Topic #15

16.31 Feedback Control

State-Space Systems

- **Full-state Feedback Control**
- How do we change the poles of the state-space system?
- Or, even if we can change the pole locations.
- **Where do we put the poles?**
  - Linear Quadratic Regulator
  - Symmetric Root Locus
- How well does this approach work?

Pole Placement

- So far we have looked at how to pick K to get the dynamics to have some nice properties (*i.e.* stabilize A)

$$\lambda_i(A) \rightsquigarrow \lambda_i(A - BK)$$

- **Classic Question:** where should we put these closed-loop poles?

- Of course we can use the time-domain specifications to locate the dominant poles – roots of:

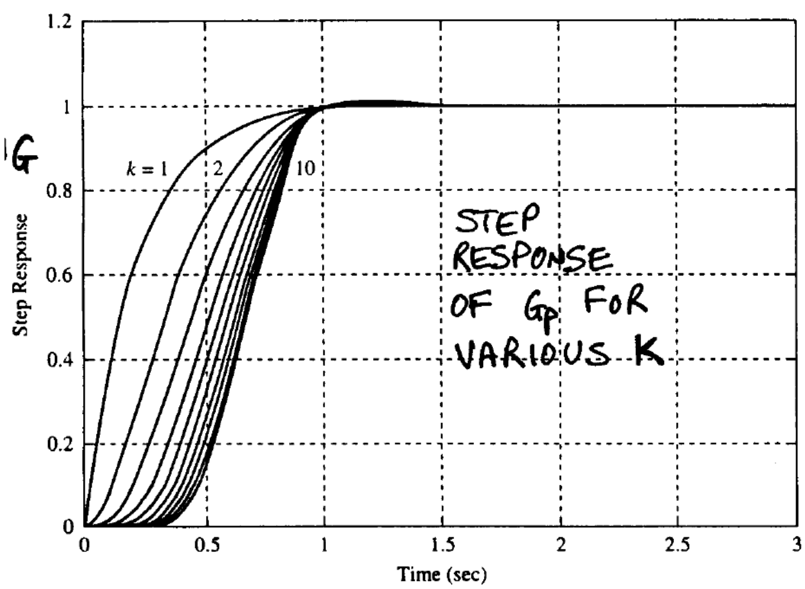
$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

- Then place rest of the poles so they are "much faster" than the dominant behavior. For example:
  - Could keep the same damped frequency w_d and then move the real part to be 2–3 times faster than real part of dominant poles $\zeta\omega_n$

- Just be careful moving the poles too far to the left because it takes a lot of control effort

- Could also choose the closed-loop poles to *mimic a system* that has similar performance to what you would like to achieve:
  - Just set pole locations equal to those of the prototype system.
  - Various options exist

- **Bessel Polynomial Systems** of order $k \to G_p(s) = \frac{1}{B_k(s)}$



Roots of normalized Bessel polynomials corresponding to a settling time of 1 second

k	Pole locations of $B_k(s)$
1	-4.6200
2	$-4.0530 \pm j2.3400$
3	$-5.0093, -3.9668 \pm j3.7845$
4	$-4.0156 \pm j5.0723, -5.5281 \pm j1.6553$
5	$-6.4480, -4.1104 \pm j6.3142, -5.9268 \pm j3.0813$
6	$-4.2169 \pm j7.5300, -6.2613 \pm j4.4018, -7.1205 \pm j1.4540$
7	$-8.0271, -4.3361 \pm j8.7519, -6.5714 \pm j5.6786, -7.6824 \pm j2.8081$
8	$-4.4554 \pm j9.9715, -6.8554 \pm j6.9278, -8.1682 \pm j4.1057, -8.7693 \pm j1.3616$
9	$9.6585, -4.5696 \pm j11.1838, -7.1145 \pm j8.1557, -8.5962 \pm j5.3655, -9.4013 \pm j2.6655$
10	$-4.6835 \pm j-12.4022, -7.3609 \pm j9.3777, -8.9898 \pm j6.6057, -9.9657 \pm j3.9342, -10.4278 \pm j1.3071$

- All scaled to give settling times of 1 second, which you can change to t_s by dividing the poles by t_s.

- Procedure for an n^{th} order system:
  - Determine the desired settling time t_s
  - Find the $k = n$ polynomial from the table.
  - Divide pole locations by t_s
  - Form desired characteristic polynomial $\Phi_d(s)$ and use `acker/place` to determine the feedback gains.
  - Simulate to check performance and control effort.

- **Example:**

$$G(s) = \frac{1}{s(s+4)(s+1)}$$

with

$$A = \begin{bmatrix} -5 & -4 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

so that $n = k = 3$.

  - Want $t_s = 2$ sec. So there are 3 poles at:

$$-5.0093/2 = -2.5047 \quad \text{and}$$
$$(-3.9668 \pm 3.7845i)/2 = -1.9834 \pm 1.8922i$$

  - Use these to form $\Phi_d(s)$ and find the gains using `acker`

- The Bessel approach is fine, but the step response is a bit slow.

- Another approach is to select the poles to match the n^{th} polynomial that was designed to minimize the ITAE "integral of the time multiplied by the absolute value of the error"

$$J_{ITAE} = \int_0^\infty t \; |e(t)| \; dt$$

in response to a step function.

- Both Bessel and ITAE are tabulated in FPE-508.
  - Comparison for $k = 3$ (Given for $\omega_0 = 1$ rad/sec, so slightly different than numbers given on previous page)

$$\phi_d^B = (s + 0.9420)(s + 0.7465 \pm 0.7112i)$$

$$\phi_d^{ITAE} = (s + 0.7081)(s + 0.5210 \pm 1.068i)$$

- So the ITAE poles are not as heavily damped.
  - Some overshoot
  - Faster rise-times.

- Problem with both of these approaches is that they completely ignore the control effort required – the designer must iterate.

Linear Quadratic Regulator

- An alternative approach is to place the pole locations so that the closed-loop (SISO) system optimizes the cost function:

$$J_{LQR} = \int_0^\infty \left[x^T(t)(C^T C)x(t) + r\, u(t)^2 \right] dt$$

Where:

  - $y^T y = x^T(C^T C)x$ {assuming $D = 0$} is called the **State Cost**
  - u^2 is called the **Control Cost**, and
  - r is the **Control Penalty**

  - Simple form of the **Linear Quadratic Regulator** Problem.

- Can show that the optimal control is a linear state feedback:

$$u(t) = -K_{lqr} x(t)$$

  - K_{lqr} found by solving an **Algebraic Riccati Equation** (ARE).

- We will look at the details of this solution procedure later. For now, let's just look at the optimal closed-loop pole locations.

- Consider a SISO system with a *minimal model*

$$\dot{x} = Ax + Bu\ , \quad y = Cx$$

where

$$a(s) = \det(sI - A) \;\text{ and }\; C(sI - A)^{-1}B \equiv \frac{b(s)}{a(s)}$$

- Then[1] with $u(t) = -K_{lqr}x(t)$, closed-loop dynamics are:

$$\det(sI - A + BK_{lqr}) = \prod_{i=1}^{n}(s - p_i)$$

where the $p_i = \{$ the left-hand-plane roots of $\Delta(s)\}$, with

$$\Delta(s) = a(s)a(-s) + r^{-1}b(s)b(-s)$$

- **Use this to find the optimal pole locations, and then use those to find the feedback gains required using `acker`.**

- The pole locations can be found using standard root-locus tools.

$$\Delta(s) = a(s)a(-s) + r^{-1}b(s)b(-s) = 0$$

$$\Rightarrow \quad 1 + r^{-1}G(s)G(-s) = 0$$

  – The plot is symmetric about the real and imaginary axes.

$$\Rightarrow \quad \textbf{Symmetric Root Locus}$$

  – $2n$ poles are plotted as a function of r

  – **The poles we pick are always the n in the LHP.**

[1]Several leaps made here for now. We will come back to this LQR problem later.

LQR Notes

1. The state cost was written using the output $y^T y$, but that does not need to be the case.
   - We are free to define a new system output $z = C_z x$ that is not based on a physical sensor measurement.

$$\Rightarrow \quad J_{LQR} = \int_0^\infty \left[x^T(t)(C_z^T C_z)x(t) + r\; u(t)^2 \right] dt$$

   - Selection of z used to isolate the system states you are most concerned about, and thus would like to be regulated to "zero".

2. Note what happens as $r \rightsquigarrow \infty$ – **high control cost case**

$$a(s)a(-s) + r^{-1}b(s)b(-s) = 0 \quad \Rightarrow \quad \mathbf{a(s)a(-s) = 0}$$

   - So the n closed-loop poles are:
     – Stable roots of the open-loop system (already in the LHP.)
     – **Reflection** about the jω-axis of the unstable open-loop poles.

3. Note what happens as $r \rightsquigarrow 0$ – **low control cost case**

$$a(s)a(-s) + r^{-1}b(s)b(-s) = 0 \quad \Rightarrow \quad \mathbf{b(s)b(-s) = 0}$$

   - Assume order of $b(s)b(-s)$ is $2m < 2n$
   - So the n closed-loop poles go to:
     – The m finite zeros of the system that are in the LHP (or the reflections of the systems zeros in the RHP).
     – The system zeros at infinity (there are $n - m$ of these).

- Note that the poles tending to infinity do so along very specific paths so that they form a **Butterworth Pattern**:
  - At high frequency we can ignore all but the highest powers of s in the expression for $\Delta(s) = 0$

$$\Delta(s) = 0 \quad \rightsquigarrow \quad (-1)^n s^{2n} + r^{-1}(-1)^m (b_o s^m)^2 = 0$$

$$\Rightarrow s^{2(n-m)} = (-1)^{n-m+1}\frac{b_o^2}{r}$$

- The $2(n-m)$ solutions of this expression lie on a circle of radius

$$(b_0^2/r)^{1/2(n-m)}$$

at the intersection of the radial lines with **phase from the negative real axis:**

$$\pm\frac{l\pi}{n-m}\ , \quad l = 0, 1, \ldots, \frac{n-m-1}{2}\ , \qquad \mathbf{(n-m)\ odd}$$

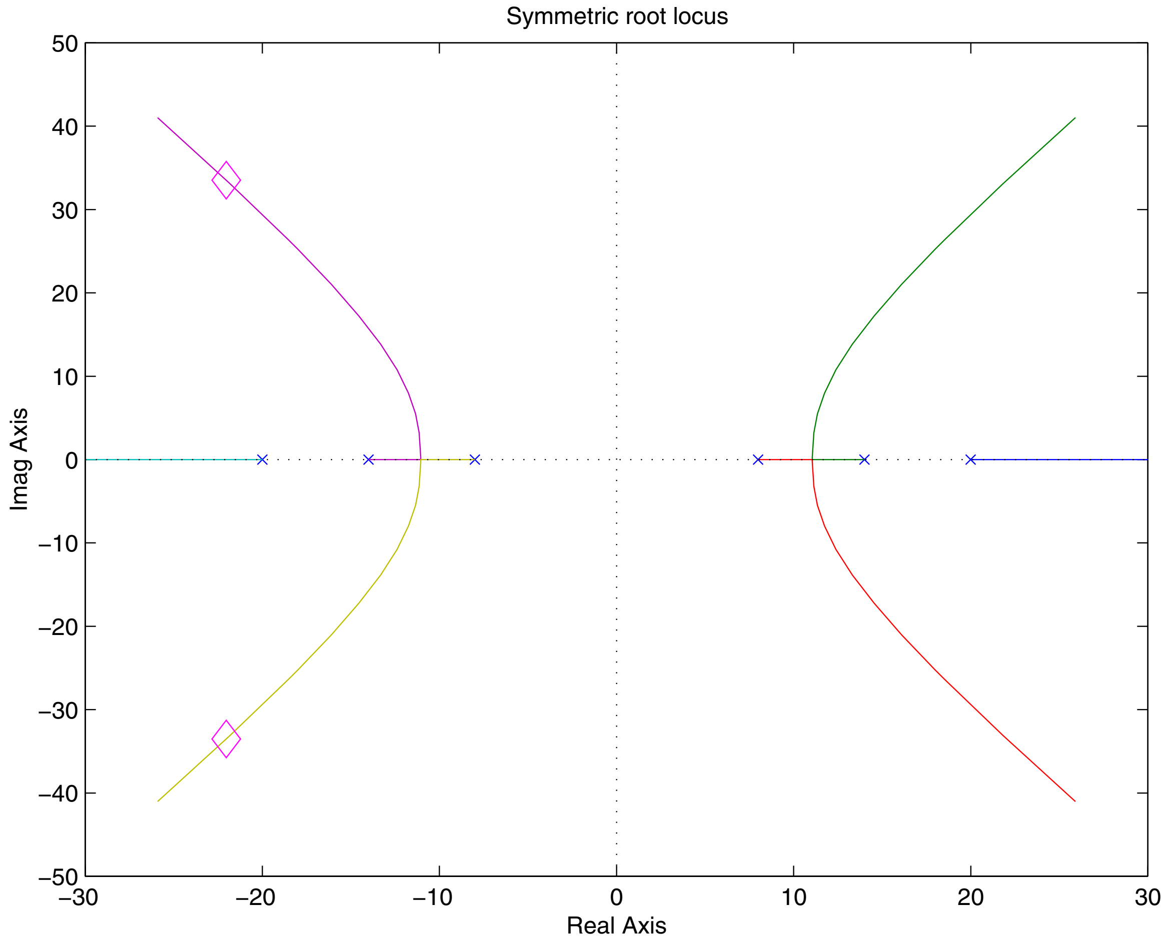
$$\pm\frac{(l+1/2)\pi}{n-m}\ , \quad l = 0, 1, \ldots, \frac{n-m}{2} - 1\ , \qquad \mathbf{(n-m)\ even}$$

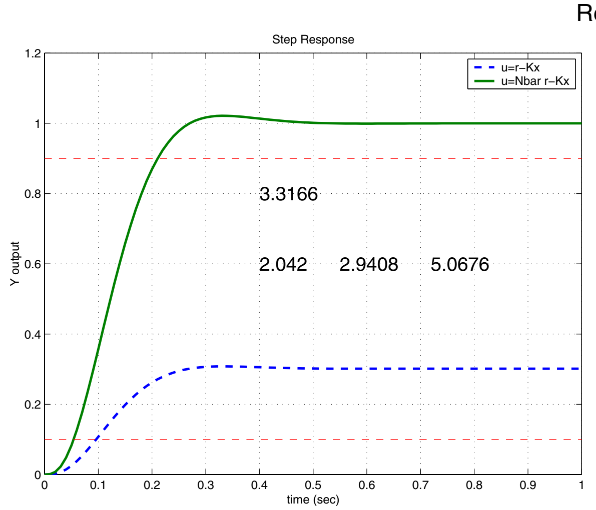
- Examples:

$n-m$	Phase
1	0
2	$\pm\pi/4$
3	$0,\ \pm\pi/3$
4	$\pm\pi/8,\ \pm3\pi/8$

- **Note:** Plot the SRL using the 180^o rules (normal) if $n-m$ is even and the 0^o rules if $n-m$ is odd.

Figure 1: Example #1: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s+8)(s+14)(s+20)}$





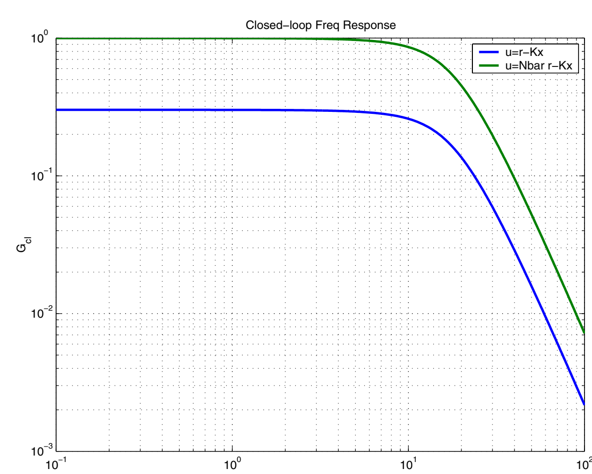


Figure 2: Example #2: $G(s) = \frac{0.94}{s^2 - 0.0297}$

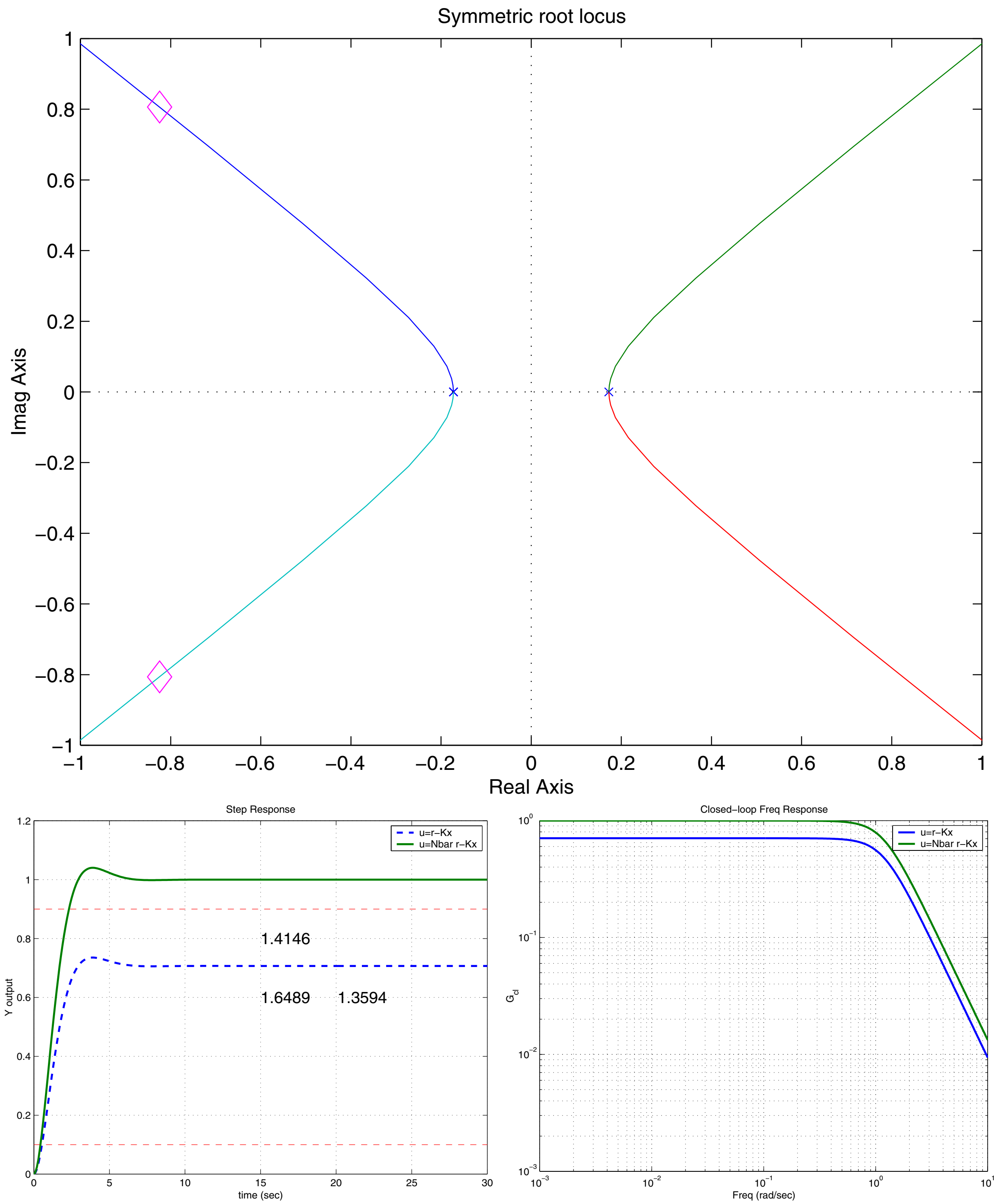
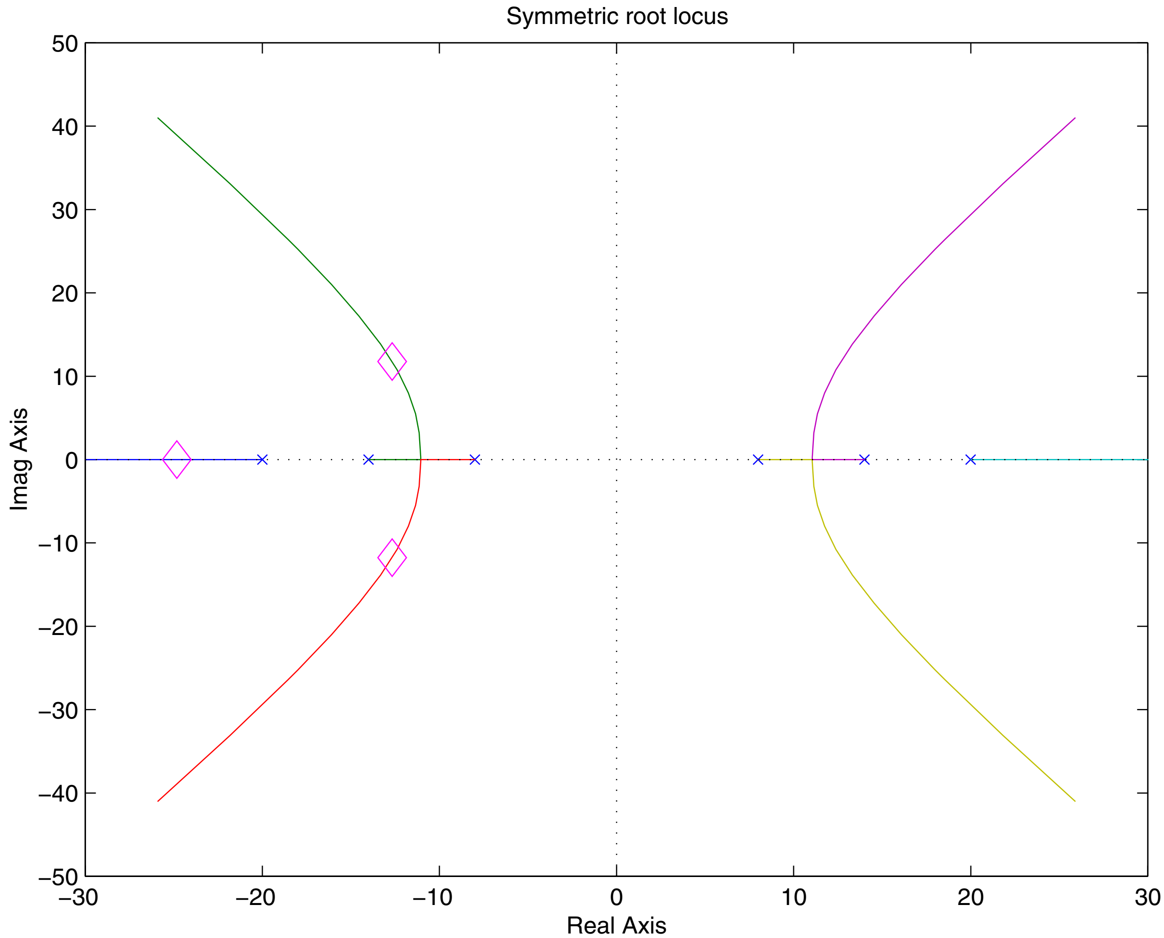
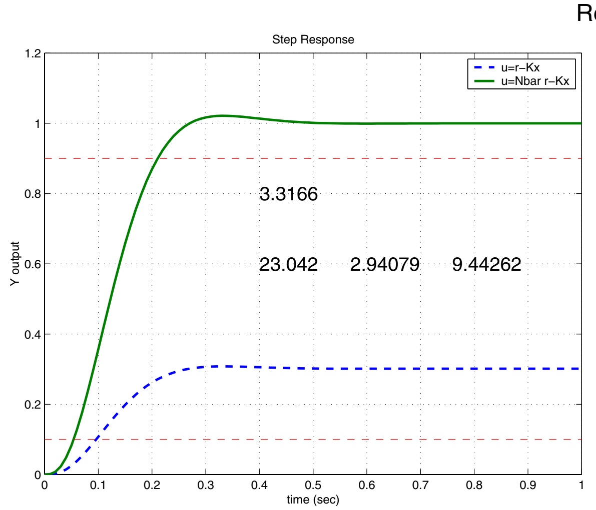


Figure 3: Example #3: $G(s) = \frac{8 \cdot 14 \cdot 20}{(s-8)(s-14)(s-20)}$





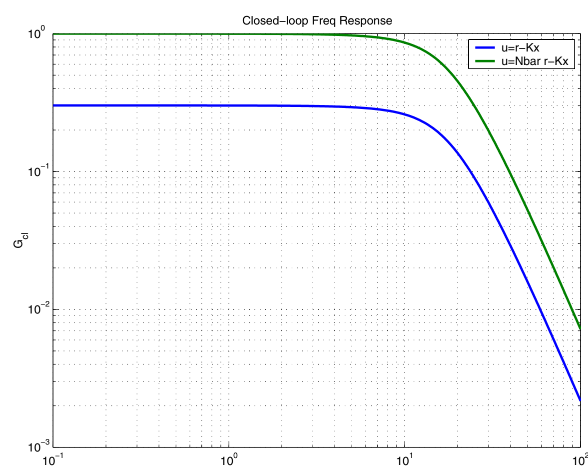
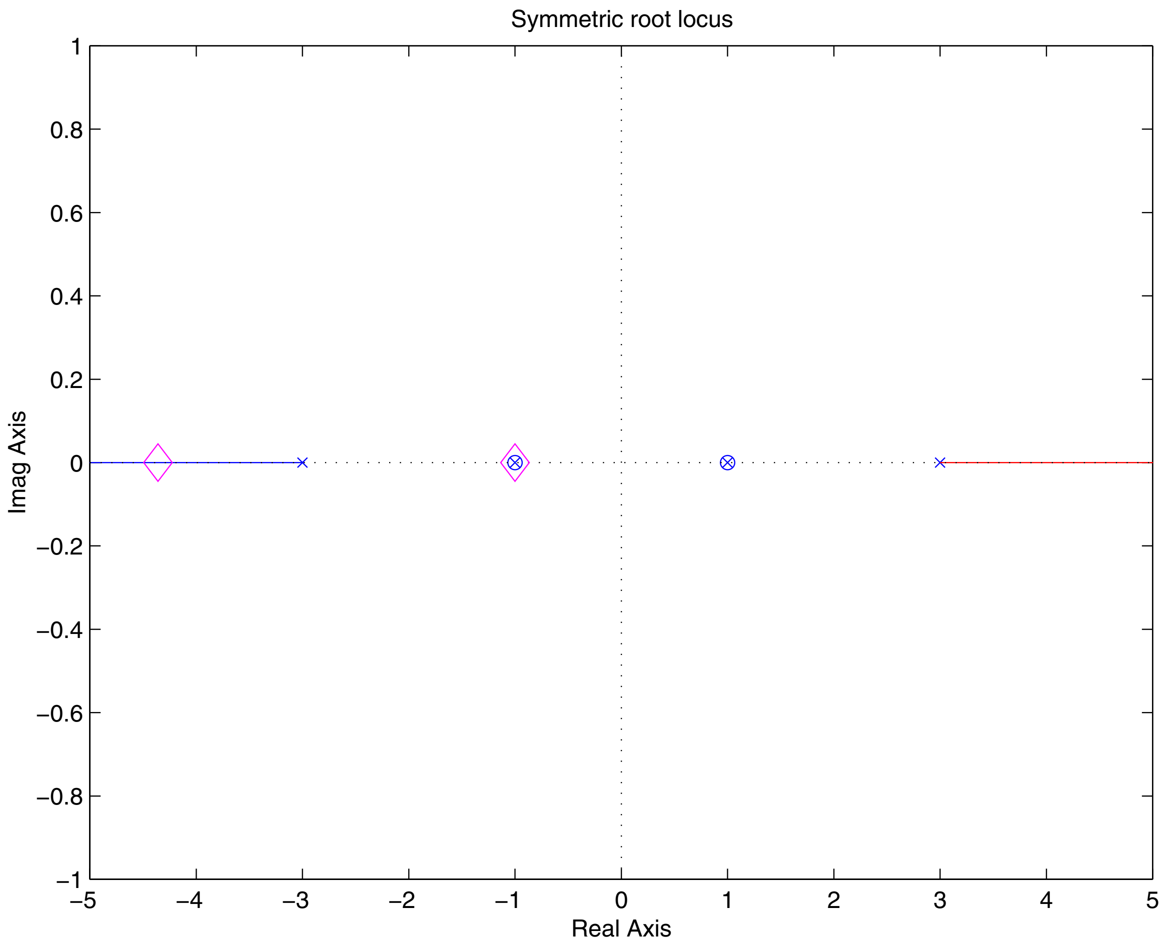
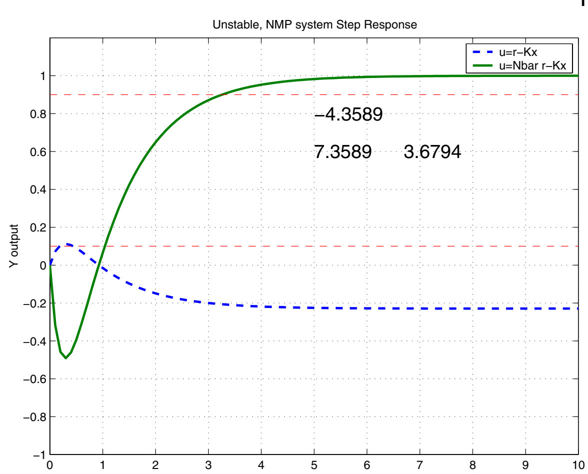


Figure 4: Example #4: $G(s) = \frac{(s-1)}{(s+1)(s-3)}$





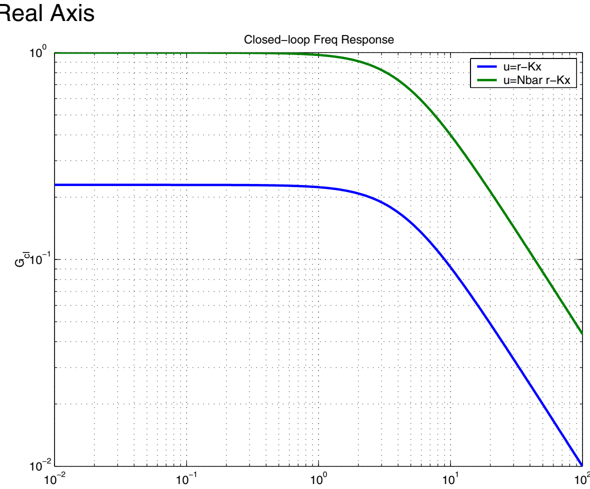
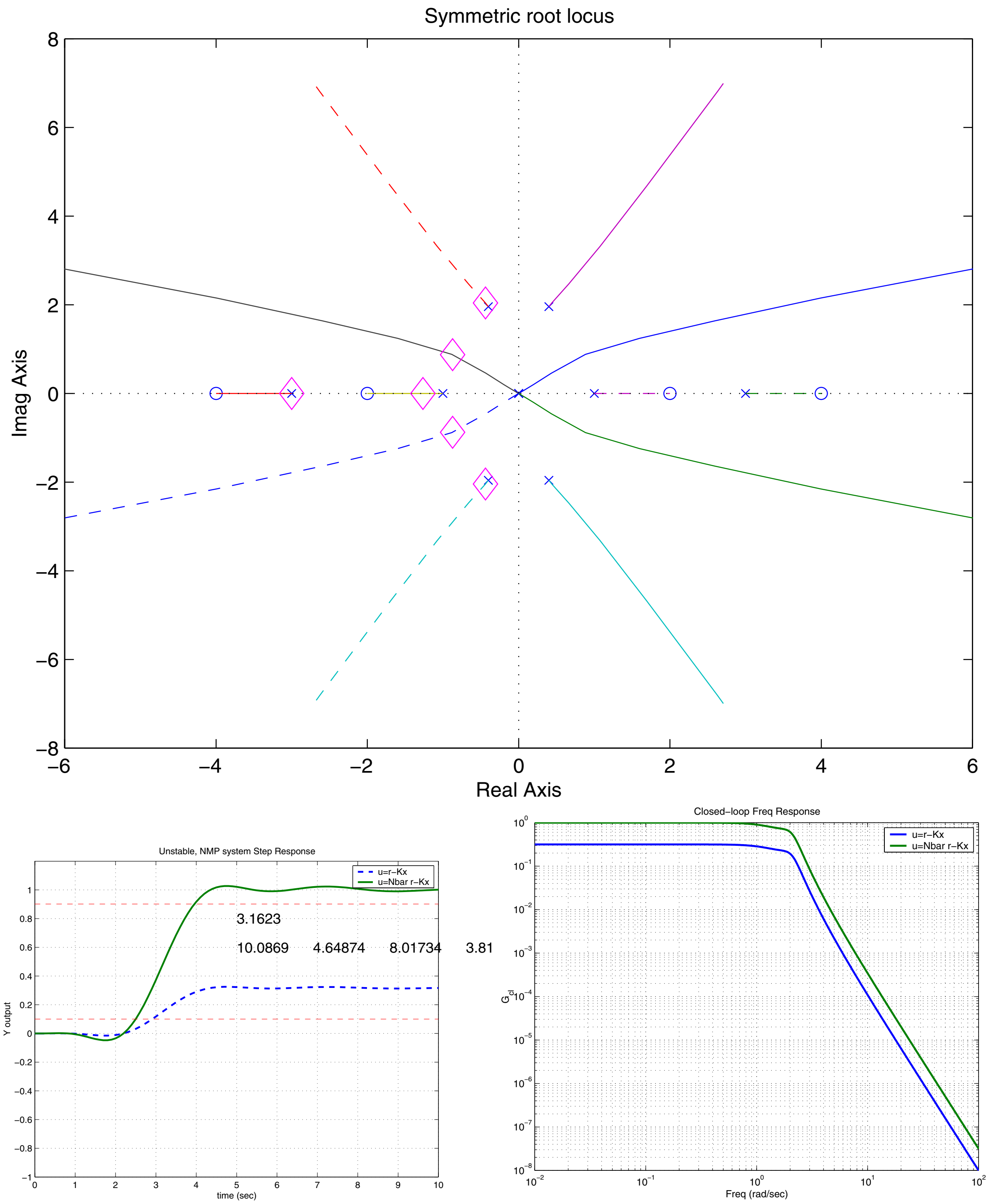


Figure 5: Example #5: $G(s) = \frac{(s-2)(s-4)}{(s-1)(s-3)(s^2+0.8s+4)s^2}$



- As noted previously, we are free to pick the state weighting matrices C_z to penalize the parts of the motion we are most concerned with.

- Simple example – oscillator with $x = [\ p\ ,\quad v\ ]^T$

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

  but we choose two cases for z

$$z = p = \begin{bmatrix} 1 & 0 \end{bmatrix} x \qquad \text{and} \qquad z = v = \begin{bmatrix} 0 & 1 \end{bmatrix} x$$

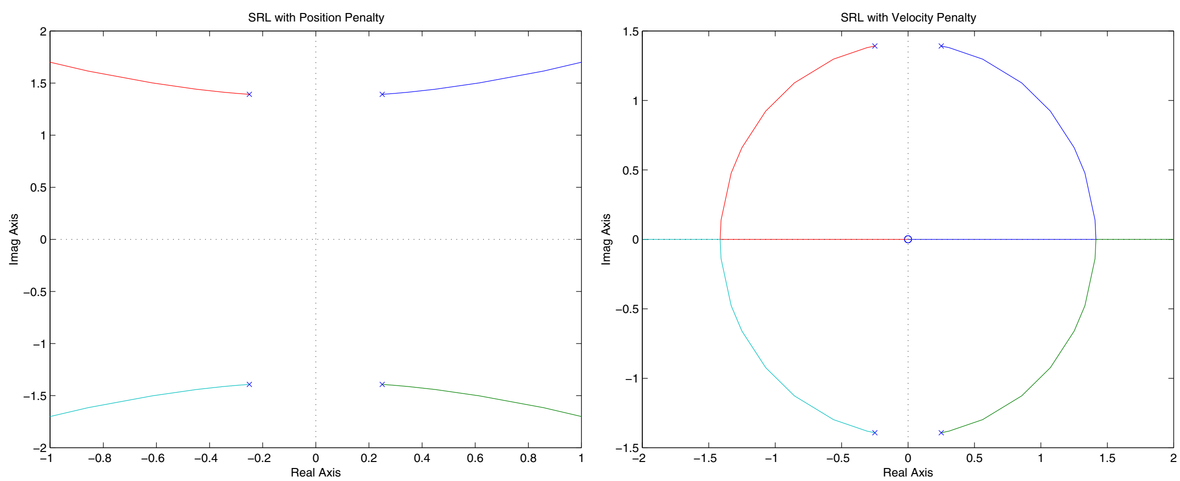


Figure 6: SRL with position (left) and velocity penalties (right)

- Clearly, choosing a different C_z impacts the SRL because it completely changes the zero-structure for the system.

Summary

- Dominant second and prototype design approaches (Bessel and ITAE) place the closed-loop pole locations **with no regard to the amount of control effort required.**
  - Designer must iterate on the selected bandwidth (ω_n) to ensure that the control effort is reasonable.

- LQR/SRL approach selects closed-loop poles that **balance** between system errors and the control effort.
  - Easy design iteration using r – poles move along the SRL.
  - Sometimes difficult to relate the desired transient response to the LQR cost function.

- Nice thing about the LQR approach is that the designer is focused on system performance issues
  - The pole locations are then supplied using the SRL.