Proportional Control Design using Root Locus

1.0 Learning Outcomes 1
2.0 Introduction 2
3.0 Effect of the Proportional Control Gain on the Closed-Loop Poles 3
4.0 Rules of Root Locus 8
5.0 Exercises 15
6.0 Summary 17
Appendix A: Effect of Natural Frequency and Damping Ratio 17
Appendix B: MATLAB Code for the Examples in Section 3.0 18
Appendix C: Fully Worked Solutions to Exercises 19
Recommended Citation 23

1.0 Learning Outcomes

After studying this ACE-Lab learning guide, you should be able to demonstrate the Learning Outcomes listed in Table 1. On completion of the ACE-Lab learning guide, revisit Table 1 and confirm whether each Learning Outcome has been achieved.

Table 1: *Learning Outcomes from the Root Locus Learning Guide*

ID	Learning Outcome	Achieved
1.	Explain the effect of the proportional control gain K_p on the closed-loop pole locations on the s-plane, and how the paths of the poles form the root locus of a closed-loop control system.	☐
2.	State the assumptions of the rules of root locus: negative unity feedback and a proportional control gain K_p in the feedforward path, with the open-loop transfer function as the starting point.	☐
3.	Apply the rules of root locus to sketch the root locus of a closed-loop control system as K_p is varied from zero to infinity.	☐
4.	Use the root locus, with MATLAB, to select a value of K_p that meets a set of system performance criteria (SPC).	☐

2.0 Introduction

In the Routh-Hurwitz Stability Criterion Learning Guide, the stability of a closed-loop control system was shown to depend on the location of the roots of the characteristic equation, that is, the poles of the system on the s-plane. The root locus shows how the closed-loop pole locations change as the proportional control gain K_p is varied from zero to infinity.

For a closed-loop control system with a proportional control gain K_p in the feedforward path:

$$Y(s) = \left[\frac{K_p N(s)}{D(s) + K_p N(s)}\right] R(s) \tag{1}$$

The characteristic equation, $D(s) + K_p N(s) = 0$, determines the stability of the system and its transient response. Equation (1) shows that changing K_p changes the characteristic equation, and therefore the poles and their locations on the s-plane. Recall that, for a second-order system, a pole location corresponds to a radial line of constant damping ratio and an arc of constant natural frequency. Once the effect of K_p on the pole locations is understood, a value of K_p can be selected that gives a desired system response, that is, one that meets the system performance criteria (SPC).

When $K_p = 0$, the characteristic equation reduces to $D(s) = 0$, so the closed-loop poles start at the open-loop poles. An important question therefore arises:

How do the closed-loop poles move as the controller gain is increased, and where should they be placed?

Why is Understanding the Root Locus Important?

The main reasons why the root locus is important are:

i. As well as the Routh-Hurwitz stability criterion, the root locus provides information about the stability of a system;
ii. The root locus shows the potential performance of a system, as each point on it corresponds to a location on the s-plane, for example a damping ratio and natural frequency for a second-order system;
iii. A particular location can be selected to give a desired response, for example a damping ratio of 0.6, which gives approximately 10% peak overshoot for a second-order system;
iv. Root Locus helps in the design of three-term proportional, integral and derivative (PID) controllers;
v. Root Locus is an easy-to-use visual technique for designing and analysing closed-loop control systems with negative unity feedback and a proportional control gain K_p.

3.0 Effect of the Proportional Control Gain on the Closed-Loop Poles

Before the rules of root locus are given, three examples show how the closed-loop poles move as K_p is increased, for systems with a relative degree of 1, 2 and 3. Recall that the relative degree is $rd = n - m$, where n is the number of poles and m the number of zeros of the open-loop system.

Water Tank Level Control (Relative Degree of 1)

The open-loop transfer function of a water tank, from inflow to level, is:

$$G(s) = \frac{H(s)}{F_{in}(s)} = \frac{1}{As + \frac{\rho g}{R}} \tag{2}$$

where $F_{in}(s)$ is the inflow, $H(s)$ the liquid level, A the cross-sectional area of the tank, ρ the liquid density, g the acceleration due to gravity and R the resistance of the outflow restriction. Here the model coefficients are kept constant, with $A = 2$ and $\rho g/R = 5$, and only K_p is varied. With an ideal sensor, the closed-loop transfer function is:

$$G_{CL}(s) = \frac{K_p}{(2s + 5) + K_p} \tag{3}$$

The relative degree is 1, as there is one real pole and no zeros. As K_p increases from zero, the pole moves to the left along the negative real axis, as shown in Figure 1 and Table 2. At $K_p = 0$ it is at the open-loop pole, $s = -2.5$. Moving left corresponds to a smaller time constant and therefore a faster response, as shown in Figure 2. The MATLAB code for these figures is given in Appendix B.

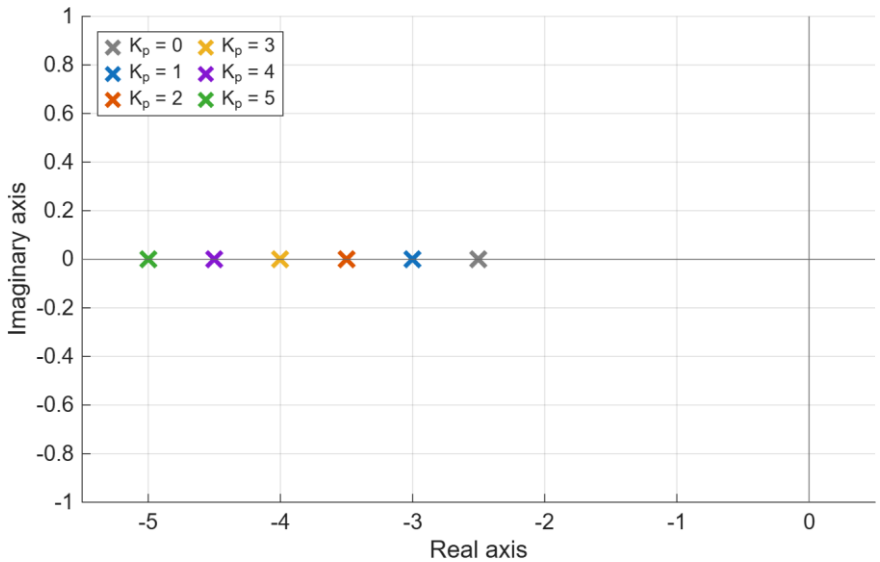


Figure 1: *Pole Locations as K_p is Increased, for the Water Tank Level Control System*

Table 2: *Pole Location and Time Constant as K_p is Increased, for the Water Tank Level Control System*

Proportional control gain, K_p	Pole location	Time constant, τ (s)
0	−2.5	0.40
1	−3.0	0.33
2	−3.5	0.29
3	−4.0	0.25
4	−4.5	0.22
5	−5.0	0.20

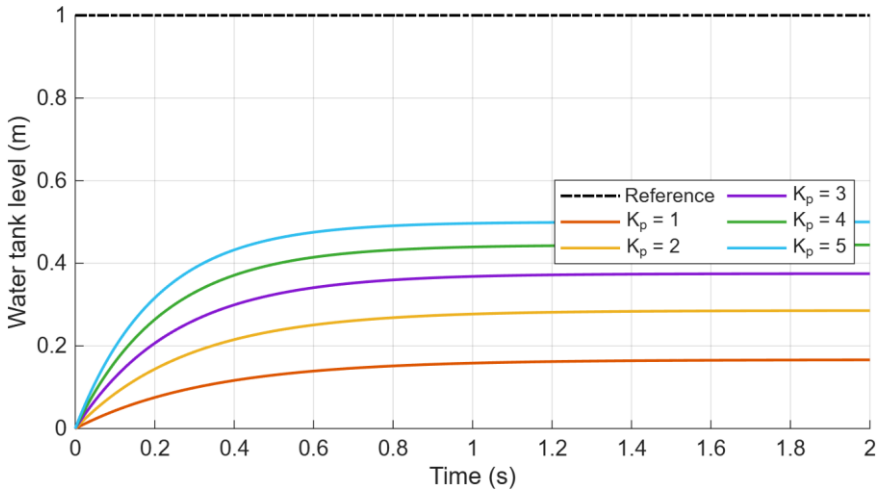


Figure 2: *Step Responses as K_p is Increased, for the Water Tank Level Control System*

DC Motor Angle Control (Relative Degree of 2)

The open-loop transfer function of a DC motor, from voltage to angle, is:

$$G(s) = \frac{113.7}{s^2 + 3.398s} \tag{4}$$

With a proportional control gain K_p and an ideal sensor, the closed-loop transfer function is:

$$G_{CL}(s) = \frac{113.7K_p}{s^2 + 3.398s + 113.7K_p} \tag{5}$$

The relative degree is 2. At $K_p = 0$ there are two real poles, one at the origin, because the open-loop system is Type 1. As K_p increases, the poles move towards each other, meet (real and equal poles, critical damping with $\zeta = 1$), and then become a complex conjugate pair, as shown in Figure 3. Once complex, their real part stays constant at -1.699 and only their imaginary parts

change, so the poles always remain in the left-half plane, as expected for a relative degree of 2.

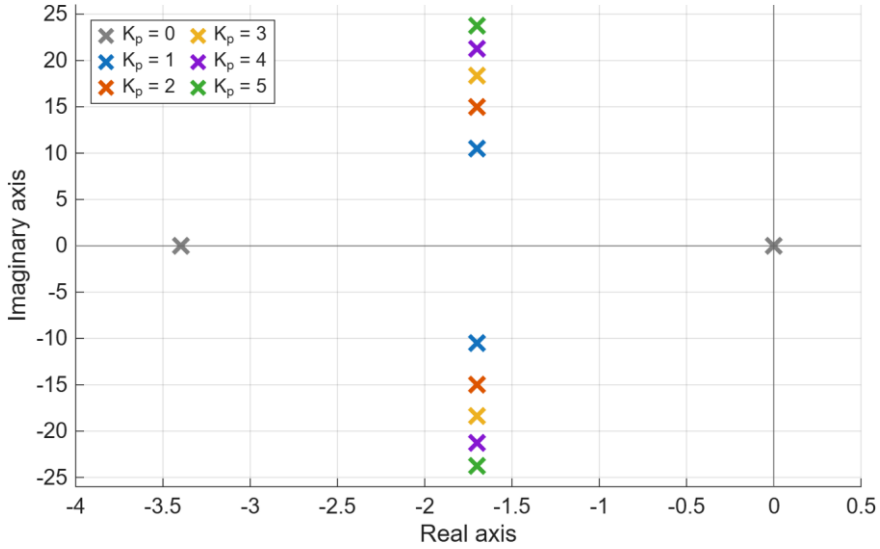


Figure 3: *Pole Locations as K_p is Increased, for the DC Motor Angle Control System*

Table 3 shows that as the poles move away from the real axis, the natural frequency increases and the damping ratio decrease. The response is therefore faster but more oscillatory, with a larger peak overshoot, as shown in Figure 4. Because the open-loop system is Type 1, the closed-loop response has zero steady-state error (SSE).

Table 3: *Pole Location, Natural Frequency and Damping Ratio as K_p is Increased, for the DC Motor Angle Control System*

K_p	**Pole location**	ω_n **(rad/s)**	**Damping ratio,** ζ
0	0, −3.40	—	overdamped
1	$-1.7 \pm j10.5$	10.7	0.159
2	$-1.7 \pm j15.0$	15.1	0.113
3	$-1.7 \pm j18.4$	18.5	0.092
4	$-1.7 \pm j21.3$	21.3	0.080
5	$-1.7 \pm j23.8$	23.8	0.071

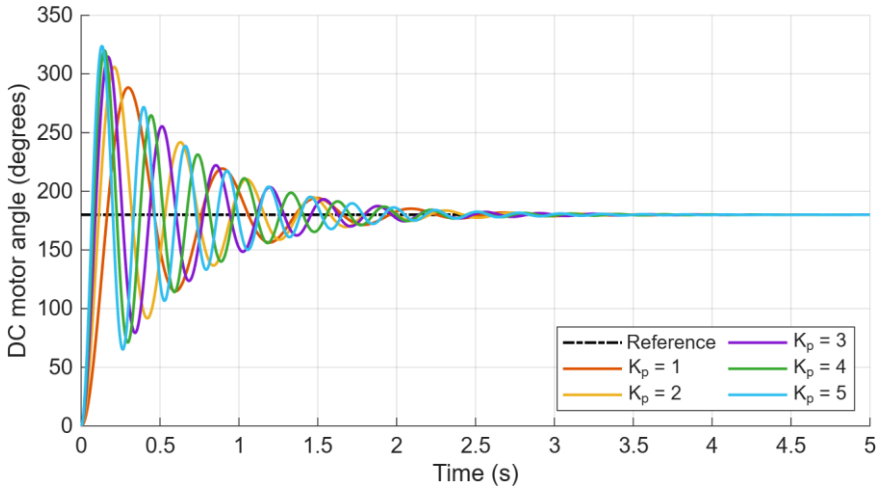


Figure 4: *Step Responses as K_p is Increased, for the DC Motor Angle Control System*

Soldering Robot Displacement Control (Relative Degree of 3)

A soldering robot moves a soldering iron to a desired position, 0.1 m, as illustrated in Figure 5, with the closed-loop control system shown in Figure 6. The open-loop transfer function, with a mass $m = 0.1$ kg and damping $c = 10$ Ns/m, is:

$$G(s) = \frac{10}{s(s+10)(ms+c)} = \frac{10}{s(s+10)(0.1s+10)} \tag{6}$$

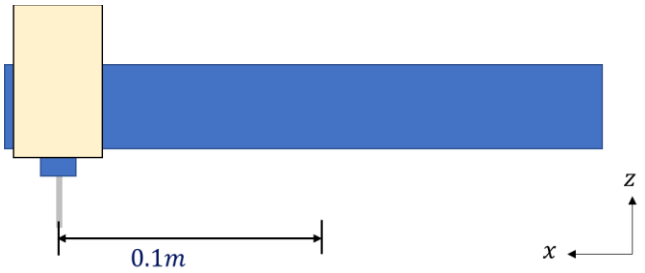


Figure 5: *Longitudinal Displacement Control of a Soldering Robot*

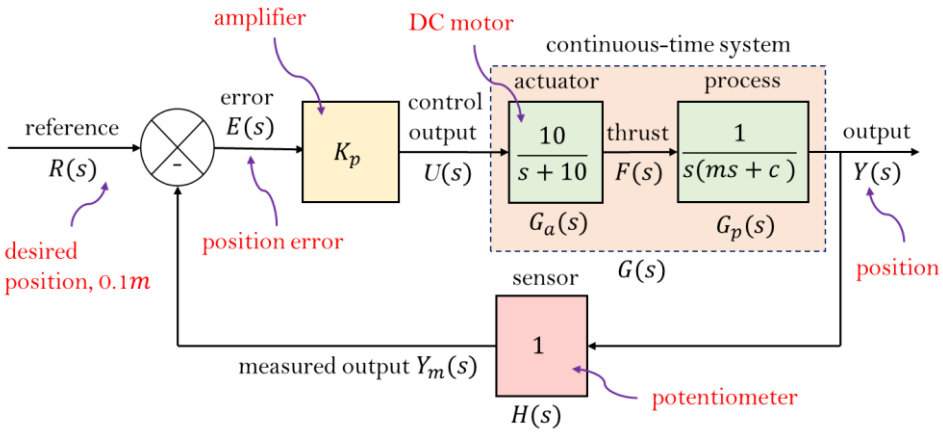


Figure 6: *Closed-Loop Control System for the Soldering Robot*

The closed-loop transfer function is:

$$G_{CL}(s) = \frac{10K_p}{s(s+10)(0.1s+10)+10K_p} \tag{7}$$

The relative degree is 3. At $K_p = 0$ there are three real poles, at 0, −10 and −100. As K_p increases, the two right-hand poles meet and become a complex conjugate pair, but, unlike the DC motor, both their real and imaginary parts now change, and they move towards the right-half plane, as shown in Figure 7. The left-hand pole stays real, and moves further left, as in the water tank example. As explained in the Routh-Hurwitz Stability Criterion Learning Guide, a system with a relative degree of 3 becomes unstable for a large enough gain.

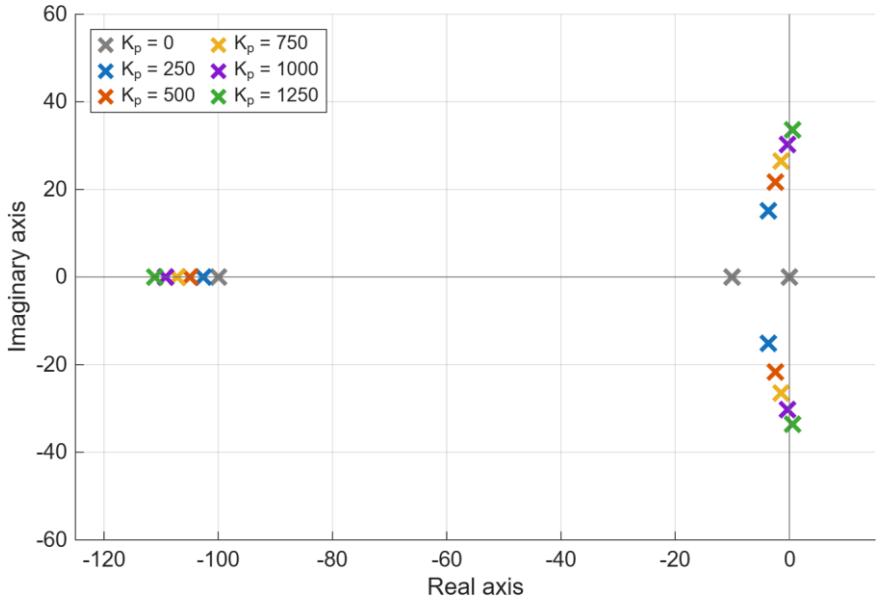


Figure 7: *Pole Locations as K_p is Increased, for the Soldering Robot*

Figure 8 shows that, as K_p increases, the response becomes more oscillatory. The open-loop system is Type 1, so the steady-state error is zero whenever the closed-loop system is stable. The complex poles reach the imaginary axis at $K_p = 1100$; above this, for example at $K_p = 1250$, they lie in the right-half plane, and the response grows without limit, as shown in Figure 9.

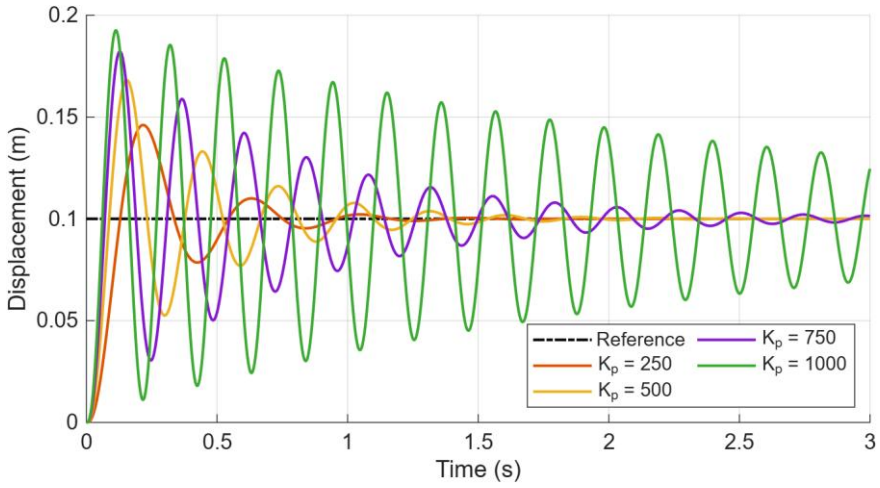


Figure 8: *Step Responses as K_p is Increased, for the Soldering Robot*

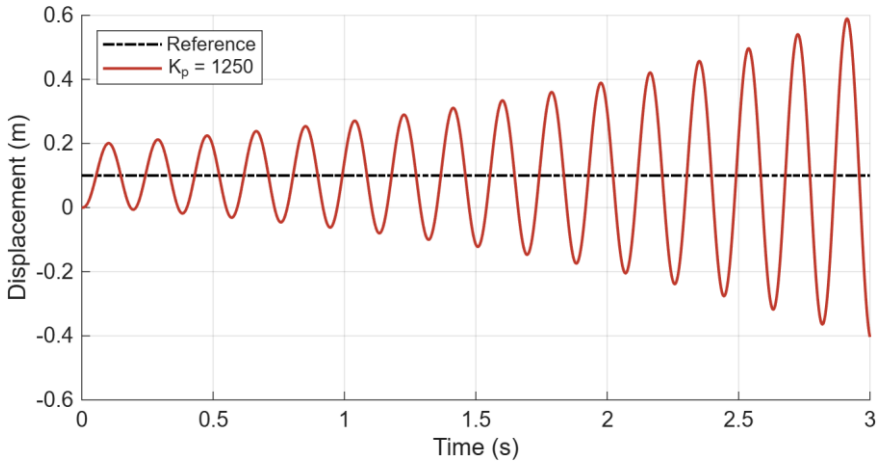


Figure 9: *Unstable Step Response of the Soldering Robot with $K_p = 1250$ (Poles in the Right-Half Plane)*

4.0 Rules of Root Locus

The rules for constructing a root locus were introduced by W. R. Evans in 1948. The root locus is a graphical method that shows how the closed-loop poles move as K_p is increased from 0 to ∞. The rules start from the open-loop transfer function $G(s)$, and assume the closed-loop form in Figure 10:

i. negative unity feedback;
ii. a proportional control gain K_p in the feedforward path.

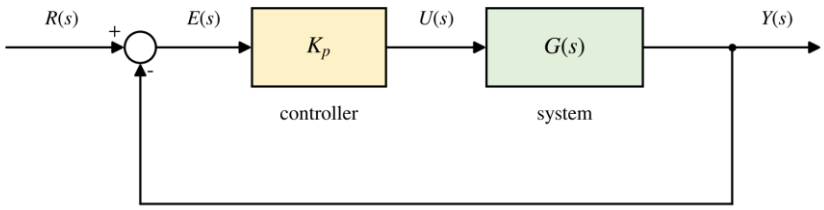


Figure 10: *Closed-Loop Control System Assumed by the Rules of Root Locus, with Negative Unity Feedback and a Proportional Control Gain* K_p

The rules of root locus are presented in different forms by different authors. Nine rules are used here, as listed in Table 4. Rules 1 to 7 are used for systems with a relative degree of 1 or 2, and Rules 1 to 8 for a relative degree of 3 or more. Rule 9 is given in its basic form for completeness; for precise values at points on the root locus, MATLAB should be used. The two examples that follow show how the rules are applied.

Table 4: *The Rules of Root Locus*

Rule	Description
1	The root locus is symmetrical about the real axis.
2	The number of **branches** of the root locus, the paths of the closed-loop poles, equals the number of open-loop poles.
3	A point on the real axis lies on the root locus if the total number of open-loop poles and zeros to its right is odd (the **real-axis segments**).
4	Each branch starts at an open-loop pole ($K_p = 0$) and ends either at an open-loop zero or at infinity ($K_p = \infty$). The number of branches that end at infinity equals the relative degree, $rd = n - m$.
5	If a real-axis segment lies between two poles, the branches meet and **break away** from the real axis between them. For two poles and no zeros the breakaway point is exactly midway; otherwise, the midpoint is a first approximation.
6	Branches that go to infinity approach straight-line **asymptotes**, which meet the real axis at $\sigma_a = \frac{\sum \text{poles} - \sum \text{zeros}}{rd}$.
7	The **asymptote angles**, measured from the positive real axis, are $\theta_a = \frac{(2k+1)180^\circ}{rd}$, $k = 0,1,\ldots,rd-1$; that is, the asymptotes are spaced $360^\circ/rd$ apart. Rules 6 and 7 allow the asymptotes to be added to the sketch.
8	The **imaginary-axis crossing** is found with the Routh-Hurwitz stability criterion, which gives the range of K_p for a stable system. This is only needed for a relative degree of 3 or more.

Rule	Description
9	A point lies on the root locus if it satisfies the **angle criterion**: the sum of the angles from the open-loop zeros to the point, minus the sum of the angles from the open-loop poles, equals an odd multiple of 180°.

Example 1: DC motor angle control (relative degree of 2)

For the open-loop DC motor transfer function $G(s) = \frac{113.7}{s^2+3.398s} = \frac{113.7}{s(s+3.398)}$, apply the rules of root locus and sketch the root locus on the s-plane.

Solution:

Rule 2: there are two open-loop poles, at 0 and −3.398, so there are two branches.

Rule 3: between 0 and −3.398 there is one pole to the right (odd), so this segment is on the root locus. To the left of −3.398 there are two (even), so it is not.

Rule 4: $rd = 2 - 0 = 2$, so both branches end at infinity.

Rule 5: with two poles and no zeros, the branches break away exactly midway between the poles, at -1.699.

Rules 6 and 7: the asymptotes meet the real axis at

$$\sigma_a = \frac{(0) + (-3.398) - 0}{2} = -1.699$$

at angles of $90°$ and $270°$, that is, $180°$ apart. The asymptotes are therefore the vertical line through -1.699.

The branches leave the poles, meet at -1.699, and then follow the vertical asymptote symmetrically into the upper and lower half-planes, as shown in Figure 11. They never cross the imaginary axis, so the closed-loop system is stable for all positive K_p.

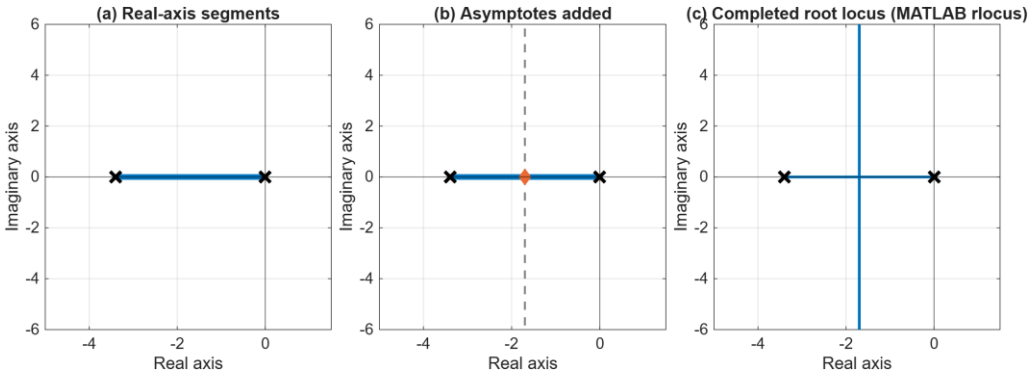


Figure 11: *Construction of the Root Locus for Example 1: (a) Rule 3, (b) Rules 6 and 7, (c) Completed Root Locus*

Using MATLAB to Solve Example 1

MATLAB Script 1 plots the root locus of the DC motor. The root locus is then used to select K_p for the following SPC: a peak overshoot A_p of no more than 5% and a rise time t_r of less than 2 s. A damping ratio of $\zeta = 0.7$ gives approximately 4.6% overshoot (see Appendix A). On the root locus, the closed-loop characteristic equation is $s^2 + 3.398s + 113.7K_p = 0$, so $2\zeta\omega_n = 3.398$ and $\omega_n^2 = 113.7K_p$. With $\zeta = 0.7$:

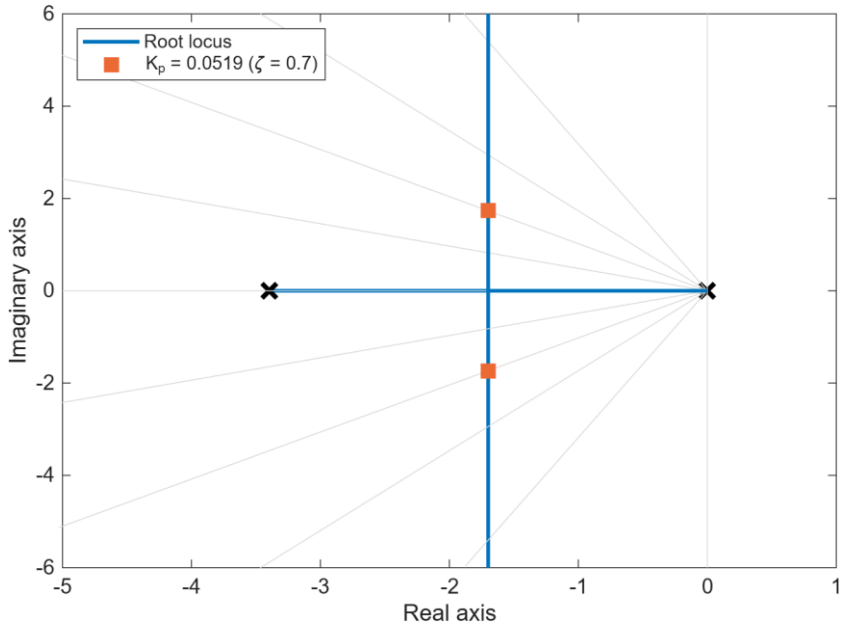
$$\omega_n = \frac{3.398}{2(0.7)} = 2.43 \text{ rad/s}, \qquad K_p = \frac{\omega_n^2}{113.7} = \frac{5.89}{113.7} = 0.0519 \tag{8}$$

In MATLAB, the same value is found by clicking on the root locus where it crosses the $\zeta = 0.7$ line.

MATLAB Script 1: root locus of the DC motor angle control system

```
clear; close all; clc;
%% Root locus
s = tf('s');
G = 113.7/(s^2 + 3.398*s)   % open-loop DC motor transfer function
rlocus(G)
sgrid
grid on
```

Output:



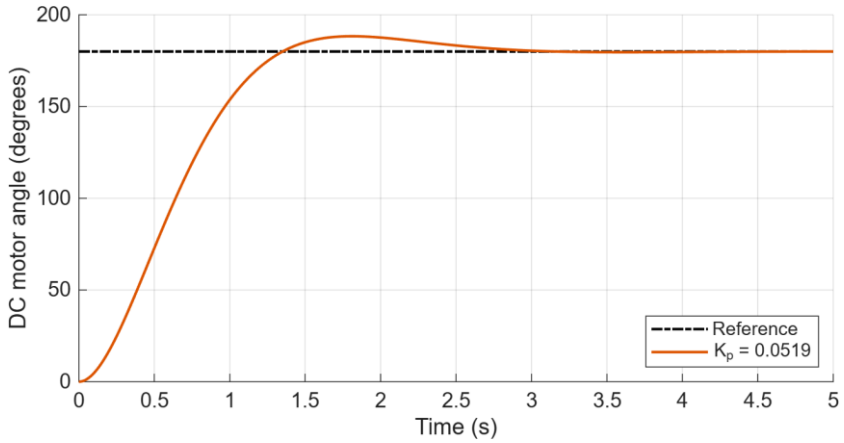
The closed-loop poles for $K_p = 0.0519$ are $s = -1.70 \pm j1.74$, on the $\zeta = 0.7$ line.

MATLAB Script 2: closed-loop step response with the selected proportional control gain

```
clear; close all; clc;
%% Open-loop system
```

```
s = tf('s');
G = 113.7/(s^2 + 3.398*s);
%% Closed-loop control system with selected Kp value
r = 180;      % reference (degrees)
t = 5;        % simulation time (s)
Kp = 0.0519;  % proportional control gain
G_cl = feedback(Kp*G,1);   % negative unity feedback
opt = stepDataOptions('StepAmplitude',r);
[y,t] = step(G_cl,opt,t);
plot(t,y); grid on
xlabel('Time (s)'); ylabel('Angle (degrees)')
stepinfo(G_cl)
```

Output:



The response has a peak overshoot of 4.6% and a rise time of 0.88 s, so it meets the SPC.

Example 2: Soldering robot displacement control (relative degree of 3)

For the open-loop transfer function of the soldering robot, $G(s) = \frac{10}{s(s+10)(0.1s+10)}$, apply the rules of root locus and sketch the root locus on the s-plane.

Solution:

The poles are at 0, −10 and −100 (from $0.1s + 10 = 0$), and there are no zeros.

Rule 2: three poles, so three branches.

Rule 3: the segments between 0 and −10 (one pole to the right) and to the left of −100 (three poles to the right) are on the root locus.

Rule 4: $rd = 3 - 0 = 3$, so all three branches end at infinity.

Rule 5: the branches from 0 and −10 break away between them, at approximately the midpoint, -5. (The exact breakaway point is -4.87, at $K_p = 23.8$.)

Rules 6 and 7: the asymptotes meet the real axis at

$$\sigma_a = \frac{(0) + (-10) + (-100) - 0}{3} = -36.67$$

at angles of $60°$, $180°$ and $300°$, that is, $120°$ apart.

The two branches that break away follow the $\pm 60°$ asymptotes into the right-half plane, while the branch from −100 moves left along the real axis, as shown in Figure 12.

Rule 8: the characteristic equation is $0.1s^3 + 11s^2 + 100s + 10K_p = 0$, or $s^3 + 110s^2 + 1000s + 100K_p = 0$. The Routh array gives $b_1 = (110000 - 100K_p)/110$, so the system is stable for $0 < K_p < 1100$. At $K_p = 1100$ the branches cross the imaginary axis at $s = \pm j\sqrt{1000} = \pm j31.6$.

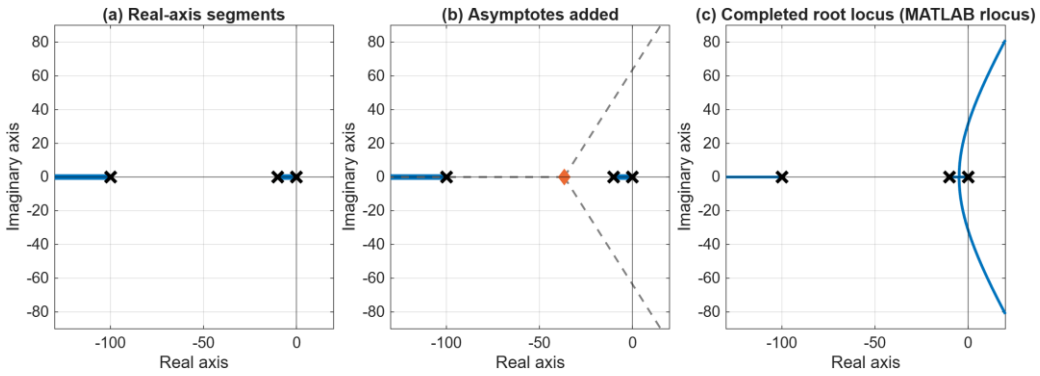


Figure 12: *Construction of the Root Locus for Example 2: (a) Rule 3, (b) Rules 6 and 7, (c) Completed Root Locus*

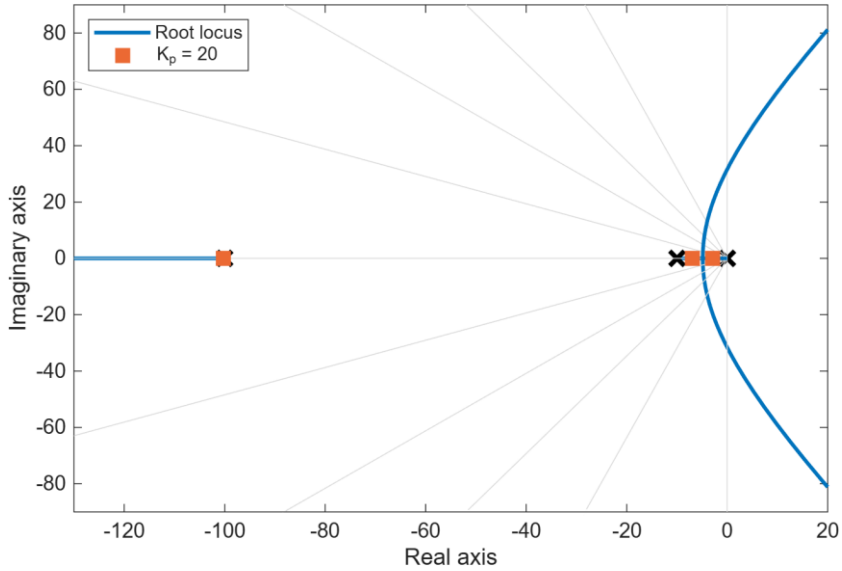
Using MATLAB to Solve Example 2

The SPC for the soldering robot are no overshoot ($A_p = 0$), no steady-state error, and a rise time of less than 2 s. No overshoot requires real closed-loop poles, that is, $\zeta \geq 1$, which on the root locus means a gain below the breakaway value, $K_p \leq 23.8$. A slightly lower gain, $K_p = 20$, is selected to allow for modelling error.

MATLAB Script 3: root locus of the soldering robot displacement control system

```
clear; close all; clc;
%% Root locus
s = tf('s');
G = 10/(s*(s+10)*(0.1*s+10))   % open-loop transfer function
rlocus(G)
axis([-130 20 -90 90])
sgrid
grid on
```

Output:

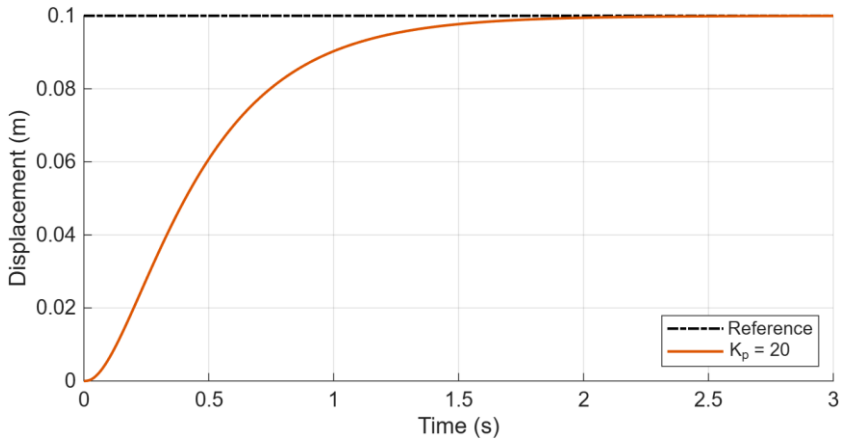


With $K_p = 20$, the closed-loop poles are real, at -2.90, -6.88 and -100.2.

MATLAB Script 4: closed-loop step response with the selected proportional control gain

```
clear; close all; clc;
s = tf('s');
G = 10/(s*(s+10)*(0.1*s+10));
r = 0.1;    % reference (m)
t = 3;      % simulation time (s)
Kp = 20;    % proportional control gain
G_cl = feedback(Kp*G,1);
opt = stepDataOptions('StepAmplitude',r);
[y,t] = step(G_cl,opt,t);
plot(t,y); grid on
xlabel('Time (s)'); ylabel('Displacement (m)')
stepinfo(G_cl)
```

Output:



The response has no overshoot, zero steady-state error and a rise time of 0.86 s, so it meets the SPC.

5.0 Exercises

For each of the closed-loop control systems below: (i) determine the root locus using the rules of root locus, and (ii) check the solution using MATLAB. Each system has negative unity feedback and a proportional control gain K_p; the disturbance does not affect the root locus.

Exercise 1: consider the closed-loop control system in Figure 13.

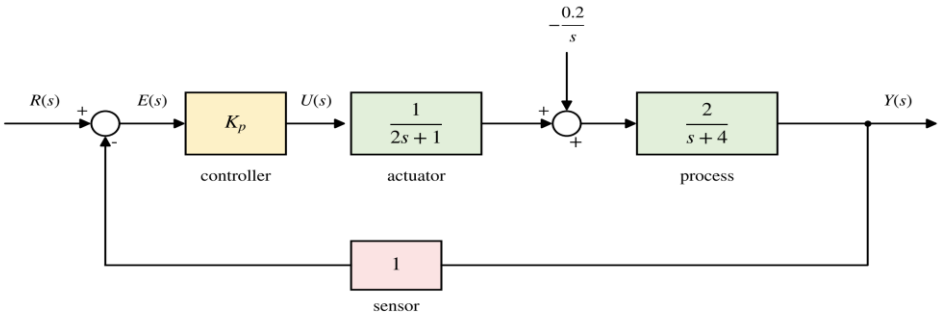


Figure 13: *Closed-Loop Control System for Exercise 1*

Exercise 2: consider the closed-loop control system in Figure 14.

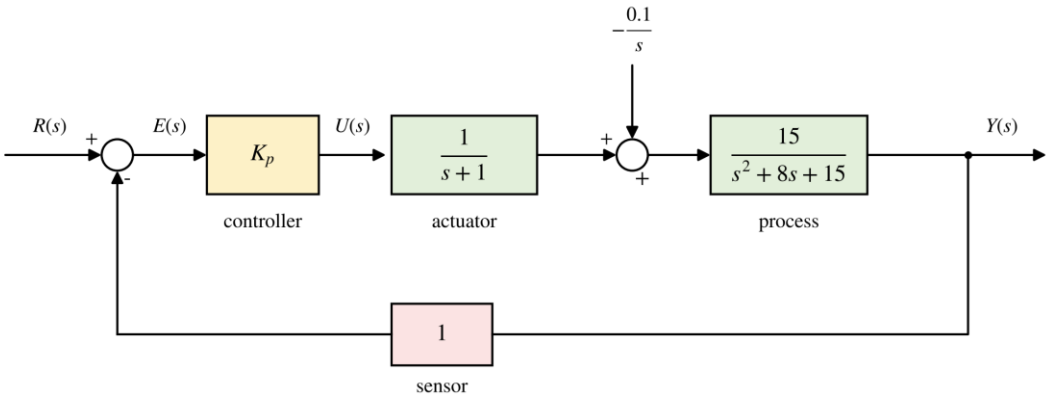


Figure 14: *Closed-Loop Control System for Exercise 2*

Exercise 3: consider the closed-loop control system in Figure 15.

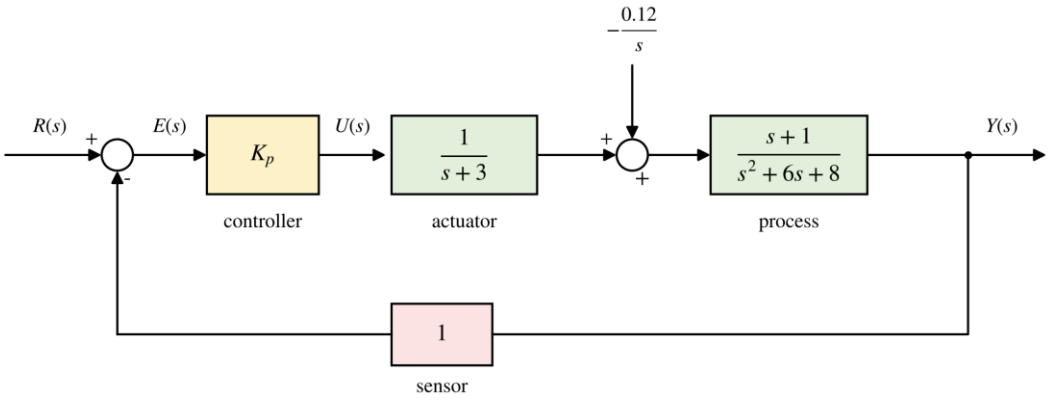


Figure 15: *Closed-Loop Control System for Exercise 3*

Exercise 4: consider the closed-loop control system in Figure 16.

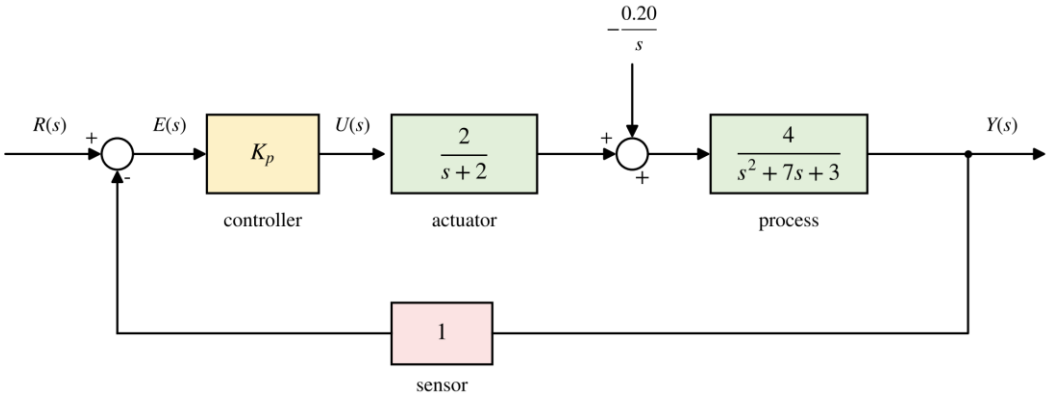


Figure 16: *Closed-Loop Control System for Exercise 4*

Exercise 5: consider the closed-loop control system in Figure 17.

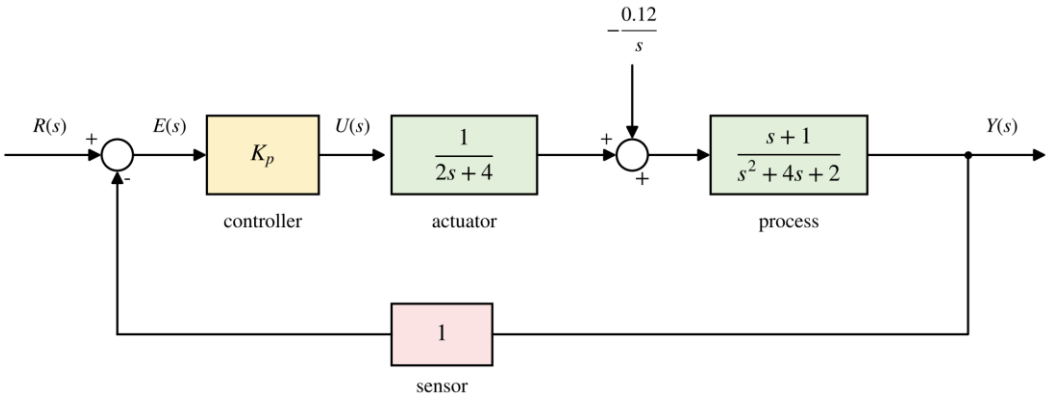


Figure 17: *Closed-Loop Control System for Exercise 5*

Fully worked solutions to the above exercises can be found in **Appendix C**.

6.0 Summary

This Learning Guide showed how to construct the root locus. The root locus follows on from the Routh-Hurwitz stability criterion: it shows the stability of a closed-loop control system and also its potential performance. Its branches are the paths of the closed-loop poles as K_p is increased from zero, where the poles equal the open-loop poles, to infinity.

Three practical examples, with relative degrees of 1, 2 and 3, showed how the poles move. With a relative degree of 1 the pole moves along the real axis; with 2 the poles become a complex pair with a constant real part; with 3 they move towards, and eventually into, the right-half plane. Each point on the root locus corresponds to a damping ratio and natural frequency, so a point can be chosen to meet the SPC, which then form the requirements of the closed-loop control system.

The rules of root locus give a manual method of sketching the root locus, and MATLAB gives precise values at any point on it. When designing a closed-loop control system, MATLAB should be used to confirm the hand sketch and to read off the gain.

Appendix A: Effect of Natural Frequency and Damping Ratio

Figures 18 and 19 show the effect of the natural frequency and the damping ratio on the step response of a second-order system $G(s) = \omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$. A higher natural frequency gives a faster response with the same overshoot; a higher damping ratio gives less overshoot and fewer oscillations.

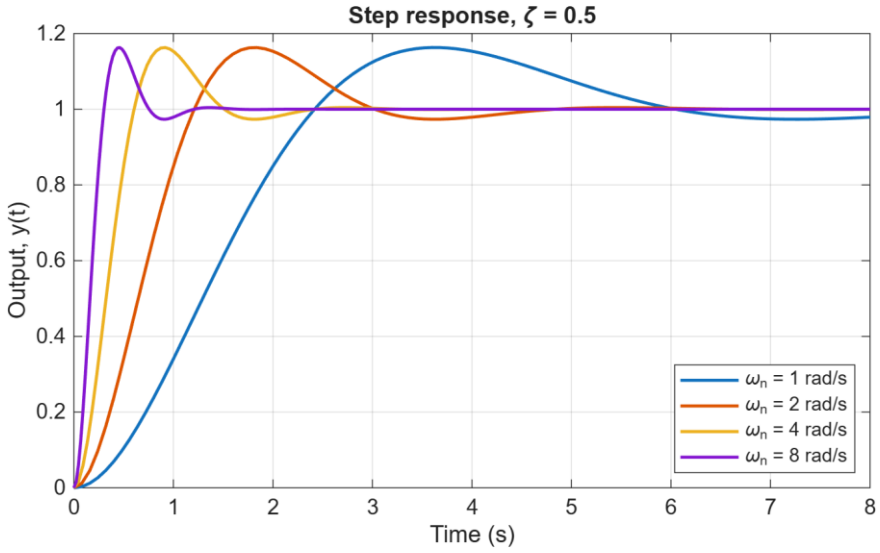


Figure 18: *Effect of the Natural Frequency on the Step Response ($\zeta = 0.5$)*

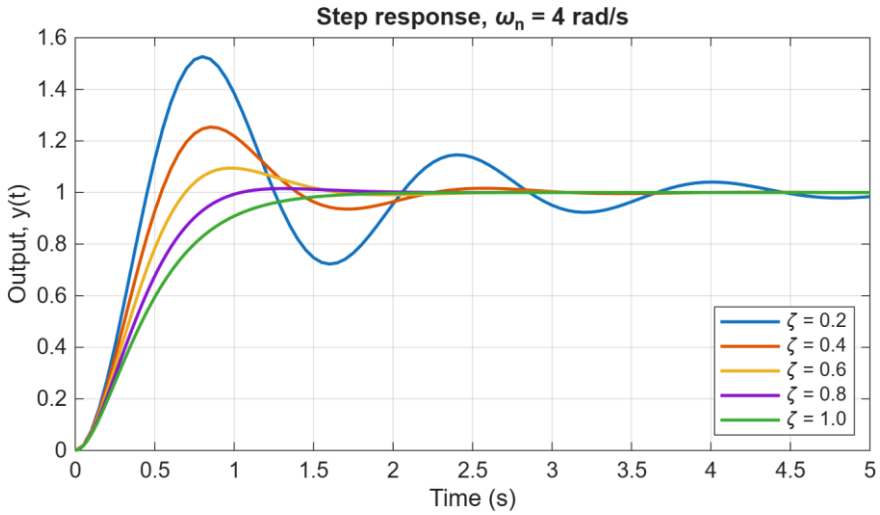


Figure 19: *Effect of the Damping Ratio on the Step Response ($\omega_n = 4$ rad/s)*

MATLAB Script 5: effect of the damping ratio on the step response

```
clear; clc; close all;
wn = 4;                     % natural frequency (rad/s)
for z = 0.2:0.2:1.0         % damping ratio
    sys = tf(wn^2,[1 2*z*wn wn^2]);
    step(sys); hold on
end
grid on
legend('\zeta = 0.2','\zeta = 0.4','\zeta = 0.6','\zeta = 0.8','\zeta = 1.
0')
```

Appendix B: MATLAB Code for the Examples in Section 3.0

MATLAB Script 6: pole locations as K_p is increased (water tank)

```
clear; close all; clc;
col = {'k','b','g','r','m','c'};
for Kp = 0:5
    G_CL = tf(Kp,[2 5+Kp]);
    pzmap(G_CL,col{Kp+1}); hold on
end
grid on
```

MATLAB Script 7: step responses as K_p is increased (water tank)

```
clear; close all; clc;
s = tf('s');
G = 1/(2*s + 5);
figure; hold on; grid on
for Kp = 1:5
    G_cl = feedback(Kp*G,1);
    step(G_cl,2)
end
legend('K_p = 1','K_p = 2','K_p = 3','K_p = 4','K_p = 5')
```

The same scripts apply to the DC motor, with G = 113.7/(s^2 + 3.398*s) and a step amplitude of 180°, and to the soldering robot, with G = 10/(s*(s+10)(0.1*s+10)), Kp = 250:250:1000 and a step amplitude of 0.1 m.

Appendix C: Fully Worked Solutions to Exercises

Exercise 1:

$$G(s) = G_a(s)G_p(s) = \frac{1}{2s+1} \cdot \frac{2}{s+4}$$

Poles: -0.5 and -4. Zeros: none. Two branches (Rule 2).

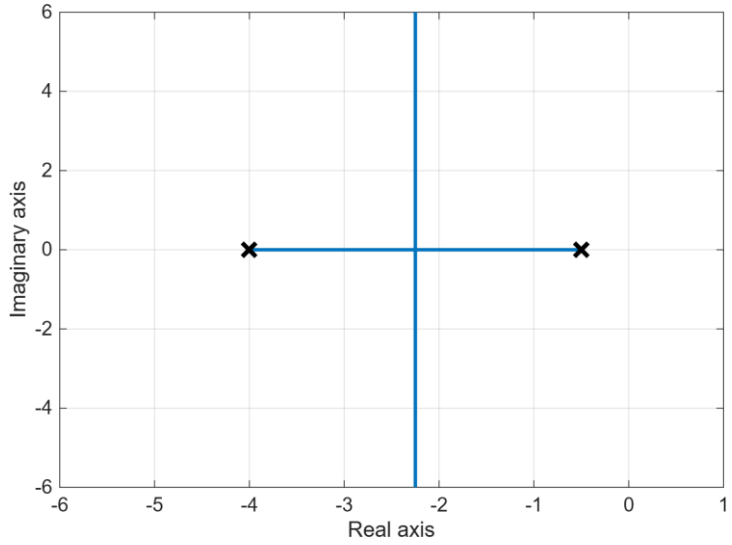
Rule 3: the segment between −0.5 and −4 is on the root locus. **Rule 4:** $rd = 2$, so both branches end at infinity.

Rules 5 to 7: with two poles and no zeros, the breakaway point and the asymptote intercept are both at

$$\sigma_a = \frac{(-0.5) + (-4)}{2} = -2.25$$

and the asymptotes are vertical ($90°$ and $270°$). The root locus never enters the right-half plane, so the system is stable for all positive K_p.

```
s = tf('s');
G = (1/(2*s+1))*(2/(s+4));
rlocus(G); grid on
```



Exercise 2:

$$G(s) = \frac{1}{s+1} \cdot \frac{15}{s^2+8s+15} = \frac{15}{(s+1)(s+3)(s+5)}$$

Poles: -1, -3 and -5. Zeros: none. Three branches.

Rule 3: the segments between −1 and −3, and to the left of −5, are on the root locus. **Rule 4:** $rd = 3$. **Rule 5:** breakaway between −1 and −3, at approximately −2.

Rules 6 and 7:

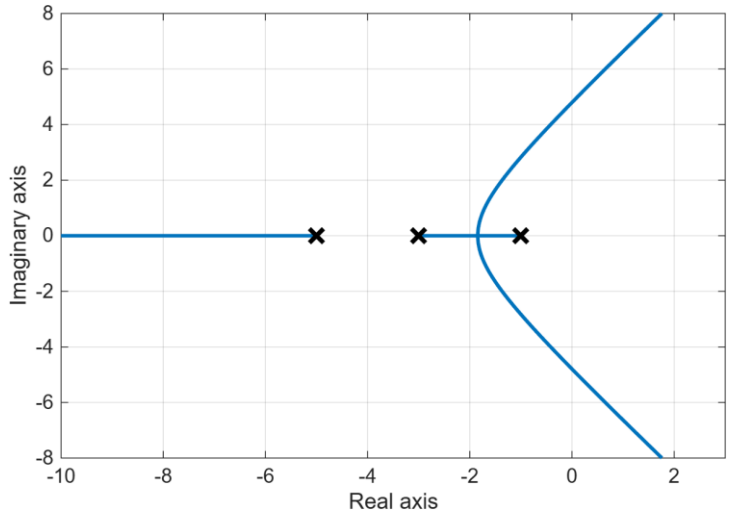
$$\sigma_a = \frac{(-1) + (-3) + (-5)}{3} = -3, \qquad \theta_a = 60°,\ 180°,\ 300°$$

Rule 8: the characteristic equation is $s^3 + 9s^2 + 23s + (15 + 15K_p) = 0$. The Routh array gives

$$b_1 = \frac{(1)(15 + 15K_p) - (9)(23)}{-9} = \frac{192 - 15K_p}{9}$$

so the system is stable for $0 < K_p < 12.8$. At $K_p = 12.8$ the branches cross the imaginary axis at $s = \pm j\sqrt{23} = \pm j4.80$.

```
s = tf('s');
G = 15/((s+1)*(s+3)*(s+5));
rlocus(G); grid on
```



Exercise 3:

$$G(s) = \frac{1}{s+3} \cdot \frac{s+1}{s^2+6s+8} = \frac{s+1}{(s+2)(s+3)(s+4)}$$

Poles: -2, -3 and -4. Zero: -1. Three branches.

Rule 3: the segments between −1 and −2 (one zero to the right) and between −3 and −4 (three to the right) are on the root locus.

Rule 4: $rd = 3 - 1 = 2$. The branch from −2 ends at the zero at −1; the other two end at infinity.

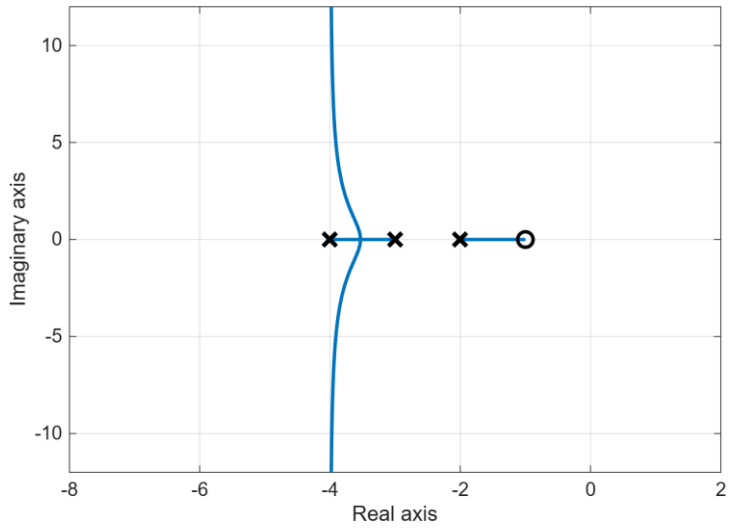
Rule 5: the branches from −3 and −4 break away between them, at approximately −3.5.

Rules 6 and 7:

$$\sigma_a = \frac{(-2) + (-3) + (-4) - (-1)}{2} = -4, \qquad \theta_a = 90°,\ 270°$$

With a relative degree of 2, the system is stable for all positive K_p.

```
s = tf('s');
G = (1/(s+3))*((s+1)/(s^2+6*s+8));
rlocus(G); grid on
```



Exercise 4:

$$G(s) = \frac{2}{s+2} \cdot \frac{4}{s^2+7s+3} = \frac{8}{(s+2)(s^2+7s+3)}$$

Poles: -0.459, -2 and -6.541 (the roots of s^2+7s+3 are $-3.5 \pm \sqrt{9.25}$). Zeros: none. Three branches.

Rule 3: the segments between −0.459 and −2, and to the left of −6.541, are on the root locus. **Rule 4:** $rd = 3$. **Rule 5:** breakaway between −0.459 and −2, at approximately −1.2.

Rules 6 and 7:

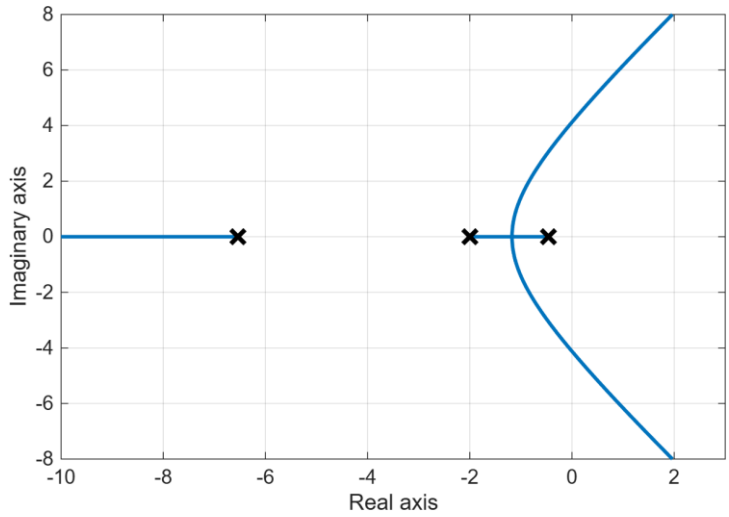
$$\sigma_a = \frac{(-0.459) + (-2) + (-6.541)}{3} = -3, \qquad \theta_a = 60°,\ 180°,\ 300°$$

Rule 8: the characteristic equation is $s^3 + 9s^2 + 17s + \left(6 + 8K_p\right) = 0$, and

$$b_1 = \frac{(1)\left(6 + 8K_p\right) - (9)(17)}{-9} = \frac{147 - 8K_p}{9}$$

so the system is stable for $0 < K_p < 18.4$. At $K_p = 18.4$ the branches cross the imaginary axis at $s = \pm j\sqrt{17} = \pm j4.12$.

```
s = tf('s');
G = (2/(s+2))*(4/(s^2+7*s+3));
rlocus(G); grid on
```



Exercise 5:

$$G(s) = \frac{1}{2s+4} \cdot \frac{s+1}{s^2+4s+2} = \frac{0.5(s+1)}{(s+2)(s^2+4s+2)}$$

Poles: -0.586, -2 and -3.414 (the roots of s^2+4s+2 are $-2 \pm \sqrt{2}$). Zero: -1. Three branches.

Rule 3: the segments between −0.586 and −1, and between −2 and −3.414, are on the root locus.

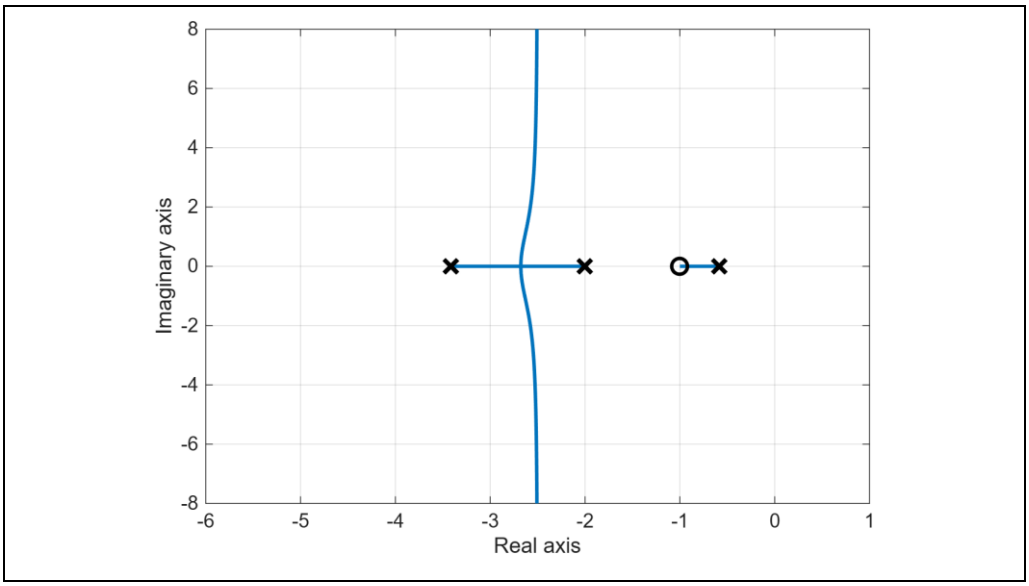
Rule 4: $rd = 2$. The branch from −0.586 ends at the zero at −1; the other two end at infinity. **Rule 5:** breakaway between −2 and −3.414.

Rules 6 and 7:

$$\sigma_a = \frac{(-0.586) + (-2) + (-3.414) - (-1)}{2} = -2.5, \qquad \theta_a = 90°,\ 270°$$

With a relative degree of 2, the system is stable for all positive K_p.

```
s = tf('s');
G = (1/(2*s+4))*((s+1)/(s^2+4*s+2));
rlocus(G); grid on
```



Recommended Citation

If you have used this learning guide in your teaching, learning or research, please cite it using the reference below, adapting the format as required by your preferred referencing style.

Pickering, J. E. (2026). Proportional Control Design using Root Locus. ACE-Lab. Teaching Notes: ACE-Lab Learning Guide, Version 1.0. https://doi.org/10.5281/zenodo.23511338