^{-1})$$

$$\cancel{-10} + 19.05z^{-1} - 9.05z^{-2} + z^{-1} - 0.905z^{-2} + \cancel{10} - 20z^{-1} + 10z^{-2}$$

$$= \frac{0.05z^{-1} + 0.045z^{-2}}{(1-z^{-1})(1-0.905z^{-1})}$$

$$\boxed{G(z) = \frac{0.05z^{-1} + 0.045z^{-2}}{1 - 1.905z^{-1} + 0.905z^{-2}}}$$ ✓

**Example 4**

11/22/13

$$\underline{G(s) \to G(z) \quad (ZOH)}$$

(e.g.) $G(s) = \dfrac{1}{s(s+1)}$ $\qquad T = 1$

$$G(z) = (1 - z^{-1})\, \mathcal{Z}\left[\frac{G(s)}{s}\right]$$
$$= (1 - z^{-1})\, \mathcal{Z}\left[\frac{1}{s^2(s+1)}\right]$$

$$\frac{1}{s^2(s+1)} = \frac{A}{s} + \frac{B}{s^2} + \frac{C}{s+1}$$

$C = 1$

Let $s = 1$

$$\frac{1}{2} = A + B + \frac{C}{2}$$

$$A + B = 0$$

Let $s = 2$

$$\frac{1}{s+1} = A \cdot s + B + \frac{C \cdot s^2}{s+1}$$

$$\frac{1}{3} = 2A + B + \frac{4}{3}$$

$$2A + B = -1$$

$$A + B = 0$$

$A = -1$

$B = 1$

$$G(z) = (1 - z^{-1})\, \mathcal{Z}\left[\frac{-1}{s} + \frac{1}{s^2} + \frac{1}{s+1}\right]$$

(cont)

(cont)

$$G(z) = (1 - z^{-1})\left[\frac{-1}{1-z^{-1}} + \frac{T \cdot z^{-1}}{(1-z^{-1})^2} + \frac{1}{1-e^{-T}z^{-1}}\right]$$

$(T = 1)$

$$= (1-z^{-1})\left[\frac{-1}{1-z^{-1}} + \frac{z^{-1}}{(1-z^{-1})^2} + \frac{1}{1-e^{-1}z^{-1}}\right]$$

$$= -1 + \frac{z^{-1}}{(1-z^{-1})} + \frac{1-z^{-1}}{1-e^{-1}z^{-1}}$$

$$= \frac{-(1-z^{-1})(1-e^{-1}z^{-1}) + z^{-1}(1-e^{-1}z^{-1}) + (1-z^{-1})^2}{(1-z^{-1})(1-e^{-1}z^{-1})}$$

$$= \frac{\cancel{-1} + z^{-1}(1-e^{-1}) - e^{-1}z^{-2} + z^{-1} - e^{-1}z^{-2} \cancel{+1} - 2z^{-1} + z^{-2}}{1 - z^{-1} - e^{-1}z^{-1} + e^{-1}z^{-2}}$$

$$= \frac{z^{-1}e^{-1} + z^{-2}(1-2e^{-1})}{1 - z^{-1}(1+e^{-1}) + z^{-2} \cdot e^{-1}}$$

$$\boxed{G(z) = \frac{0.368\,z^{-1} + 0.264\,z^{-2}}{1 - 1.368\,z^{-1} + 0.368\,z^{-2}}}$$

**Example 5**

11/25/13

$$G(s) \to G(z) \quad ZOH$$

General T

$$G(s) = \frac{1}{s(s+1)}$$

$$G(z) = (1 - z^{-1})\, \mathcal{Z}\left[\frac{G(s)}{s}\right]$$

$$= (1 - z^{-1})\, \mathcal{Z}\left[\frac{1}{s^2(s+1)}\right] \qquad (ZOH)$$

$$\frac{1}{s^2(s+1)} = \frac{A}{s} + \frac{B}{s^2} + \frac{C}{s+1}$$

Let $s=-1$ $\quad$ Let $s=0$

$C = 1$ $\quad$ $B = 1$

Let $s=1$

$$\frac{1}{2} = A + B + \frac{C}{2}$$

$$0 = A + 1$$

$$A = -1$$

$$G(z) = (1 - z^{-1})\, \mathcal{Z}\left[\frac{-1}{s} + \frac{1}{s^2} + \frac{1}{s+1}\right]$$

(cont)

(cont)

$$G(z) = (1-z^{-1})\left[\frac{-1}{1-z^{-1}} + \frac{T z^{-1}}{(1-z^{-1})^2} + \frac{1}{1-e^{-T}z^{-1}}\right]$$

$$= -1 + \frac{T z^{-1}}{(1-z^{-1})} + \frac{1-z^{-1}}{1-e^{-T}z^{-1}}$$

Let

$$\gamma = 1 - z^{-1}$$

$$\tau = 1 - e^{-T} z^{-1}$$

$$= \frac{-\gamma\cdot\tau + T\cdot z^{-1}\tau + \gamma^2}{\gamma\,\tau}$$

(numerator)

$z^{-2}$: $1 - T\cdot e^{-T} - e^{-T}$

$1 - e^{-T}(T+1)$

$z^{-1}$: $-2 + T + 1 + e^{-T}$

$-1 + T + e^{-T}$

$z^{0}$: $1 - 1 = 0$

$$\gamma^2 = (1-z^{-1})(1-z^{-1})$$
$$= 1 - 2z^{-1} + z^{-2}$$

$$T\cdot z^{-1}\cdot\tau = T(z^{-1} - e^{-T}z^{-2})$$
$$= T\cdot z^{-1} - T\cdot e^{-T} z^{-2}$$

$$-\gamma\cdot\tau = (z^{-1} - 1)(1 - e^{-T}z^{-1})$$
$$= z^{-1} - 1 - e^{-T}z^{-2} + e^{-T}z^{-1}$$
$$= -1 + z^{-1}(1+e^{-T}) - e^{-T}z^{-2}$$

$$\boxed{\begin{aligned} G(z) &= \frac{(T+e^{-T}-1)z^{-1} + [1-e^{-T}(T+1)]z^{-2}}{(1-z^{-1})(1-e^{-T}z^{-1})} \\ &= \frac{(T+e^{-T}-1)z^{-1} + [1-e^{-T}(T+1)]z^{-2}}{1-(1+e^{-T})z^{-1} + e^{-T}z^{-2}} \end{aligned}}$$

substitute for T to find a specific version

# 16 Mapping (S-Domain to Z-Domain)

When designing by discrete equivalents the design is performed in the continuous domain and then converted to the discrete domain. This is in contrast to Direct Design (Section 20) where the designed is performed in the discrete domain.

This method is straightforward however it doesn't always result in a stable system.

## 16.1 Mapping: $z = e^{sT}$

**Example 1**

$s \Rightarrow z$ mapping

$z = e^{sT}$

$T = 0,1$

$$D(s) = \frac{177,94 \cdot s + 875}{s + 47,996}$$

$$s_{z1} = \frac{-875}{177,94}$$

$$s_{z1} = -4,917$$

$$s_{p1} = -47,996$$

$$z_{z1} = e^{-4,917(0,1)}$$

$$z_{z1} = 0,6115$$

$$z_{p1} = e^{-47,996(0,1)}$$

$$z_{p1} = 0,00823$$

$$D(z) = k \cdot \frac{(z - 0,6115)}{(z - 0,00823)}$$

$$\lim_{s \to 0} D(s) = \lim_{s \to 0} \frac{177,94 \cdot s + 875}{s + 47,996} = 18,231$$

$$\lim_{z \to 1} D(z) = k \cdot \frac{(z - 0,6115)}{(z - 0,00823)} = 18,231$$

$$k = 46,54$$

$$\boxed{D(z) = (46,54) \frac{(z - 0,6115)}{(z - 0,00823)}}$$

**Example 2**

$s \to z$ mapping $\left(z = e^{sT}\right)$ 10/23/12

$$D(s) = \frac{10(s+0.1)}{s+1} \qquad T = 1$$

$$D(s) = \frac{10s+1}{s+1}$$

zeros: $-\frac{1}{10}$

poles: $-1$

$$z_z = e^{\left(-\frac{1}{10}\right)T}$$

$$z_p = e^{-T}$$

$$D(z) = K \cdot \frac{z - e^{\left(-\frac{1}{10}\right)T}}{z - e^{-T}}$$

$$= K \cdot \frac{z - 0.904}{z - 0.368}$$

(to find $K$, use limits)

$$\lim_{s \to 0} D(s) = \lim_{s \to 0} \frac{10 \cdot s + 1}{s+1} = 1$$

$$\lim_{z \to 1} D(z) = K \cdot \frac{z - 0.904}{z - 0.368} = 1$$

$$k = \frac{1 - 0.368}{1 - 0.904} = 6.58$$

$$\boxed{D(z) = (6.58)\,\frac{z - 0.904}{z - 0.368}}$$

## 16.2 Mapping: Forward, Backward, Trapezoid

**Example 1**

Mapping (s-domain ⇒ z-domain)

Forward

e.g.) $$D(s) = \frac{10\,(s+0.1)}{s+1} \qquad T=1$$

$$s = \frac{z-1}{T}$$

$$D(z) = \frac{10\left(\frac{z-1}{T}+0.1\right)}{\frac{z-1}{T}+1}$$

$$= \frac{10\,(z-1+0.1\cdot T)}{z-1+T}$$

$$= \frac{10\cdot z-10+T}{z-1+T}$$

$(T=1)$

$$= \frac{10\cdot z-9}{z}$$

$$\boxed{D(z) = 10-9z^{-1}}$$

10/23/12

**Example 2**

Mapping (S-domain to Z-domain)
Backward

(e.g.) $$D(s) = \frac{10(s+0.1)}{s+1} \qquad s = \frac{z-1}{z \cdot T}$$

$$D(z) = \frac{10\left(\frac{z-1}{z \cdot T} + 0.1\right)}{\frac{z-1}{z \cdot T} + 1}$$

$$= \frac{10(z-1+0.1 \cdot zT)}{z-1+zT}$$

$$= \frac{10z - 10 + zT}{z(1+T) - 1}$$

$$= \frac{z(10+T) - 10}{z(1+T) - 1}$$

Let $T = 1$

$$D(z) = \frac{11 \cdot z - 10}{2z - 1}$$

$$\boxed{D(z) = \frac{11 - 10z^{-1}}{2 - z^{-1}}}$$

10/23/12

**Example 3**

$s \to z$, backward

$T = 0.1$

(from pole placement)

$$D(s) = \frac{177.94 \cdot s + 875}{s + 47.996}$$

$$s = \frac{z-1}{z \cdot T}$$

backward

$$D(z) = \frac{177.94 \frac{z-1}{z \cdot T} + 875}{\frac{z-1}{zT} + 47.996}$$

$$= \frac{177.94(z-1) + 875 \cdot z \cdot T}{z - 1 + 47.996 \cdot zT}$$

$$= \frac{(177.94 + 875 \cdot T)\, z - 177.94}{(1 + 47.996 \cdot T)\, z - 1}$$

$T = 0.1$

$$\boxed{D(z) = \frac{265.4 \cdot z - 177.94}{5.799 \cdot z - 1}}$$

**Example 4**

$$\underline{s \to z, \text{ trapezoid}}$$

$T = 0.1$

$$s = \frac{2}{T}\frac{z-1}{z+1}$$

trapezoid

$$D(s) = \frac{177.94 \cdot s + 875}{s + 47.996}$$

$$D(z) = \frac{177.94 \frac{2(z-1)}{T(z+1)} + 875}{\frac{2(z-1)}{T(z+1)} + 47.996}$$

$$= \frac{355.88(z-1) + 875 \cdot T \cdot (z+1)}{2(z-1) + 47.996 \cdot T(z+1)}$$

$$= \frac{(355.88 + 875 \cdot T) z + (875 \cdot T - 355.88)}{(2 + 47.996 \cdot T) z - 2 + 47.996 \cdot T}$$

$$\boxed{D(z) = \frac{443.3 \cdot z - 268.38}{6.799 \cdot z + 2.799}}$$

✓

### 16.3 Digital PID, Ghandakly's Method

# 17 Direct Design Method of Ragazzini

One method for finding $D(z)$, if $H(z)$ and $G(z)$ are given, is to simply solve for $D(z)$.[4]

$$\begin{aligned} H(z) &= \frac{D(z)G(z)}{1 + D(z)G(z)} \\ D(z) &= \frac{1}{G(z)} \frac{H(z)}{1 - H(z)} \end{aligned} \tag{4}$$

# 18 $K_v$ Direct Design Method

This method is characterized by placing a limit on the error produced from a ramp input ($K_v$). The error due to a step input ($K_p$) will also be zero.

Alternative names for this method are the Direct Method of Ragazzini[5] and the Analytical Design Method.[6]

# 19 Diophantine Equation

The Diophantine Equation is used to find a solution to a system if it is in a very specific form (Equation 5). For more information refer to Ogata[7] where this method is called the "Polynomial Equations Approach".

$$\alpha(z)A(z) + \beta(z)B(z) = D \tag{5}$$

Where $D$ is the characteristic polynomial. Typically, $A(z)$ and $B(z)$ are known and $\alpha(z)$ and $\beta(z)$ are to be found. Each element must have a specific order as shown below. The order ($n$) will correspond to the order of the Sylvester Matrix (Section 19.1).

$$\begin{array}{ccccccc} \alpha(z) & A(z) & + & \beta(z) & B(z) & = & D \\ (n-1) & (n) & & (n-1) & (n) & & (2n-1) \end{array}$$

When $A(z)$ and $B(z)$ are known $\alpha(z)$ and $\beta(z)$ can be found using Equation 6.

$$\begin{aligned} M &= E^{-1}D \\ \begin{bmatrix} \alpha \\ \beta \end{bmatrix} &= E^{-1}D \end{aligned} \tag{6}$$

[4] Franklin, Powell, and Workman, see n. ??, Pg. 265.
[5] Ibid., Pg. 264.
[6] Ogata, see n. ??, Pg. 242.
[7] Ibid., Pg. 525.

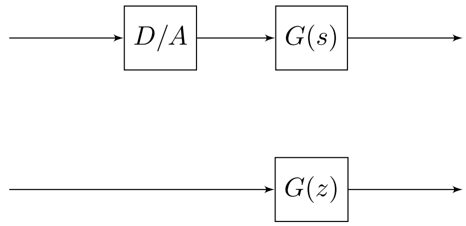



Figure 9: A Zero Order Hold in Simulink created by preceding a $G(s)$ system with a D/A converter to produce a $G(z)$ system.

## 19.1 Sylvester Matrix

Second Order

$$E = \begin{bmatrix} a_2 & 0 & b_2 & 0 \\ a_1 & a_2 & b_1 & b_2 \\ a_0 & a_1 & b_0 & b_1 \\ 0 & a_0 & 0 & b_0 \end{bmatrix}$$

Third Order

$$E = \begin{bmatrix} a_3 & 0 & 0 & b_3 & 0 & 0 \\ a_2 & a_3 & 0 & b_2 & b_3 & 0 \\ a_1 & a_2 & a_3 & b_1 & b_2 & b_3 \\ a_0 & a_1 & a_2 & b_0 & b_1 & b_2 \\ 0 & a_0 & a_1 & 0 & b_0 & b_1 \\ 0 & 0 & a_0 & 0 & 0 & b_0 \end{bmatrix}$$

# 20 Direct Design ($K$), Diophantine

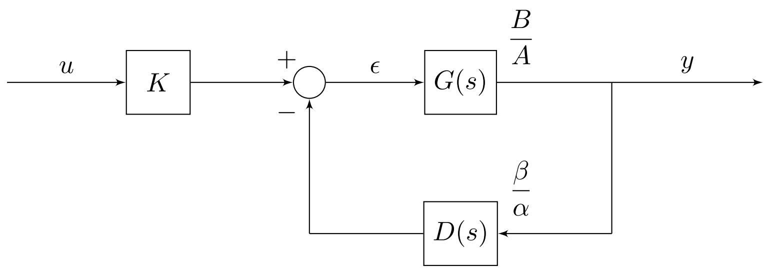



Figure 10: Direct Design system with $K$ as a scaling input, $G$ is the plant and $D$ ($\beta/\alpha$) is the controller.

The general structure for a Direct Design system is shown in Figure 10. This is different than previous designs in that the controller ($D$) is in the feedback loop and there is a gain ($K$) on the input.

A controller is found by first converting the plant ($G(s)$) in to discrete form using a Zero Order Hold (Section 15). Then the poles are placed according to roots given by the designer. Roots may need to be added to correct the order so that the Diophantine Equation can be used to find $\alpha$ and $\beta$. Examples of this process are included below. For a detailed examination refer to Ogata.[8]

$$
\begin{aligned}
H &= \frac{y}{u} \\
H &= K\frac{G}{1+DG} \qquad (7) \\
&= K\frac{B\alpha}{A\alpha + B\beta} \qquad (8)
\end{aligned}
$$

[8]Ibid., Pg. 517.

**Example 1**

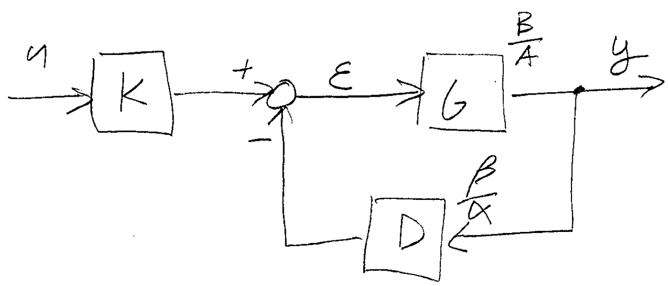
## Direct Design

(e.g.) $G(s) = \frac{1}{s^2}$

roots: $z = 0.6 \pm 0.4i$ (designer specified)

$$G(z) = (1 - z^{-1})\, \mathcal{Z}\left[\frac{G(s)}{s}\right] \qquad \text{(ZOH)}$$

$$G(z) = \frac{T^2}{2} \cdot \frac{z^{-1} + z^{-2}}{1 - 2z^{-1} + z^{-2}} \qquad \text{(refer to ZOH section)}$$




$$\frac{y}{u} = H = K \cdot \frac{G}{1 + DG}$$

Let $G = \frac{B}{A}$

$D = \frac{\beta}{\alpha}$

$$H = K \cdot \frac{B \cdot \alpha}{A \cdot \alpha + B\beta}$$

(cont)

(cont)

For the given roots

$$z = 0.6 \pm 0.4\,i$$

we want to perform pole placement

Notice that if the poles are found the zeros are also found.

$$H = K \cdot \frac{B\alpha}{A\alpha + B\beta}$$

zeros

poles

$$(z - 0.6 + 0.4i)(z - 0.6 - 0.4i)$$

$$z^2 - 1.2z + 0.520$$

Recall the Diophantine Equation

$$\underset{(n-1)}{\alpha}\ \underset{(n)}{A} + \underset{(n-1)}{\beta}\ \underset{(n)}{B} = \underset{(2n-1)}{D}$$

where $D$ is our characteristic polynomial

The orders must be checked. (include zero)

$$G(z) = \frac{B}{A} \cdot \frac{z^{-1} + z^{-2}}{1 - 2z^{-1} + z^{-2}} \quad \begin{matrix} (2) \\ (2) \end{matrix}$$

$$D = z^2 - 1.2z + 0.52 \quad (2)$$

The orders do not satisfy Diophantine

for $n = 2$

$2 \cdot n - 1 = 3$

$D$ needs another term.

(cont)

(cont)

Adding another root at zero fixes D

$$D = z^3 - 1.2z^2 + 0.52 \cdot z \qquad (2n-1) \checkmark$$

$$G(z) = \frac{B}{A} = \frac{0 \cdot z^2 + z + 1}{z^2 - 2z + 1} \cdot \left(\frac{T^2}{2}\right)$$

$$= \frac{b_0 z^2 + b_1 z + b_2}{a_0 z^2 + a_1 z + a_2}$$

$$D(z) = \frac{\beta}{\alpha} = \frac{\beta_0 z + \beta_1}{\alpha_0 z + \alpha_1} \qquad \text{(1st order)} \checkmark$$

2nd order Sylvester Matrix

Let $T = 0.2$

$$\frac{T^2}{2} = 0.020$$

$$E = \begin{bmatrix} a_2 & 0 & b_2 & 0 \\ a_1 & a_2 & b_1 & b_2 \\ a_0 & a_1 & b_0 & b_1 \\ 0 & a_0 & 0 & b_0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0.02 & 0 \\ -2 & 1 & 0.02 & 0.02 \\ 1 & -2 & 0 & 0.02 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(cont)

(cont)

$$M = E^{-1} \cdot D$$

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = E^{-1} \cdot D \qquad \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_0 \\ \beta_1 \\ \beta_0 \end{bmatrix}$$ notice order

(Perform calculations in Matlab)

$$M = E^{-1} \cdot D \qquad D = \begin{bmatrix} 0 \\ 0.52 \\ -1.2 \\ 1.0 \end{bmatrix}$$

$$M = \begin{bmatrix} 0.32 \\ 1.0 \\ -16.0 \\ 24.0 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_0 \\ \beta_1 \\ \beta_0 \end{bmatrix}$$

$$D(z) = \frac{\beta}{\alpha} = \frac{\beta_0 \cdot z + \beta_1}{\alpha_0 \cdot z + \alpha_1}$$

$$\boxed{D(z) = \frac{24 \cdot z - 16}{z + 0.32}}$$

(cont)

(cont)

K still needs to be chosen such that the error is zero for a step input,

$$H = K \cdot \frac{B\,\alpha}{A \cdot \alpha + B\beta}$$

$$= K \; \frac{\overset{B}{(z+1)}\;\overset{\alpha}{(z+0{,}32)} \cdot \overset{\frac{T^2}{2}}{0{,}02}}{z^3 - 1{,}2\,z^2 + 0{,}52\,z}$$

desired poles w/ added root

$$\lim_{z \to 1} K \cdot \frac{(z+1)(z+0{,}32)\,0{,}02}{z^3 - 1{,}2\,z^2 + 0{,}52 \cdot z} = 1$$

$$K = \frac{1 - 1{,}2 + 0{,}52}{(1+1)(1+0{,}32)\,0{,}02}$$

$$= \frac{0{,}320}{0{,}0528}$$

$$\boxed{K = 6{,}06}$$

Now K and D(z) can be used to build the controller

Listing 7 shows the Matlab code used to perform these calculations. Listing 8 shows the Matlab code to plot the step response as shown in Figure 11. The `sylvester` function is given in Appendix A.

```
%
% dd1s2_init.m
%
% Direct Design
%
% (double integrator)
%
%          1
% G(s):   ---
%         s^2
%

addpath('../lib');

% Designer provided specifications
T = 0.2;
% Characteristic polynomial with extra root
% so that 2n-1 = 3 for n = 2.
D = transpose(poly([(0.6 + 0.4i) (0.6 - 0.4i) 0]));

n = 2;

Gs = tf([1], [1 0 0]);
Gz = c2d(Gs, T, 'ZOH');
[Bz, Az] = tfdata(Gz, 'v');

E = sylvester(Az, Bz);

% D, Alpha and Beta are in descending order
% so they will be reversed.

%M = E^-1*transpose(D);
M = E\flipud(D);

% Alpha = a0*z + a1
% Beta = b0*z + b1
Alpha = fliplr(transpose(M(1:n)));
Beta = fliplr(transpose(M((n+1):end)));

Dz = tf(Beta, Alpha, T);

% To find K, the limit should go to 1
% for a step input.
% (refer to the notes for a better description)
K = sum(D)/sum(conv(Bz, Alpha));
```

Listing 7: Matlab script to find Direct Design of double integrator.

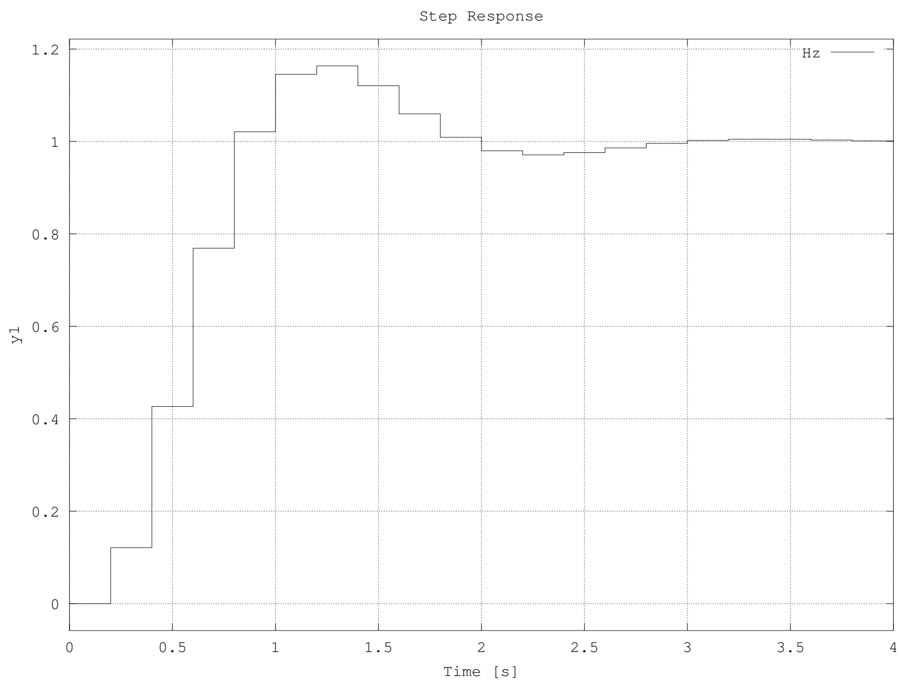



Figure 11: Step response of double integrator for controller built using Direct Design. Uses a time step of 0.2 seconds.

```
%
% dd1s2_plot.m
%

Tf = 4;  % time final (sec)

dd1s2_init;  % dd1s2_init.m

Hz = K*Gz/(1 + Dz*Gz);

step(Hz, Tf);

print('dd1s2_plot.eps', '-deps');
```

Listing 8: Matlab script to plot the response of the double integrator.

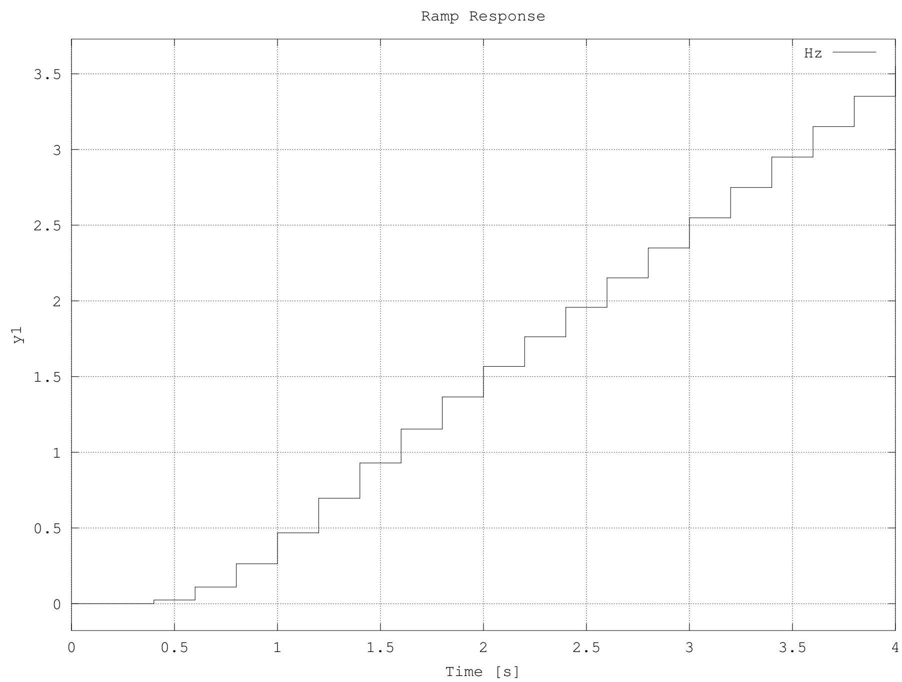



Figure 12: Ramp response of double integrator for controller built using Direct Design. Uses a time step of 0.2 seconds.

Listing 9 shows the Matlab code to plot the ramp response as shown in Figure 12.

```
%
% dd1s2_plot.m
%
% ramp response
%

Tf = 4;  % time final (sec)

dd1s2_init;  % dd1s2_init.m

Hz = K*Gz/(1 + Dz*Gz);

ramp(Hz, Tf);

print('dd1s2r_plot.eps', '-deps');
```

Listing 9: Matlab script to plot the ramp response of the double integrator.

**Example 2**

11/25/13

## Direct Design

$$G(s) = \frac{1}{s(s+1)} \qquad T = 0.2$$

roots : $z = 0.6 \pm 0.4i$

---

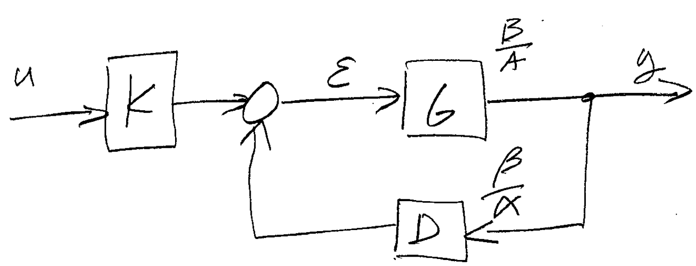
$$G(z) = (1 - z^{-1})\, \mathcal{Z}\left[\frac{G(s)}{s}\right]$$

$$= (1 - z^{-1})\, \mathcal{Z}\left[\frac{1}{s^2(s+1)}\right]$$

$$= \frac{(T + e^{-T} - 1) z^{-1} + [1 - e^{-T}(T+1)] z^{-2}}{1 - (1 + e^{-T}) z^{-1} + e^{-T} z^{-2}}$$

(see ZOH general $T$)

$$G(z) = \frac{0.0187 z^{-1} + 0.0175 z^{-2}}{1 - 1.8187 z^{-1} + 0.8187 z^{-2}}$$




$$\frac{y}{u} = H = K \cdot \frac{G}{1 + DG}$$

$$= K \cdot \frac{B\alpha}{A\alpha + B\beta}$$

(cont)

(cont)

$$G(z) = \frac{B}{A} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

$$= \frac{0 + 0.0187 z^{-1} + 0.0175 z^{-2}}{1 - 1.8187 z^{-1} + 0.8187 z^{-2}}$$

$$E = \begin{bmatrix} a_2 & 0 & b_2 & 0 \\ a_1 & a_2 & b_1 & b_2 \\ a_0 & a_1 & b_0 & b_1 \\ 0 & a_0 & 0 & b_0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.8187 & 0 & 0.0175 & 0 \\ -1.8187 & 0.8187 & 0.0187 & 0.0175 \\ 1 & -1.8187 & 0 & 0.0187 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D = d_0 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}$$

$$= z^3 - 1.2 z^2 + 0.520 z + 0$$

$$D = \begin{bmatrix} d_3 \\ d_2 \\ d_1 \\ d_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.52 \\ -1.2 \\ 1 \end{bmatrix}$$

$$M = E^{-1} \cdot D \qquad M = \begin{bmatrix} \alpha_1 \\ \alpha_0 \\ \beta_1 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} 0.242 \\ 1 \\ -11.30 \\ 20.1 \end{bmatrix}$$

(calc using Matlab)

(cont)

(cont)

Perform pole placement for the given roots,

$$z = 0.6 \pm 0.4i$$

$$(z - 0.6 + 0.4i)(z - 0.6 - 0.4i)$$

$$z^2 - 1.2z + 0.520 = D$$

$$H = K \cdot \frac{B\alpha}{A\alpha + B\beta}$$

$$\underset{(n-1)}{\alpha}\, \underset{(n)}{A} + \underset{(n-1)}{\beta}\, \underset{(n)}{B} = \underset{(2n-1)}{D} \qquad \text{(Diophantine)}$$

$$G(z) = \frac{B}{A} = \frac{0 + 0.0187z^{-1} + 0.0175z^{-2}}{1 - 1.8187z^{-1} + 0.8187z^{-2}} \qquad \begin{matrix} \text{order} \\ (2) \\ (2) \end{matrix}$$

For a second order $(n=2)$, $D$ needs another root to satisfy the Diophantine equation. Another can be added at zero,

$$D = z^3 - 1.2z^2 + 0.520z \qquad (2n-1) \checkmark$$

(cont)

(cont)

$$D(z) = \frac{\beta}{\alpha} = \frac{\beta_0 z + \beta_1}{\alpha_0 z + \alpha_1}$$

$$\boxed{D(z) = \frac{20.1 z - 11.30}{z + 0.242}}$$

Still need to find $K$

$$H = K \cdot \frac{B\alpha}{A\alpha + B\beta}$$

$$= K \cdot \frac{\overbrace{(0.0187 z + 0.0175)}^{B}\overbrace{(z + 0.242)}^{\alpha}}{z^3 - 1.2 z^2 + 0.52 z}$$

desired poles w/root

Find $K$ so that it goes to 1 for a step response.

$$\lim_{z \to 1} K \cdot \frac{(0.0187 z + 0.0175)(z + 0.242)}{z^3 - 1.2 z^2 + 0.52 z} = 1$$

$$\boxed{K = 7.11}$$

Listing 10 shows the Matlab code used to perform these calculations. Listing 11 shows the Matlab code to plot the step response as shown in Figure 13. The `sylvester` function is given in Appendix A.

```
%
% dd1s2s_init.m
%
% Direct Design
%
%              1
% G(s):   ---------
%          s(s + 1)
%

addpath('../lib');

% Designer provided specifications
T = 0.2;
% Characteristic polynomial with extra root
% so that 2n-1 = 3 for n = 2.
D = transpose(poly([(0.6 + 0.4i) (0.6 - 0.4i) 0]));

n = 2;

Gs = tf([1], [1 1 0]);
Gz = c2d(Gs, T, 'ZOH');
[Bz, Az] = tfdata(Gz, 'v');

E = sylvester(Az, Bz);

% D, Alpha and Beta are in descending order
% so they will be reversed.

%M = E^-1*transpose(D);
M = E\flipud(D);

% Alpha = a0*z + a1
% Beta = b0*z + b1
Alpha = fliplr(transpose(M(1:n)));
Beta = fliplr(transpose(M((n+1):end)));

Dz = tf(Beta, Alpha, T);

% To find K, the limit should go to 1
% for a step input.
% (refer to the notes for a better description)
K = sum(D)/sum(conv(Bz, Alpha));
```

Listing 10: Matlab script to perform the Direct Design calculations.

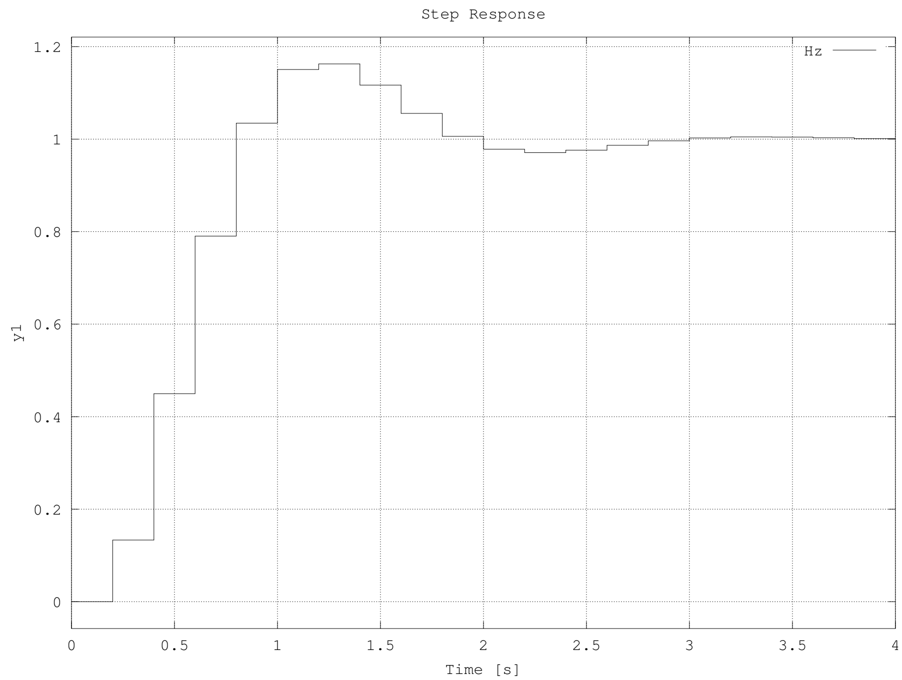



Figure 13: Step response of controller built using Direct Design. Uses a time step of 0.2 seconds.

```
%
% dd1s2s_plot.m
%
% step response
%

Tf = 4;  % time final (sec)

dd1s2s_init;  % dd1s2_init.m

Hz = K*Gz/(1 + Dz*Gz);

step(Hz, Tf);

print('dd1s2s_plot.eps', '-deps');
```

Listing 11: Matlab script to plot the step response.

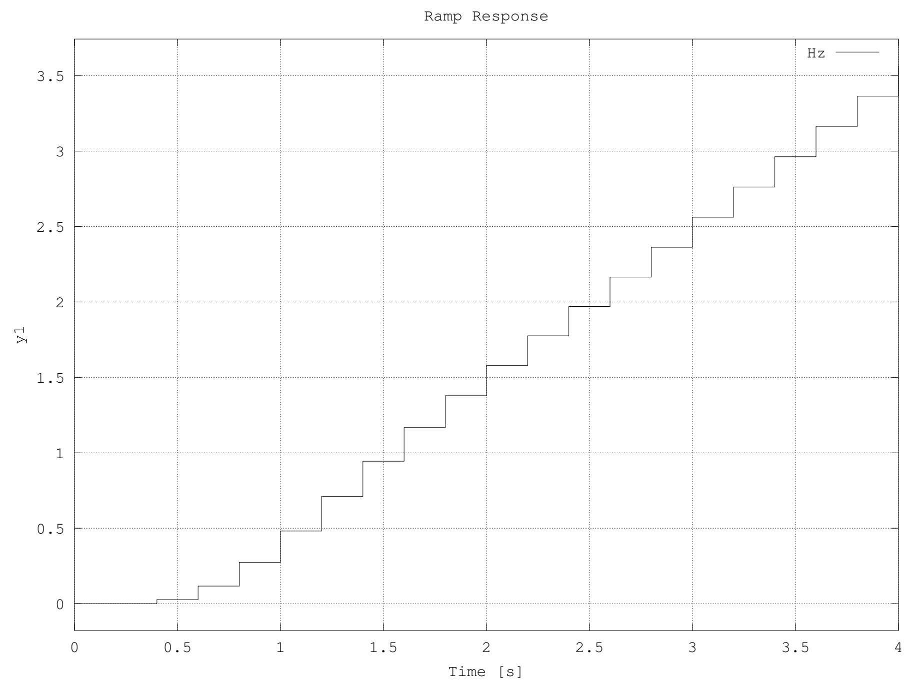



Figure 14: Ramp response of controller built using Direct Design. Uses a time step of 0.2 seconds.

Listing 12 shows the Matlab code to plot the ramp response as shown in Figure 14.

```
%
% dd1s2s_plot.m
%
% ramp response
%

Tf = 4;  % time final (sec)

dd1s2s_init;  % dd1s2_init.m

Hz = K*Gz/(1 + Dz*Gz);

ramp(Hz, Tf);

print('dd1s2sr_plot.eps', '-deps');
```

Listing 12: Matlab script to plot the ramp response.

# 21 Model Matching ($G_{model}$)

Suppose the perfect system was created and then the plant changed. How could the system response be reproduced identically with this new plant? One solution is to use Model Matching[9].

Given a model system ($G_{model}$) a controller is found for the given plant ($G(z)$).

$$\frac{Y(z)}{R(z)} = G_{model} = \frac{B_m(z)}{A_m(z)} \tag{9}$$

For more details refer to Ogata.[10]

[9] If this method was called System Matching it would make more sense because it matches $H(z)$.
[10] Ogata, see n. **??**, Pg. 532.

**Example 1**

# Model Matching

10/15/13

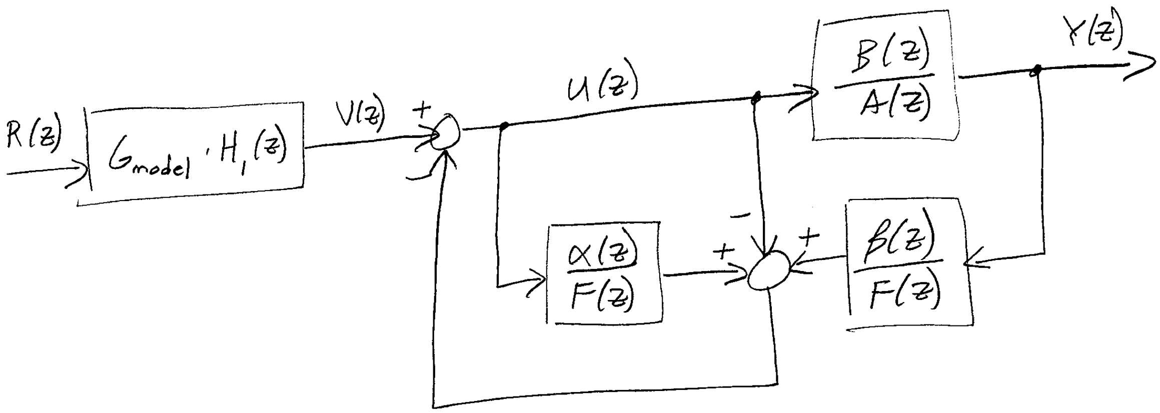
[Ogata 532]

$$G(z) = \frac{B}{A} = \frac{0.3679z + 0.2642}{(z-0.3679)(z-1)}$$

$T = 1$

$$G_{model} = \frac{Y_m(z)}{R_m(z)} = \frac{0.62z - 0.3}{z^2 - 1.2z + 0.52}$$

(H(z) to acheive)




(cont)

(cont)

$$\alpha A + \beta B = \underset{(2n-1)}{D} \qquad \text{(Diophantine)}$$

$n = 2$

$m = 1$

$$D = \underset{(n)}{H} \cdot \underset{(n-1)}{F}$$

$$= \underset{(n-1)}{F} \cdot \underset{(m)}{B} \cdot \underset{(n-m)}{H_1}$$

$H_1$ can be any stable $(n-m)$ order polynomial

Let

$$H_1(z) = z + 0.5$$

$F$ can be any stable $(n-1)$th degree polynomial

Let

$$F(z) = z$$

$$D(z) = F \cdot B \cdot H_1$$

$$= z(0.3679z + 0.2642)(z + 0.5)$$

$$= 0.3679z^3 + 0.4481z^2 + 0.1321z$$

$$= d_0 z^3 + d_1 z^2 + d_2 z + d_3$$

(cont)

(cont)

Setup and solve using Diophantine

$$\frac{B}{A} = \frac{0.3679 z + 0.2642}{z^2 - 1.3679 + 0.3679}$$

$$= \frac{b_0 z^2 + b_1 z + b_2}{a_0 z^2 + a_1 z + a_2}$$

$$E = \begin{bmatrix} a_2 & 0 & b_2 & 0 \\ a_1 & a_2 & b_1 & b_2 \\ a_0 & a_1 & b_0 & b_1 \\ 0 & a_0 & 0 & b_0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.3679 & 0 & 0.2642 & 0 \\ -1.3679 & 0.3679 & 0.3679 & 0.2642 \\ 1 & -1.3679 & 0 & 0.3679 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} d_3 \\ d_2 \\ d_1 \\ d_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1321 \\ 0.4481 \\ 0.3679 \end{bmatrix}$$

$$E^{-1} \cdot D = \begin{bmatrix} \alpha_1 \\ \alpha_0 \\ \beta_1 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} 0.264 \\ 0.368 \\ -0.368 \\ 1.868 \end{bmatrix}$$

(cont)

(cont)

$$\frac{\beta}{\alpha} = \frac{\beta_0 z + \beta_1}{\alpha_0 z + \alpha_1}$$

$$\boxed{\frac{\beta}{\alpha} = \frac{1.87z - 0.368}{0.368z + 0.264}}$$

Listing 13 shows the Matlab code used to perform these calculations. Listing 14 shows the Matlab code to plot the step response as shown in Figure 15. The `sylvester` function is given in Appendix A.

```
%
% mm3679_init.m
%
% Model Matching
%
% Example from Ogata Pg. 532
%
% Requires: Octave with control toolbox
%

addpath('../lib');

T = 1;

Bz = [0.3679 0.2642];
Az = [1 -1.3679 0.3679];
Gz = tf(Bz, Az, T);
set(Gz, 'inname', 'u1', 'outname', 'y1');

% model to match
Bm = [0.62 -0.3];
Am = [1 -1.2 0.52];
Gm = tf(Bm, Am, T);

% Chosen to acheive order 2n-1 in D for Ackerman's
% order n=2, m=1
F = [1 0];  % z
H1 = [1 0.5];  % z + 0.5
D = conv(F, conv(H1, Bz));  % D = F*B*H1
D = flipud(transpose(D));

E = sylvester(Az, Bz);

M = E\D;

Alpha = fliplr(transpose(M(1:2)));
Beta = fliplr(transpose(M(3:4)));

GmH1 = tf(conv(Bm, H1), Am, T);
set(GmH1, 'inname', 'r1', 'outname', 'v1');

AlphaF = tf(Alpha, F, T);
set(AlphaF, 'inname', 'u1', 'outname', 's1');

BetaF = tf(Beta, F, T);
set(BetaF, 'inname', 'y1', 'outname', 's3');

sum1 = sumblk('u1 = v1 - q1');
sum2 = sumblk('q1 = s1 + s3 - u1');

sys = connect(GmH1, sum1, AlphaF, BetaF, sum2, Gz, 'r1', 'y1');
```

Listing 13: Matlab script to perform Model Matching for Example 1.

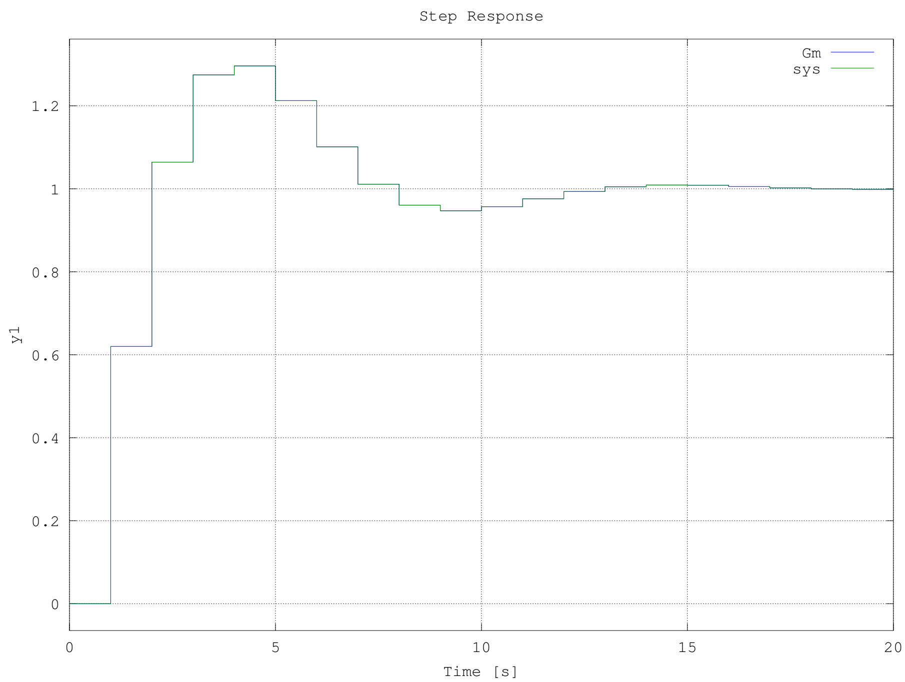



Figure 15: Step response of controller built using Model Matching. Notice that $G_{model}$ is identical to the whole system as expected.

```
%
% mm3679_plot.m
%

clear;

addpath('../lib');

mm3679_init;

step(Gm, sys)
print('mm3679_plot.eps', '-deps2', '-color');
```

Listing 14: Matlab script to plot the step response of Example 1.

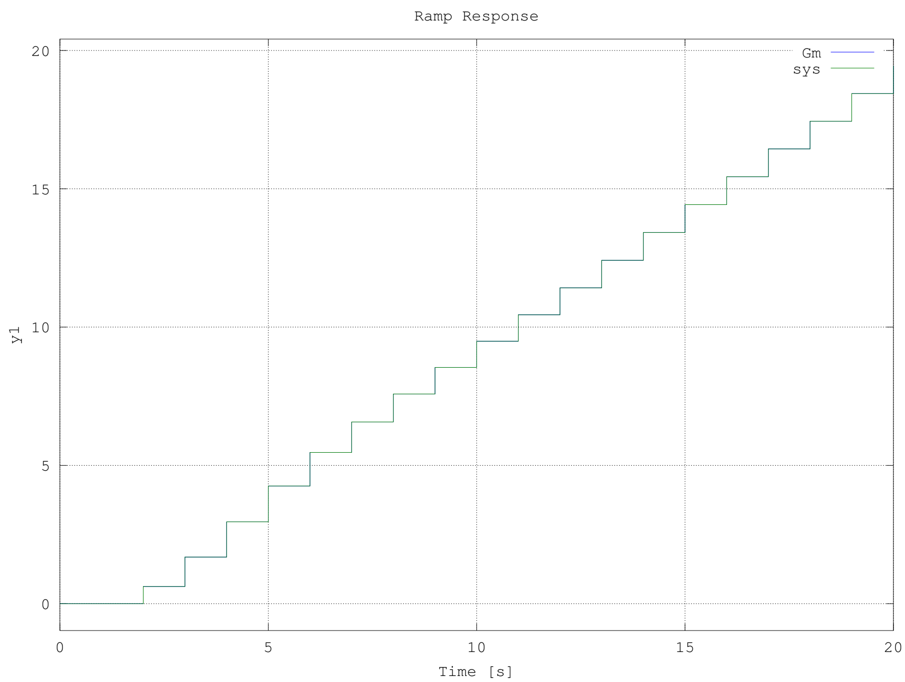



Figure 16: Ramp response of controller built using Model Matching.

Listing 15 shows the Matlab code to plot the ramp response as shown in Figure 16.

```
%
% mm3679r_plot.m
%

clear;

addpath('../lib');

mm3679_init;

ramp(Gm, sys)
print('mm3679r_plot.eps', '-deps2', '-color');
```

Listing 15: Matlab script to plot the ramp response of Example 1.

## 22 Poll Placement with Ackerman's Formula

# 23 Control Law, Full Order, No Predicition

A Control Law system behaves as a regulator as shown in Figure 17. In this case it is assumed that all outputs are observable and so are fed back. Later examples will investigate partial feedback with estimation. Ackermann's formula is used to find $K$.

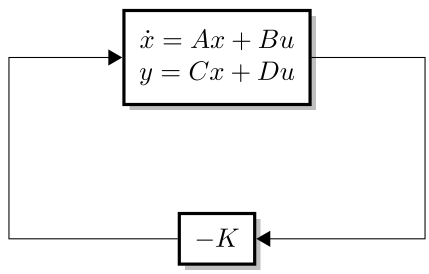



Figure 17: Control Law system with full feedback. It behaves as a regulator since not input is available.

## 23.1 Example 1

For this example the calculation of $K$ is shown in Listing 16. The `myacker` function is a re-implementation of the Matlab `acker` function and is given in Appendix B. Figure 18 shows the response of this system.

```
%
% cl_fonp02.m
%
% Control Law, Full Order, No Estimation
% Plots of two sets of roots are given.
%
% Derived from Example 8.3 in Franklin & Powell on Pg. 288.
%

clear;

addpath('../lib');

T = 0.4;  % time step

% Continuous state space system
A = [0 1 0 0;
        -0.91 -0.036 0.91 0.036;
        0 0 0 1;
        0.091 0.0036 -0.091 -0.0036];
B = [0; 0; 0; 1];

n = length(A);  % order
Gs = ss(A, B, eye(n), zeros(n,1));
% convert to digital
Gz = c2d(Gs, T, 'ZOH');
[Phi, Gamma] = ssdata(Gz);

% y = H*x, filter
H = zeros(1,n);
H(1) = 1;
% create a gain of H using D and zeroing others
Hz = ss(zeros(1,1), zeros(1,n), zeros(1,1), H, T);

% Find K
% roots (arbitrary)
```

```matlab
z1 = [0.9 0.9 0.9 0.9];
z2 = [(0.9 + 0.05i) (0.9 - 0.5i) (0.8 + 0.4i) (0.8 - 0.4i)];
% can't use 'place' with identical roots
%K1 = place(Phi, Gamma, z1);
K1 = myacker(Phi, Gamma, z1);
K2 = myacker(Phi, Gamma, z2);
% create a gain of K using D and zeroing others
K1z = ss(zeros(1,1), zeros(1,n), zeros(1,1), K1, T);
K2z = ss(zeros(1,1), zeros(1,n), zeros(1,1), K2, T);

% Build system(s)
X1z  = feedback(Gz, K1z, -1);
CL1z = series(X1z, Hz);
X2z  = feedback(Gz, K2z, -1);
CL2z = series(X2z, Hz);
% simplify, structural pole/zero cancellation
CL1z = sminreal(CL1z);
CL2z = sminreal(CL2z);

% Simulate
Tend = 50;
u = zeros([(Tend/T) 1]);
% x0, initial conditions
x0 = zeros(length(CL1z.B), 1);
x0(1) = 1;
[y1,t1] = lsim(CL1z, u, [], x0);
[y2,t2] = lsim(CL2z, u, [], x0);

% Plot
clf;
figure(1);
subplot(2, 1, 1);
stairs(t1, y1);
grid on;
axis tight;
title('Regulator Response, roots: (0.9 0.9 0.9 0.9)');
%xlabel('time (sec)');
ylabel('y');

subplot(2, 1, 2);
stairs(t2, y2);
grid on;
axis tight;
title('Regulator Response, roots: (0.9 + 0.05i) (0.9 - 0.5i) (0.8 + 0.4i) (0.8 - 0.4i)');
xlabel('time (sec)');
ylabel('y');

% Print to file
print('cl_fonp02.eps', '-depsc2');
```

Listing 16: Matlab script to calculate $K$ using Ackermann's formula.

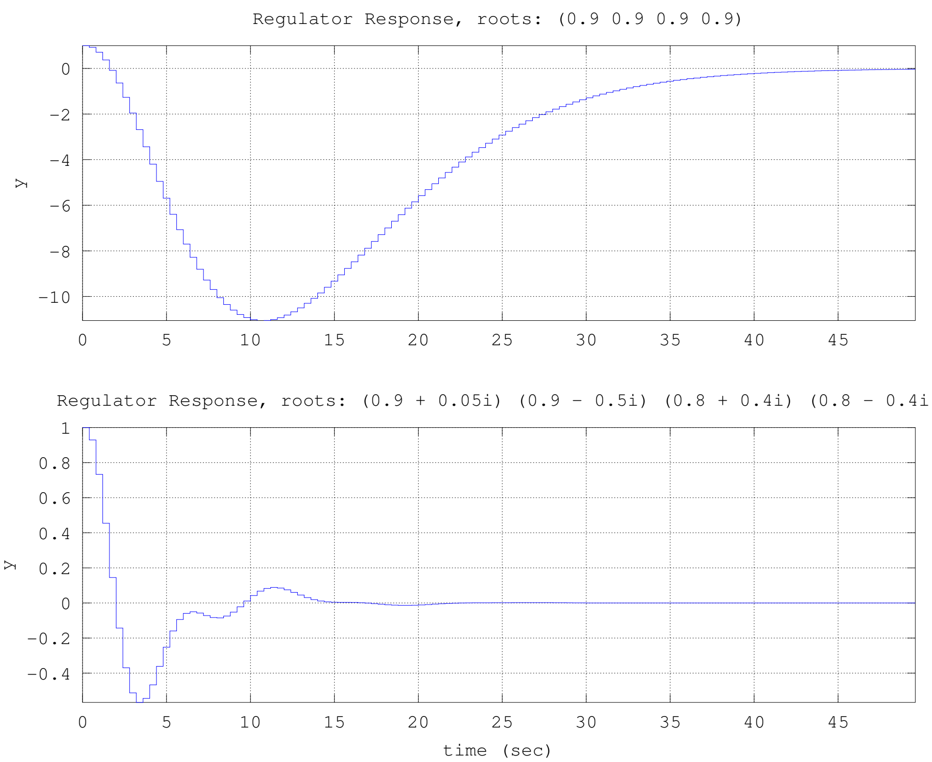



Figure 18: Response of Control Law designed regulator system for two different sets of roots.

# 24 Control Law, Full Order, Predicitor Estimator

## 24.1 Example 1

```
%
% cl_fope01.m
%
% Control Law, Full Order, Predictor Estimator.
% Predictor is shown along with non-predictor.
%
% Derived from Example 8.5 on Pg. 294 of Franklin & Powell.
% Which is an expansion of Example 8.1 and 8.2.
%

clear;

T = 0.1;  % time step

Phi = [ 1 T;
               0 1];
Gamma = [T^2/2; T];

n = length(Phi);  % order
Gz = ss(Phi, Gamma, eye(n), zeros(n,1), T);

% y = H*x
H = [1 0];  % filter
Hz = ss(zeros(1,1), zeros(1,n), zeros(1,1), H, T);

% Find K
% roots (arbitrary)
z1 = [0.8 + 0.25i; 0.8 - 0.25i];
K1 = place(Phi, Gamma, z1);
% create a gain of K using D and zeroing others
K1z = ss(zeros(1,1), zeros(1,n), zeros(1,1), K1, T);

% Find Lp and L system, for predictor estimator
z2 = [(0.4 + 0.4i) (0.4 - 0.4i)];
Lp = place(Phi', H', z2)';
Lz = ss((Phi - Gamma*K1 - Lp*H), Lp, eye(n), zeros(n,1), T);

% Build system(s)
% no prediction
X1z = feedback(Gz, K1z, -1);
CL1z = series(X1z, Hz);
CL1z = sminreal(CL1z);  % simiplify, pole/zero cancellation
% predictor estimator
G1z = series(Gz, Hz);
G2z = series(Lz, K1z);
CL2z = feedback(G1z, G2z, -1);
CL2z = sminreal(CL2z);  % simiplify, pole/zero cancellation

% Simulate
Tend = 5;
u = zeros([(Tend/T) 1]);
% x0, initial conditions
x0 = zeros(length(CL1z.B), 1);
x0(1) = 1;
[y1,t1] = lsim(CL1z, u, [], x0);
x0 = zeros(length(CL2z.B), 1);
x0(1) = 1;
[y2,t2] = lsim(CL2z, u, [], x0);

```

```
% Plot
clf;
figure(1);
[ts1,ys1] = stairs(t1, y1);
[ts2,ys2] = stairs(t2, y2);
plot(ts1,ys1,ts2,ys2);
title('Regulator Response, Full Order Predictor Estimator');
grid on;
axis tight;
legend('non-predictor', 'predictor');
xlabel('time (sec)');
ylabel('y');

% Print to file
print('cl_fope01.eps', '-depsc2');
```

Listing 17: Matlab script to calculate $K$ and estimate $L$ using Ackermann's formula.

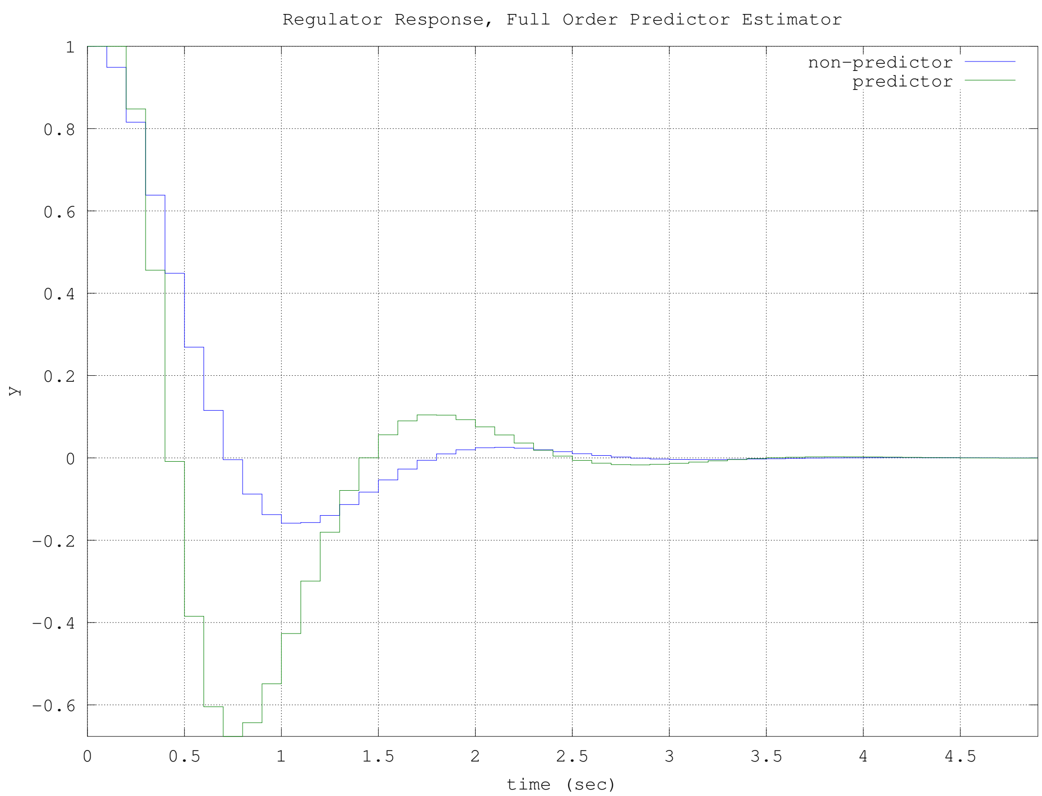



Figure 19: Response of Control Law system designed with a full order predictor estimator compared to a system with no prediction. Initial $x$ has one in first position with zeros elsewhere.

# 25 Control Law, Reduced Order Estimator

Partial feedback with estimation.

# References

Franklin, G.F., J.D. Powell, and M.L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley world student series. Addison-Wesley Longman, Incorporated, 1998. ISBN: 9780201331530.

Octave community. *GNU/Octave*. 2012. URL: `www.gnu.org/software/octave/`.

Ogata, K. *Discrete-Time Control Systems*. Prentice Hall International editions. Prentice-Hall International, 1995. ISBN: 9780133286427.

# A Sylvester Matrix Generation in Matlab

```
function [E] = sylvester(A, B)
% SYLVESTER Construct a Sylvester matrix of the two vectors.
%
% Given two vectors, the largest order is determined.
% Then a Sylvester matrix is constructed for that order.
%

nA = max(size(A));
nB = max(size(B));

n = max(nA, nB) - 1;  % order

% First create a matrix of zeros
E = zeros((n)*2, (n)*2);

% Then assign specific values for A and B

% A
for (col = 1:n)
        for (i = 1:nA)
                row = (nA - i) + col;
                E(row, col) = A(i);
        end
end

% B
for (col = (n+1):n*2)
        for (i = 1:nB)
                row = (nB - i) + (col - n);
                E(row, col) = B(i);
        end
end

endfunction
```

Listing 18: Matlab function to calculate a Sylvester Matrix.

# B Ackermann's Formula in Matlab

```

function [K] = myacker(A, B, z)
% MYACKER An implementation of ACKER.
%
% MYACKER is an implementation of the Matlab ACKER function.
%
% Given a feedback system it is necessary to find K such
% that the regulator will go to zero.
%
%        +--------------+
%     +-->| x' = Ax + Bu |---+
%     |  +--------------+    |
%     |                      |
%     |        +----+        |
%     +--------| -K |<-------+
%              +----+
%
% By specifying the desired roots (z) K can be found
% using Ackermann's Formula.
%
% The number of roots must match the order of the system.
```

```
% And the choice of roots determines how well the system performs.
%
% See also ACKER, PLACE.

n = size(A, 1);  % order of problem
zp = poly(z);

% Controllability Matrix
% [B AB A^2B ... A^(n-1)B]
W = ctrb(A, B);

% [0 0 ... 1]
en = zeros([1 n]);
en(n) = 1;

% alpha = A^n + z(1)*A^(n-1) + ... + z(n)*I
alpha = 0;
for i = 0:n
    if (i == n)
        % last
        alpha = alpha + zp(i+1)*eye(n);
    else
        alpha = alpha + zp(i+1)*A^(n-i);
    end
end

% z = eig(A - B*K)
K = real(en*(W\alpha));
```

Listing 19: Matlab function to calculate Ackermann's Formula. Designed as an example of how Matlab's acker could be implemented.

# C Quick Reference

# D Table of Laplace and Z-transforms