16.323 Principles of Optimal Control
Spring 2008

16.323 Lecture 6

Calculus of Variations applied to Optimal Control

$$
\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{a}(\mathbf{x}, \mathbf{u}, t) \\
\dot{\mathbf{p}} &= -H_{\mathbf{x}}^T \\
H_{\mathbf{u}} &= 0
\end{aligned}
$$

Optimal Control Problems

- Are now ready to tackle the optimal control problem
  - Start with simple terminal constraints

$$J = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)dt$$

  with the system dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)$$

  - t_0, $\mathbf{x}(t_0)$ fixed
  - t_f free
  - $\mathbf{x}(t_f)$ are fixed or free by element

- Note that this looks a bit different because we have $\mathbf{u}(t)$ in the integrand, but consider that with a simple substitution, we get

$$\tilde{g}(\mathbf{x}, \dot{\mathbf{x}}, t) \overset{\dot{\mathbf{x}}=\mathbf{a}(\mathbf{x},\mathbf{u},t)}{\longrightarrow} \hat{g}(\mathbf{x}, \mathbf{u}, t)$$

- Note that the differential equation of the dynamics acts as a constraint that we must adjoin using a Lagrange multiplier, as before:

$$J_a = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \left[g(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{p}^T\{\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t) - \dot{\mathbf{x}}\}\right] dt$$

- Find the variation:[10]

$$\begin{aligned}\delta J_a &= h_{\mathbf{x}}\delta \mathbf{x}_f + h_{t_f}\delta t_f + \int_{t_0}^{t_f} \left[g_{\mathbf{x}}\delta\mathbf{x} + g_{\mathbf{u}}\delta\mathbf{u} + (\mathbf{a} - \dot{\mathbf{x}})^T\delta\mathbf{p}(t)\right.\\ &\quad \left. + \mathbf{p}^T(t)\{\mathbf{a}_{\mathbf{x}}\delta\mathbf{x} + \mathbf{a}_{\mathbf{u}}\delta\mathbf{u} - \delta\dot{\mathbf{x}}\}\right] dt + \left[g + \mathbf{p}^T(\mathbf{a} - \dot{\mathbf{x}})\right](t_f)\delta t_f\end{aligned}$$

- Clean this up by defining the **Hamiltonian**: (See 4–4)

$$H(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) = g(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{p}^T(t)\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)$$

[10] Take partials wrt each of the variables that the integrand is a function of.

- Then

$$\begin{aligned}\delta J_a &= h_{\mathbf{x}}\delta\mathbf{x}_f + \left[h_{t_f} + g + \mathbf{p}^T(\mathbf{a}-\dot{\mathbf{x}})\right](t_f)\delta t_f \\ &\quad + \int_{t_0}^{t_f}\left[H_{\mathbf{x}}\delta\mathbf{x} + H_{\mathbf{u}}\delta\mathbf{u} + (\mathbf{a}-\dot{\mathbf{x}})^T\delta\mathbf{p}(t) - \mathbf{p}^T(t)\delta\dot{\mathbf{x}}\right]dt\end{aligned}$$

- To proceed, note that by integrating by parts [11] we get:

$$\begin{aligned}-\int_{t_0}^{t_f}\mathbf{p}^T(t)\delta\dot{\mathbf{x}}dt &= -\int_{t_0}^{t_f}\mathbf{p}^T(t)d\delta\mathbf{x} \\ &= -\mathbf{p}^T\delta\mathbf{x}\Big|_{t_0}^{t_f} + \int_{t_0}^{t_f}\left(\frac{d\mathbf{p}(t)}{dt}\right)^T\delta\mathbf{x}dt \\ &= -\mathbf{p}^T(t_f)\delta\mathbf{x}(t_f) + \int_{t_0}^{t_f}\dot{\mathbf{p}}^T(t)\delta\mathbf{x}dt \\ &= -\mathbf{p}^T(t_f)\left(\delta\mathbf{x}_f - \dot{\mathbf{x}}(t_f)\delta t_f\right) + \int_{t_0}^{t_f}\dot{\mathbf{p}}^T(t)\delta\mathbf{x}dt\end{aligned}$$

- So now can rewrite the variation as:

$$\begin{aligned}\delta J_a &= h_{\mathbf{x}}\delta\mathbf{x}_f + \left[h_{t_f} + g + \mathbf{p}^T(\mathbf{a}-\dot{\mathbf{x}})\right](t_f)\delta t_f \\ &\quad + \int_{t_0}^{t_f}\left[H_{\mathbf{x}}\delta\mathbf{x} + H_{\mathbf{u}}\delta\mathbf{u} + (\mathbf{a}-\dot{\mathbf{x}})^T\delta\mathbf{p}(t)\right]dt - \int_{t_0}^{t_f}\mathbf{p}^T(t)\delta\dot{\mathbf{x}}dt \\ &= \left(h_{\mathbf{x}} - \mathbf{p}^T(t_f)\right)\delta\mathbf{x}_f + \left[h_{t_f} + g + \mathbf{p}^T(\mathbf{a}-\dot{\mathbf{x}}) + \mathbf{p}^T\dot{\mathbf{x}}\right](t_f)\delta t_f \\ &\quad + \int_{t_0}^{t_f}\left[\left(H_{\mathbf{x}} + \dot{\mathbf{p}}^T\right)\delta\mathbf{x} + H_{\mathbf{u}}\delta\mathbf{u} + (\mathbf{a}-\dot{\mathbf{x}})^T\delta\mathbf{p}(t)\right]dt\end{aligned}$$

---

[11] $\int udv \equiv uv - \int vdu$

- So necessary conditions for $\delta J_a = 0$ are that for $t \in [t_0, t_f]$

$$\boxed{\begin{aligned} \dot{\mathbf{x}} &= \mathbf{a}(\mathbf{x}, \mathbf{u}, t) & (\texttt{dim}\ n) \\ \dot{\mathbf{p}} &= -H_{\mathbf{x}}^T & (\texttt{dim}\ n) \\ H_{\mathbf{u}} &= 0 & (\texttt{dim}\ m) \end{aligned}}$$

  – With the boundary condition (lost if t_f is fixed) that

$$h_{t_f} + g + \mathbf{p}^T\mathbf{a} = h_{t_f} + H(t_f) = 0$$

  – Add the boundary constraints that $\mathbf{x}(t_0) = \mathbf{x}_0$ (`dim n`)

  – If $\mathbf{x}_i(t_f)$ is fixed, then $\mathbf{x}_i(t_f) = x_{i_f}$

  – If $\mathbf{x}_i(t_f)$ is free, then $\mathbf{p}_i(t_f) = \dfrac{\partial h}{\partial x_i}(t_f)$ for a total (`dim n`)

- These necessary conditions have $2n$ differential and m algebraic equations with $2n+1$ unknowns (if t_f free), found by imposing the $(2n+1)$ boundary conditions.

- Note the symmetry in the differential equations:

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{a}(\mathbf{x}, \mathbf{u}, t) = \left(\frac{\partial H}{\partial \mathbf{p}}\right)^T \\
\dot{\mathbf{p}} &= -\left(\frac{\partial H}{\partial \mathbf{x}}\right)^T = -\frac{\partial (g + \mathbf{p}^T \mathbf{a})}{\partial \mathbf{x}}^T \\
&= -\left(\frac{\partial \mathbf{a}}{\partial \mathbf{x}}\right)^T \mathbf{p} - \left(\frac{\partial g}{\partial \mathbf{x}}\right)^T
\end{aligned}$$

  – So the dynamics of $\mathbf{p}$, called the **costate**, are **linearized system dynamics** (negative transpose – dual)

$$\left(\frac{\partial \mathbf{a}}{\partial \mathbf{x}}\right) = \begin{bmatrix} \frac{\partial a_1}{\partial x_1} & \cdots & \frac{\partial a_1}{\partial x_n} \\ & \ddots & \\ \frac{\partial a_n}{\partial x_1} & \cdots & \frac{\partial a_n}{\partial x_n} \end{bmatrix}$$

- These necessary conditions are extremely important, and we will be using them for the rest of the term.

Control with General Terminal Conditions

- Can develop similar conditions in the case of more general terminal conditions with t_f free and

$$\mathbf{m}(\mathbf{x}(t_f), t_f) = 0$$

- Follow the same procedure on 6–1 using the insights provided on 5–21 (using the g_a form on 5–20) to form

$$w(\mathbf{x}(t_f), \boldsymbol{\nu}, t_f) = h(\mathbf{x}(t_f), t_f) + \boldsymbol{\nu}^T \mathbf{m}(\mathbf{x}(t_f), t_f)$$

- Work through the math, and get the necessary conditions are

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{a}(\mathbf{x}, \mathbf{u}, t) && (\texttt{dim}\ n) && (6.22) \\ \dot{\mathbf{p}} &= -H_{\mathbf{x}}^T && (\texttt{dim}\ n) && (6.23) \\ H_{\mathbf{u}} &= 0 && (\texttt{dim}\ m) && (6.24) \end{aligned}$$

  – With the boundary condition (lost if t_f fixed)

$$H(t_f) + w_{t_f}(t_f) = 0$$

  – And $\mathbf{m}(\mathbf{x}(t_f), t_f) = 0$, with $\mathbf{x}(t_0)$ and t_0 given.

  – With (since $\mathbf{x}(t_f)$ is not directly given)

$$\mathbf{p}(t_f) = \left[\frac{\partial w}{\partial \mathbf{x}}(t_f)\right]^T$$

- Collapses to form on 6–3 if $\mathbf{m}$ not present – i.e., does not constrain $\mathbf{x}(t_f)$

Example 6–1

- Simple double integrator system starting at $y(0) = 10$, $\dot{y}(0) = 0$, must drive to origin $y(t_f) = \dot{y}(t_f) = 0$ to minimize the cost ($b > 0$)

$$J = \frac{1}{2}\alpha t_f^2 + \frac{1}{2}\int_0^{t_f} bu^2(t)dt$$

- Define the dynamics with $x_1 = y$, $x_2 = \dot{y}$ so that

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bu(t) \qquad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- With $\mathbf{p}(t) = [p_1(t) \;\; p_2(t)]^T$, define the Hamiltonian

$$H = g + \mathbf{p}^T(t)\mathbf{a} = \frac{1}{2}bu^2 + \mathbf{p}^T(t)\left(A\mathbf{x}(t) + Bu(t)\right)$$

- The necessary conditions are then that:

$$\dot{\mathbf{p}} = -H_{\mathbf{x}}^T, \quad \rightarrow \quad \begin{aligned} \dot{p}_1 &= -\frac{\partial H}{\partial x_1} = 0 \rightarrow p_1(t) = c_1 \\ \dot{p}_2 &= -\frac{\partial H}{\partial x_2} = -p_1 \rightarrow p_2(t) = -c_1 t + c_2 \end{aligned}$$

$$H_u = bu + p_2 = 0 \quad \rightarrow \quad u = -\frac{p_2}{b} = -\frac{c_2}{b} + \frac{c_1}{b}t$$

- Now impose the boundary conditions:

$$\begin{aligned} H(t_f) + h_t(t_f) &= \frac{1}{2}bu^2(t_f) + p_1(t_f)x_2(t_f) + p_2(t_f)u(t_f) + \alpha t_f = 0 \\ &= \frac{1}{2}bu^2(t_f) + (-bu(t_f))u(t_f) + \alpha t_f \\ &= -\frac{1}{2}bu^2(t_f) + \alpha t_f = 0 \;\rightarrow\; t_f = \frac{1}{2b\alpha}\left(-c_2 + c_1 t_f\right)^2 \end{aligned}$$

- Now go back to the state equations:

$$\dot{x}_2(t) = -\frac{c_2}{b} + \frac{c_1}{b}t \quad \longrightarrow \quad x_2(t) = c_3 - \frac{c_2}{b}t + \frac{c_1}{2b}t^2$$

and since $x_2(0) = 0$, $c_3 = 0$, and

$$\dot{x}_1(t) = x_2(t) \quad \longrightarrow \quad x_1(t) = c_4 - \frac{c_2}{2b}t^2 + \frac{c_1}{6b}t^3$$

and since $x_1(0) = 10$, $c_4 = 10$

- Now note that

$$\begin{aligned} x_2(t_f) &= -\frac{c_2}{b}t_f + \frac{c_1}{2b}t_f^2 = 0 \\ x_1(t_f) &= 10 - \frac{c_2}{2b}t_f^2 + \frac{c_1}{6b}t_f^3 = 0 \\ &= 10 - \frac{c_2}{6b}t_f^2 = 0 \quad \longrightarrow \quad c_2 = \frac{60b}{t_f^2}, \qquad c_1 = \frac{120b}{t_f^3} \end{aligned}$$

  – But that gives us:

$$t_f = \frac{1}{2b\alpha}\left(-\frac{60b}{t_f^2} + \frac{120b}{t_f^3}t_f\right)^2 = \frac{(60b)^2}{2b\alpha t_f^4}$$

  so that $t_f^5 = 1800b/\alpha$ or $t_f \approx 4.48(b/\alpha)^{1/5}$, which makes sense because t_f goes down as α goes up.

  – Finally, $c_2 = 2.99b^{3/5}\alpha^{2/5}$ and $c_1 = 1.33b^{2/5}\alpha^{3/5}$

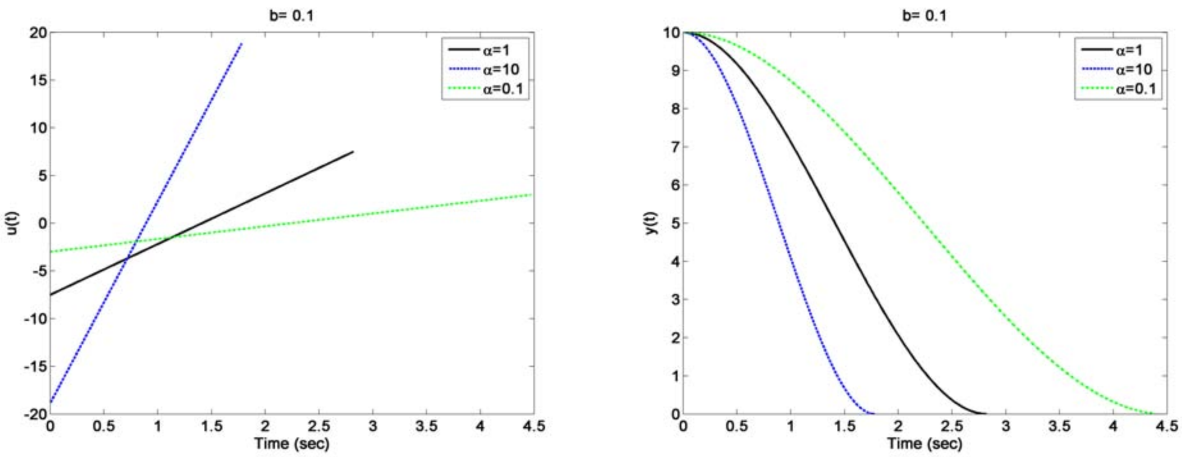


Figure 6.1: Example 6–1

Example 6–1

```
%
% Simple opt example showing impact of weight on t_f
% 16.323 Spring 2008
% Jonathan How
% opt1.m
%
clear all;close all;
set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontWeight','demi')
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontWeight','demi')
%
A=[0 1;0 0];B=[0 1]';C=eye(2);D=zeros(2,1);
G=ss(A,B,C,D);
X0=[10 0]';
b=0.1;

alp=1;
tf=(1800*b/alp)^0.2;
c1=120*b/tf^3;
c2=60*b/tf^2;
time=[0:1e-2:tf];
u=(-c2+c1*time)/b;
[y1,t1]=lsim(G,u,time,X0);

figure(1);clg
plot(time,u,'k-','LineWidth',2);hold on
alp=10;
tf=(1800*b/alp)^0.2;
c1=120*b/tf^3;
c2=60*b/tf^2;
time=[0:1e-2:tf];
u=(-c2+c1*time)/b;
[y2,t2]=lsim(G,u,time,X0);
plot(time,u,'b--','LineWidth',2);

alp=0.10;
tf=(1800*b/alp)^0.2;
c1=120*b/tf^3;
c2=60*b/tf^2;
time=[0:1e-2:tf];
u=(-c2+c1*time)/b;
[y3,t3]=lsim(G,u,time,X0);
plot(time,u,'g-.','LineWidth',2);hold off

legend('\alpha=1','\alpha=10','\alpha=0.1')
xlabel('Time (sec)')
ylabel('u(t)')
title(['b= ',num2str(b)])

figure(2);clg
plot(t1,y1(:,1),'k-','LineWidth',2);
hold on
plot(t2,y2(:,1),'b--','LineWidth',2);
plot(t3,y3(:,1),'g-.','LineWidth',2);
hold off
legend('\alpha=1','\alpha=10','\alpha=0.1')
xlabel('Time (sec)')
ylabel('y(t)')
title(['b= ',num2str(b)])

print -dpng -r300 -f1 opt11.png
print -dpng -r300 -f2 opt12.png
```

LQR Variational Solution

- Deterministic Linear Quadratic Regulator

  Plant:

  $$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B_u(t)\mathbf{u}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0$$
  $$\mathbf{z}(t) = C_z(t)\mathbf{x}(t)$$

  Cost:

$$2J_{LQR} = \int_{t_0}^{t_f} \left[\mathbf{z}^T(t)R_{\text{zz}}(t)\mathbf{z}(t) + \mathbf{u}^T(t)R_{\text{uu}}(t)\mathbf{u}(t)\right]\, dt + \mathbf{x}(t_f)^T P_{t_f}\mathbf{x}(t_f)$$

  – Where $P_{t_f} \geq 0$, $R_{\text{zz}}(t) > 0$ and $R_{\text{uu}}(t) > 0$
  – Define $R_{\text{xx}} = C_z^T R_{\text{zz}} C_z \geq 0$
  – $A(t)$ is a continuous function of time.
  – $B_u(t)$, $C_z(t)$, $R_{\text{zz}}(t)$, $R_{\text{uu}}(t)$ are piecewise continuous functions of time, and all are bounded.

- **Problem Statement:** Find input $u(t)\ \forall t \in [t_0, t_f]$ to min J_{LQR}
  – This is not necessarily specified to be a feedback controller.

- To optimize the cost, we follow the procedure of augmenting the constraints in the problem (the system dynamics) to the cost (integrand) to form the **Hamiltonian**:

  $$H = \frac{1}{2}\left(\mathbf{x}^T(t)R_{\text{xx}}\mathbf{x}(t) + \mathbf{u}^T(t)R_{\text{uu}}\mathbf{u}(t)\right) + \mathbf{p}^T(t)\left(A\mathbf{x}(t) + B_u\mathbf{u}(t)\right)$$

  – $\mathbf{p}(t) \in \mathbb{R}^{n\times 1}$ is called the **Adjoint variable** or **Costate**
  – It is the **Lagrange multiplier** in the problem.

- The necessary conditions (see 6–3) for optimality are that:
  1. $\dot{\mathbf{x}}(t) = \dfrac{\partial H}{\partial \mathbf{p}}^T = A\mathbf{x}(t) + B(t)\mathbf{u}(t)$ with $\mathbf{x}(t_0) = \mathbf{x}_0$
  2. $\dot{\mathbf{p}}(t) = -\dfrac{\partial H}{\partial \mathbf{x}}^T = -R_{\rm xx}\mathbf{x}(t) - A^T\mathbf{p}(t)$ with $\mathbf{p}(t_f) = P_{t_f}\mathbf{x}(t_f)$
  3. $\dfrac{\partial H}{\partial \mathbf{u}} = 0 \Rightarrow R_{\rm uu}\mathbf{u} + B_u^T\mathbf{p}(t) = 0$, so $\mathbf{u}^\star = -R_{\rm uu}^{-1}B_u^T\mathbf{p}(t)$
  4. As before, we can check for a minimum by looking at $\dfrac{\partial^2 H}{\partial \mathbf{u}^2} \geq 0$ (need to check that $R_{\rm uu} \geq 0$)
- Note that $\mathbf{p}(t)$ plays the same role as $J_{\mathbf{x}}^\star(\mathbf{x}(t), t)^T$ in previous solutions to the continuous LQR problem (see 4–8).
  – Main difference is there is no need to guess a solution for $J^\star(\mathbf{x}(t), t)$
- Now have:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}^\star(t) = A\mathbf{x}(t) - B_u R_{\rm uu}^{-1} B_u^T \mathbf{p}(t)$$

which can be combined with equation for the adjoint variable

$$\dot{\mathbf{p}}(t) = -R_{\rm xx}\mathbf{x}(t) - A^T\mathbf{p}(t) = -C_z^T R_{\rm zz} C_z \mathbf{x}(t) - A^T\mathbf{p}(t)$$

$$\Rightarrow \quad \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A & -B_u R_{\rm uu}^{-1} B_u^T \\ -C_z^T R_{\rm zz} C_z & -A^T \end{bmatrix}}_{H} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix}$$

where H is called the **Hamiltonian Matrix**.

  – Matrix describes coupled closed loop dynamics for both $\mathbf{x}$ and $\mathbf{p}$.
  – Dynamics of $\mathbf{x}(t)$ and $\mathbf{p}(t)$ are coupled, but $\mathbf{x}(t)$ known initially and $\mathbf{p}(t)$ known at terminal time, since $\mathbf{p}(t_f) = P_{t_f}\mathbf{x}(t_f)$
  – Two point boundary value problem $\Rightarrow$ typically hard to solve.

- However, in this case, we can introduce a new matrix variable $P(t)$ and show that:
  1. $\mathbf{p}(t) = P(t)\mathbf{x}(t)$
  2. It is relatively easy to find $P(t)$.

- How proceed?
  1. For the $2n$ system

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} A & -B_u R_{\text{uu}}^{-1} B_u^T \\ -C_z^T R_{\text{zz}} C_z & -A^T \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix}$$

define a transition matrix

$$F(t_1, t_0) = \begin{bmatrix} F_{11}(t_1, t_0) & F_{12}(t_1, t_0) \\ F_{21}(t_1, t_0) & F_{22}(t_1, t_0) \end{bmatrix}$$

and use this to relate $\mathbf{x}(t)$ to $\mathbf{x}(t_f)$ and $\mathbf{p}(t_f)$

$$\begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix} = \begin{bmatrix} F_{11}(t, t_f) & F_{12}(t, t_f) \\ F_{21}(t, t_f) & F_{22}(t, t_f) \end{bmatrix} \begin{bmatrix} \mathbf{x}(t_f) \\ \mathbf{p}(t_f) \end{bmatrix}$$

so

$$\begin{aligned} \mathbf{x}(t) &= F_{11}(t, t_f)\mathbf{x}(t_f) + F_{12}(t, t_f)\mathbf{p}(t_f) \\ &= \left[F_{11}(t, t_f) + F_{12}(t, t_f) P_{t_f} \right] \mathbf{x}(t_f) \end{aligned}$$

  2. Now find $\mathbf{p}(t)$ in terms of $\mathbf{x}(t_f)$

$$\mathbf{p}(t) = \left[F_{21}(t, t_f) + F_{22}(t, t_f) P_{t_f} \right] \mathbf{x}(t_f)$$

  3. Eliminate $\mathbf{x}(t_f)$ to get:

$$\begin{aligned} \mathbf{p}(t) &= \left[F_{21}(t, t_f) + F_{22}(t, t_f) P_{t_f} \right] \left[F_{11}(t, t_f) + F_{12}(t, t_f) P_{t_f} \right]^{-1} \mathbf{x}(t) \\ &\triangleq P(t)\mathbf{x}(t) \end{aligned}$$

- Now have $\mathbf{p}(t) = P(t)\mathbf{x}(t)$, must find the equation for $P(t)$

$$\begin{aligned} \dot{\mathbf{p}}(t) &= \dot{P}(t)\mathbf{x}(t) + P(t)\dot{\mathbf{x}}(t) \\ \Rightarrow \quad -C_z^T R_{\rm zz} C_z \mathbf{x}(t) - A^T \mathbf{p}(t) &= \end{aligned}$$

$$\begin{aligned} -\dot{P}(t)\mathbf{x}(t) &= C_z^T R_{\rm zz} C_z \mathbf{x}(t) + A^T \mathbf{p}(t) + P(t)\dot{\mathbf{x}}(t) \\ &= C_z^T R_{\rm zz} C_z \mathbf{x}(t) + A^T \mathbf{p}(t) + P(t)(A\mathbf{x}(t) - B_u R_{\rm uu}^{-1} B_u^T \mathbf{p}(t)) \\ &= (C_z^T R_{\rm zz} C_z + P(t)A)\mathbf{x}(t) + (A^T - P(t)B_u R_{\rm uu}^{-1} B_u^T)\mathbf{p}(t) \\ &= (C_z^T R_{\rm zz} C_z + P(t)A)\mathbf{x}(t) + (A^T - P(t)B_u R_{\rm uu}^{-1} B_u^T)P(t)\mathbf{x}(t) \\ &= \left[A^T P(t) + P(t)A + C_z^T R_{\rm zz} C_z - P(t)B_u R_{\rm uu}^{-1} B_u^T P(t)\right]\mathbf{x}(t) \end{aligned}$$

- This must be true for arbitrary $\mathbf{x}(t)$, so $P(t)$ must satisfy

$$-\dot{P}(t) = A^T P(t) + P(t)A + C_z^T R_{\rm zz} C_z - P(t)B_u R_{\rm uu}^{-1} B_u^T P(t)$$

  - Which, of course, is the matrix differential **Riccati Equation**.
  - Optimal value of $P(t)$ is found by solving this equation *backwards* in time from t_f with $P(t_f) = P_{t_f}$

- The control gains are then

$$u_{\rm opt} \;=\; -R_{\rm uu}^{-1}B_u^T\mathbf{p}(t) = -R_{\rm uu}^{-1}B_u^T P(t)\mathbf{x}(t) = -K(t)\mathbf{x}(t)$$

- **Optimal control inputs can in fact be computed using linear feedback on the full system state**
  – Find optimal steady state feedback gains $\mathbf{u}(t) = -K\mathbf{x}(t)$ using

$$K = \texttt{lqr}(A, B, C_z^T R_{\rm zz} C_z, R_{\rm uu})$$

- **Key point:** This controller works equally well for MISO and MIMO regulator designs.

Alternate Derivation of DRE

- On 6-10 we showed that:

$$P(t) = \left[F_{21}(t,t_f) + F_{22}(t,t_f)P_{t_f}\right]\left[F_{11}(t,t_f) + F_{12}(t,t_f)P_{t_f}\right]^{-1}$$

- To find the Riccati equation, note that

$$\frac{d}{dt}M^{-1}(t) = -M^{-1}(t)\dot{M}(t)M^{-1}(t)$$

which gives

$$\begin{aligned}\dot{P}(t) = & \left[\dot{F}_{21}(t,t_f) + \dot{F}_{22}(t,t_f)P_{t_f}\right]\left[F_{11}(t,t_f) + F_{12}(t,t_f)P_{t_f}\right]^{-1} \\ & - \left[F_{21}(t,t_f) + F_{22}(t,t_f)P_{t_f}\right]\left[F_{11}(t,t_f) + F_{12}(t,t_f)P_{t_f}\right]^{-1} \cdot \\ & \left[\dot{F}_{11}(t,t_f) + \dot{F}_{12}(t,t_f)P_{t_f}\right]\left[F_{11}(t,t_f) + F_{12}(t,t_f)P_{t_f}\right]^{-1}\end{aligned}$$

- Since F is the transition matrix [12] for the system (see 6–10), then

$$\frac{d}{dt}F(t,t_f) = HF(t,t_f)$$

$$\begin{bmatrix} \dot{F}_{11} & \dot{F}_{12} \\ \dot{F}_{21} & \dot{F}_{22} \end{bmatrix}(t,t_f) = \begin{bmatrix} A & -B_u R_{\text{uu}}^{-1} B_u^T \\ -R_{\text{xx}} & -A^T \end{bmatrix}(t,t_f) \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}(t,t_f)$$

[12] Consider homogeneous system $\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t)$ with initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. The general solution to this differential equation is given by $\mathbf{x}(t) = \Phi(t,t_0)\mathbf{x}(t_0)$ where $\Phi(t_1,t_1) = I$. Can show the following properties of the state transition matrix Φ:

1. $\Phi(t_2,t_0) = \Phi(t_2,t_1)\Phi(t_1,t_0)$, regardless of the order of the t_i
2. $\Phi(t,\tau) = \Phi(\tau,t)^{-1}$
3. $\frac{d}{dt}\Phi(t,t_0) = A(t)\Phi(t,t_0)$

- Now substitute and re-arrange:

$$
\begin{aligned}
\dot{P} &= \left\{[\dot{F}_{21} + \dot{F}_{22}P_{t_f}] - P[\dot{F}_{11} + \dot{F}_{12}P_{t_f}]\right\}[F_{11} + F_{12}P_{t_f}]^{-1} \\
\dot{F}_{11} &= AF_{11} - B_u R_{\text{uu}}^{-1} B_u^T F_{21} \\
\dot{F}_{12} &= AF_{12} - B_u R_{\text{uu}}^{-1} B_u^T F_{22} \\
\dot{F}_{21} &= -R_{\text{xx}}F_{11} - A^T F_{21} \\
\dot{F}_{22} &= -R_{\text{xx}}F_{12} - A^T F_{22}
\end{aligned}
$$

$$
\begin{aligned}
\dot{P} &= \left\{\left(-R_{\text{xx}}F_{11} - A^T F_{21} + (-R_{\text{xx}}F_{12} - A^T F_{22})P_{t_f}\right)\right. \\
&\left.-P\left(AF_{11} - B_u R_{\text{uu}}^{-1} B_u^T F_{21} + (AF_{12} - B_u R_{\text{uu}}^{-1} B_u^T F_{22})P_{t_f}\right)\right\}[F_{11} + F_{12}P_{t_f}]^{-1}
\end{aligned}
$$

- There are four terms:

$$-R_{\text{xx}}(F_{11} + F_{12}P_{t_f})[F_{11} + F_{12}P_{t_f}]^{-1} = -R_{\text{xx}}$$

$$-A^T(F_{21} + F_{22}P_{t_f})[F_{11} + F_{12}P_{t_f}]^{-1} = -A^T P$$

$$-PA(F_{11} + F_{12}P_{t_f})[F_{11} + F_{12}P_{t_f}]^{-1} = -PA$$

$$PB_u R_{\text{uu}}^{-1} B_u^T(F_{21} + F_{22}P_{t_f})[F_{11} + F_{12}P_{t_f}]^{-1} = PB_u R_{\text{uu}}^{-1} B_u^T P$$

- Which, as expected, gives that

$$-\dot{P} = A^T P + PA + R_{\text{xx}} - PB_u R_{\text{uu}}^{-1} B_u^T P$$

CARE Solution Algorithm

- Recall from (6–10) that

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} A & -B_u R_{\rm uu}^{-1} B_u^T \\ -C_z^T R_{\rm zz} C_z & -A^T \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix}$$

- Assuming that the eigenvalues of H are unique, the Hamiltonian can be diagonalized into the form:

$$\begin{bmatrix} \dot{\mathbf{z}}_1(t) \\ \dot{\mathbf{z}}_2(t) \end{bmatrix} = \begin{bmatrix} -\Lambda & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} \mathbf{z}_1(t) \\ \mathbf{z}_2(t) \end{bmatrix}$$

  where diagonal matrix Λ is comprised of RHP eigenvalues of H.

- A similarity transformation exists between the states $\mathbf{z}_1, \mathbf{z}_2$ and $\mathbf{x}, \mathbf{p}$:

$$\begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix} = \Psi \begin{bmatrix} \mathbf{z}_1(t) \\ \mathbf{z}_2(t) \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{z}_1(t) \\ \mathbf{z}_2(t) \end{bmatrix} = \Psi^{-1} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix}$$

  where

$$\Psi = \left[\begin{array}{c|c} \Psi_{11} & \Psi_{12} \\ \hline \Psi_{21} & \Psi_{22} \end{array}\right] \text{ and } \Psi^{-1} = \left[\begin{array}{c|c} (\Psi^{-1})_{11} & (\Psi^{-1})_{12} \\ \hline (\Psi^{-1})_{21} & (\Psi^{-1})_{22} \end{array}\right]$$

  and the columns of Ψ are the eigenvectors of H.

- Solving for $\mathbf{z}_2(t)$ gives

$$\begin{aligned} \mathbf{z}_2(t) = e^{\Lambda t}\mathbf{z}_2(0) &= [(\Psi^{-1})_{21}\mathbf{x}(t) + (\Psi^{-1})_{22}\mathbf{p}(t)] \\ &= [(\Psi^{-1})_{21} + (\Psi^{-1})_{22}P(t)]\mathbf{x}(t) \end{aligned}$$

  – For the cost to be finite, need $\lim_{t\to\infty} \mathbf{x}(t) = 0$, so can show that

$$\lim_{t\to\infty} \mathbf{z}_2(t) = 0$$

  – But given that the Λ dynamics in the RHP, this can only be true if $\mathbf{z}_2(0) = 0$, which means that $\mathbf{z}_2(t) = 0\ \forall t$

- With this fact, note that

$$\begin{aligned}\mathbf{x}(t) &= \Psi_{11}\mathbf{z}_1(t)\\ \mathbf{p}(t) &= \Psi_{21}\mathbf{z}_1(t)\end{aligned}$$

which can be combined to give:

$$\mathbf{p}(t) = \Psi_{21}(\Psi_{11})^{-1}\mathbf{x}(t) \equiv P_{ss}\mathbf{x}(t)$$

- Summary of solution algorithm:
  - Find the eigenvalues and eigenvectors of H
  - Select the n eigenvectors associated with the n eigenvalues in the LHP.
  - Form Ψ_{11} and Ψ_{21}.
  - Compute the steady state solution of the Riccati equation using

$$P_{ss} = \Psi_{21}(\Psi_{11})^{-1}$$

```
% alternative calc of Riccati solution
H=[A -B*inv(Ruu)*B' ; -Rxx -A'];
[V,D]=eig(H); % check order of eigenvalues
Psi11=V(1:2,1:2);
Psi21=V(3:4,1:2);
Ptest=Psi21*inv(Psi11);
```

Optimal Cost

- Showed in earlier derivations that the optimal cost-to-go from the initial (or any state) is of the form

$$J = \frac{1}{2}\mathbf{x}^T(t_0)P(t_0)\mathbf{x}(t_0)$$

  – Relatively clean way to show it for this derivation as well.

- Start with the standard cost and add zero ($A\mathbf{x} + B_u\mathbf{u} - \dot{\mathbf{x}} = 0$)

$$\begin{aligned} J_{LQR} &= \frac{1}{2}\int_{t_0}^{t_f} \left[\mathbf{x}^T R_{\text{xx}}\mathbf{x} + \mathbf{u}^T R_{\text{uu}}\mathbf{u} + \mathbf{p}^T(A\mathbf{x} + B_u\mathbf{u} - \dot{\mathbf{x}})\right] dt \\ &\quad + \frac{1}{2}\mathbf{x}(t_f)^T P_{t_f}\mathbf{x}(t_f) \end{aligned}$$

- Now use the results of the necessary conditions to get:

$$\begin{aligned} \dot{\mathbf{p}} = -H_{\mathbf{x}}^T \quad &\Rightarrow \mathbf{p}^T A = -\dot{\mathbf{p}}^T - \mathbf{x}^T R_{\text{xx}} \\ H_{\mathbf{u}} = 0 \quad &\Rightarrow \mathbf{p}^T B_u = -\mathbf{u}^T R_{\text{uu}} \end{aligned}$$

  with $\mathbf{p}(t_f) = P_{t_f}\mathbf{x}(t_f)$

- Substitute these terms to get

$$\begin{aligned} J_{LQR} &= \frac{1}{2}\mathbf{x}(t_f)^T P_{t_f}\mathbf{x}(t_f) - \frac{1}{2}\int_{t_0}^{t_f} \left[\dot{\mathbf{p}}^T\mathbf{x} + \mathbf{p}^T\dot{\mathbf{x}}\right] dt \\ &= \frac{1}{2}\mathbf{x}(t_f)^T P_{t_f}\mathbf{x}(t_f) - \frac{1}{2}\int_{t_0}^{t_f} \left[\frac{d}{dt}(\mathbf{p}^T\mathbf{x})\right] dt \\ &= \frac{1}{2}\mathbf{x}(t_f)^T P_{t_f}\mathbf{x}(t_f) - \frac{1}{2}\left[\mathbf{p}^T(t_f)\mathbf{x}(t_f) - \mathbf{p}^T(t_0)\mathbf{x}(t_0)\right] \\ &= \frac{1}{2}\mathbf{x}(t_f)^T P_{t_f}\mathbf{x}(t_f) - \frac{1}{2}\left[\mathbf{x}^T(t_f)P_{t_f}\mathbf{x}(t_f) - \mathbf{x}^T(t_0)P(t_0)\mathbf{x}(t_0)\right] \\ &= \frac{1}{2}\mathbf{x}^T(t_0)P(t_0)\mathbf{x}(t_0) \end{aligned}$$

Pole Locations

- The closed-loop dynamics couple $\mathbf{x}(t)$ and $\mathbf{p}(t)$ and are given by

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} A & -B_u R_{\rm uu}^{-1} B_u^T \\ -C_z^T R_{\rm zz} C_z & -A^T \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix}$$

with the appropriate boundary conditions.

- OK, so where are the closed-loop poles of the system?
  - Answer: must be eigenvalues of Hamiltonian matrix for the system:

$$H \triangleq \begin{bmatrix} A & -B_u R_{\rm uu}^{-1} B_u^T \\ -C_z^T R_{\rm zz} C_z & -A^T \end{bmatrix}$$

so we must solve $\det(sI - H) = 0$.

- **Key point:** For a SISO system, we can relate the closed-loop poles to a **Symmetric Root Locus** (SRL) for the transfer function

$$G_{zu}(s) = C_z(sI - A)^{-1}B_u = \frac{N(s)}{D(s)}$$

  - Poles and zeros of $G_{zu}(s)$ play an integral role in determining SRL
  - Note $G_{zu}(s)$ is the transfer function from control inputs to performance variable.

- In fact, the closed-loop poles are given by the LHP roots of

$$\Delta(s) = D(s)D(-s) + \frac{R_{\rm zz}}{R_{\rm uu}} N(s)N(-s) = 0$$

  - $D(s)D(-s) + \frac{R_{\rm zz}}{R_{\rm uu}} N(s)N(-s)$ is drawn using standard root locus rules - but it is symmetric wrt to both the real and imaginary axes.
  - For a stable system, we clearly just take the poles in the LHP.

Derivation of the SRL

- The closed-loop poles are given by the eigenvalues of

$$H \triangleq \begin{bmatrix} A & -B_u R_{\rm uu}^{-1} B_u^T \\ -C_z^T R_{\rm zz} C_z & -A^T \end{bmatrix} \rightarrow \qquad \det(sI - H) = 0$$

- **Note:** if A is invertible:

$$\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(A)\det(D - CA^{-1}B)$$

$$\Rightarrow \det(sI - H) = \det(sI - A)\det\left[(sI + A^T) - C_z^T R_{\rm zz} C_z (sI - A)^{-1} B_u R_{\rm uu}^{-1} B_u^T\right]$$

$$= \det(sI - A)\det(sI + A^T)\det\left[I - C_z^T R_{\rm zz} C_z (sI - A)^{-1} B_u R_{\rm uu}^{-1} B_u^T (sI + A^T)^{-1}\right]$$

- **Also:** $\det(I + ABC) = \det(I + CAB)$, and if $D(s) = \det(sI - A)$, then $D(-s) = \det(-sI - A^T) = (-1)^n \det(sI + A^T)$

$$\det(sI - H) = (-1)^n D(s) D(-s) \det\left[I + R_{\rm uu}^{-1} B_u^T (-sI - A^T)^{-1} C_z^T R_{\rm zz} C_z (sI - A)^{-1} B_u\right]$$

- If $G_{zu}(s) = C_z(sI - A)^{-1}B_u$, then $G_{zu}^T(-s) = B_u^T(-sI - A^T)^{-1}C_z^T$, so for SISO systems

$$\begin{aligned} \det(sI - H) &= (-1)^n D(s) D(-s) \det\left[I + R_{\rm uu}^{-1} G_{zu}^T(-s) R_{\rm zz} G_{zu}(s)\right] \\ &= (-1)^n D(s) D(-s) \left[I + \frac{R_{\rm zz}}{R_{\rm uu}} G_{zu}(-s) G_{zu}(s)\right] \\ &= (-1)^n \left[D(s) D(-s) + \frac{R_{\rm zz}}{R_{\rm uu}} N(s) N(-s)\right] = 0 \end{aligned}$$

Example 6–2

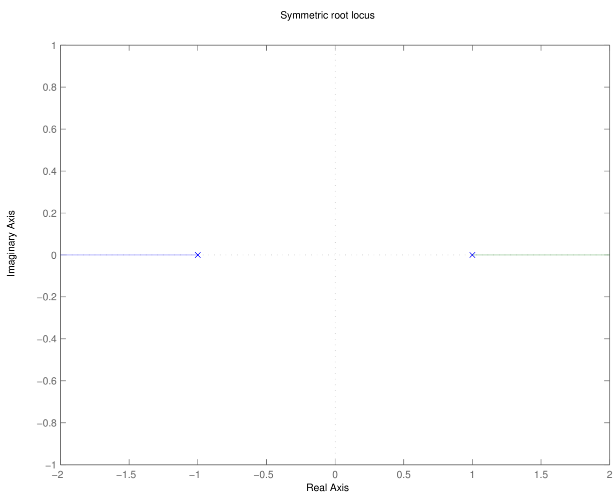
- Simple example from 4–12: A scalar system with $\dot{x} = ax + bu$ with cost ($R_{\rm xx} > 0$ and $R_{\rm uu} > 0$) $J = \int_0^\infty (R_{\rm zz}x^2(t) + R_{\rm uu}u^2(t))\,dt$

- The steady-state P solves $2aP + R_{\rm zz} - P^2b^2/R_{\rm uu} = 0$ which gives that $P = \frac{a+\sqrt{a^2+b^2R_{\rm zz}/R_{\rm uu}}}{R_{\rm uu}^{-1}b^2} > 0$
  - So that $u(t) = -Kx(t)$ where $K = R_{\rm uu}^{-1}bP = \frac{a+\sqrt{a^2+b^2R_{\rm zz}/R_{\rm uu}}}{b}$
  - and the closed-loop dynamics are

$$\begin{aligned}\dot{x} &= (a-bK)x = \left(a - \frac{b}{b}(a + \sqrt{a^2+b^2R_{\rm zz}/R_{\rm uu}})\right)x \\ &= -\sqrt{a^2+b^2R_{\rm zz}/R_{\rm uu}}\;x = A_{cl}x(t)\end{aligned}$$

- In this case, $G_{zu}(s) = b/(s-a)$ so that $N(s) = b$ and $D(s) = (s-a)$, and the SRL is of the form:

$$(s-a)(-s-a) + \frac{R_{\rm zz}}{R_{\rm uu}}b^2 = 0$$



- SRL is the same whether $a < 0$ (OL stable) or $a > 0$ (OL unstable)
  - But the CLP is always the one in the LHP
  - Explains result on 4–12 about why gain $K \neq 0$ for OL unstable systems, even for expensive control problem ($R_{\rm uu} \to \infty$)

SRL Interpretations

- For SISO case, define $R_{zz}/R_{uu} = 1/r$.

- Consider what happens as $r \rightsquigarrow \infty$ – **high control cost case**

$$\Delta(s) = D(s)D(-s) + r^{-1}N(s)N(-s) = 0 \quad \Rightarrow \quad \textbf{D(s)D(-s)=0}$$

  – So the n closed-loop poles are:
    - ◇ Stable roots of the open-loop system (already in the LHP.)
    - ◇ **Reflection** about the $\mathbf{j}\omega$-axis of the unstable open-loop poles.

- Consider what happens as $r \rightsquigarrow 0$ – **low control cost case**

$$\Delta(s) = D(s)D(-s) + r^{-1}N(s)N(-s) = 0 \quad \Rightarrow \quad \textbf{N(s)N(-s)=0}$$

  – Assume order of $N(s)N(-s)$ 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 system zeros in the RHP).
    - ◇ The system zeros at infinity (there are $n - m$ of these).

- 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**:

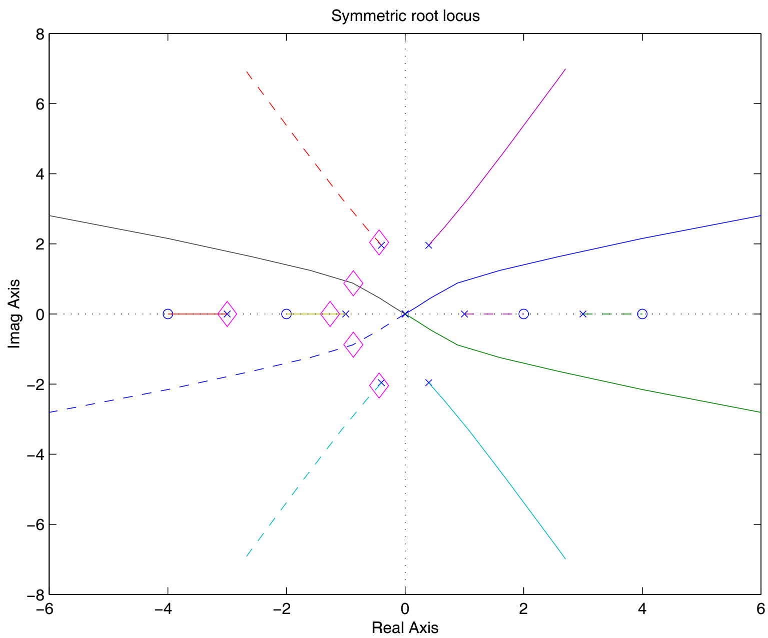
$$\pm\frac{l\pi}{n-m}\ ,\quad l=0,1,\ldots,\frac{n-m-1}{2}\ ,\qquad \textbf{(n-m) odd}$$

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

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

- **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 6.2: $G(s) = \frac{(s-2)(s-4)}{(s-1)(s-3)(s^2+0.8s+4)s^2}$



- 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 – consider oscillator with $\mathbf{x} = [\, p \;,\quad v \,]^T$

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

but we choose two cases for z

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

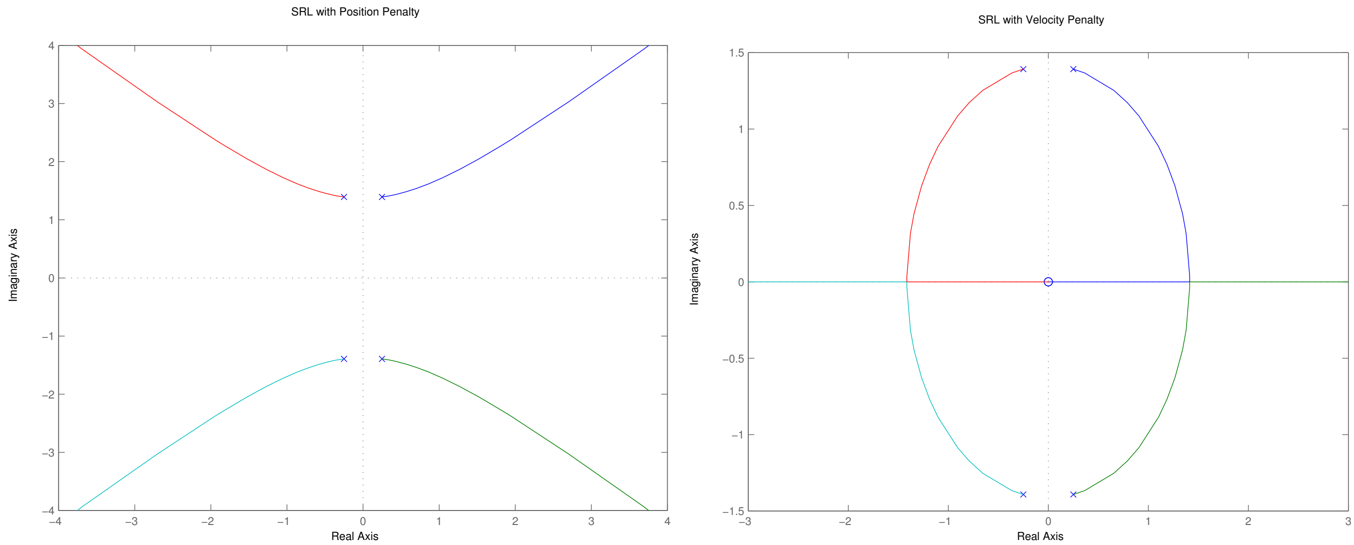


Figure 6.3: SRL with position (left) and velocity penalties (right)

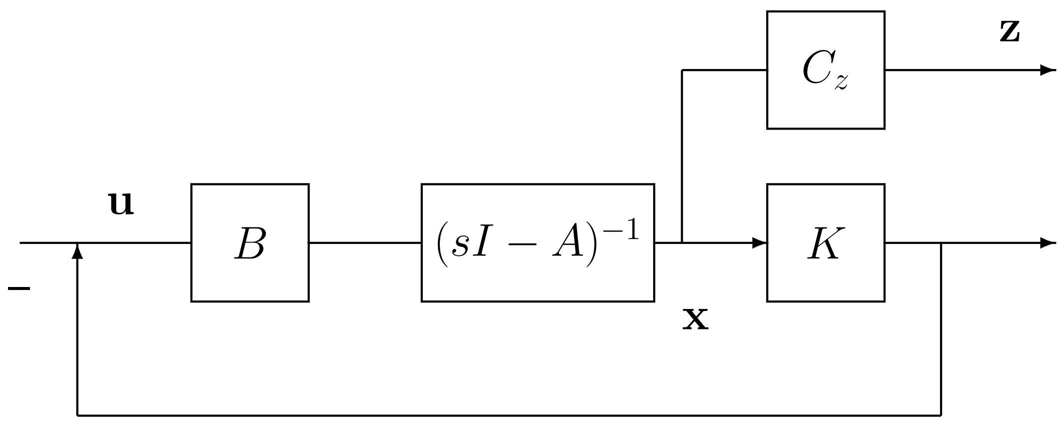
- Clearly, choosing a different C_z impacts the SRL because it completely changes the zero-structure for the system.

LQR Stability Margins

- 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.
- Particularly nice thing about the LQR approach is that the designer is focused on system performance issues

- Turns out that the news is even better than that, because LQR exhibits very good stability margins
  - Consider the LQR stability robustness.

$$\begin{aligned} J &= \int_0^\infty \mathbf{z}^T\mathbf{z} + \rho\mathbf{u}^T\mathbf{u}\; dt \\ \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{z} &= C_z\mathbf{x}, \quad R_{\rm xx} = C_z^T C_z \end{aligned}$$



- Study robustness in the frequency domain.
  - Loop transfer function $L(s) = K(sI - A)^{-1}B$
  - Cost transfer function $C(s) = C_z(sI - A)^{-1}B$

- Can develop a relationship between the open-loop cost $C(s)$ and the closed-loop return difference $I+L(s)$ called the **Kalman Frequency Domain Equality**

$$[I + L(-s)]^T [I + L(s)] = 1 + \frac{1}{\rho} C^T(-s) C(s)$$

- Sketch of Proof
  – Start with $\mathbf{u} = -K\mathbf{x}$, $K = \frac{1}{\rho} B^T P$, where

$$0 = -A^T P - PA - R_{xx} + \frac{1}{\rho} PBB^T P$$

  – Introduce Laplace variable s using $\pm sP$

$$0 = (-sI - A^T)P + P(sI - A) - R_{xx} + \frac{1}{\rho} PBB^T P$$

  – Pre-multiply by $B^T(-sI - A^T)^{-1}$, post-multiply by $(sI - A)^{-1} B$
  – Complete the square . . .

$$[I + L(-s)]^T [I + L(s)] = 1 + \frac{1}{\rho} C^T(-s) C(s)$$

- Can handle the MIMO case, but look at the SISO case to develop further insights ($s = \mathbf{j}\omega$)

$$\begin{aligned} [I + L(-s)]^T [I + L(s)] &= (I + L_r(\omega) - \mathrm{j}L_i(\omega))(I + L_r(\omega) + \mathrm{j}L_i(\omega)) \\ &\equiv |1 + L(\mathrm{j}\omega)|^2 \end{aligned}$$

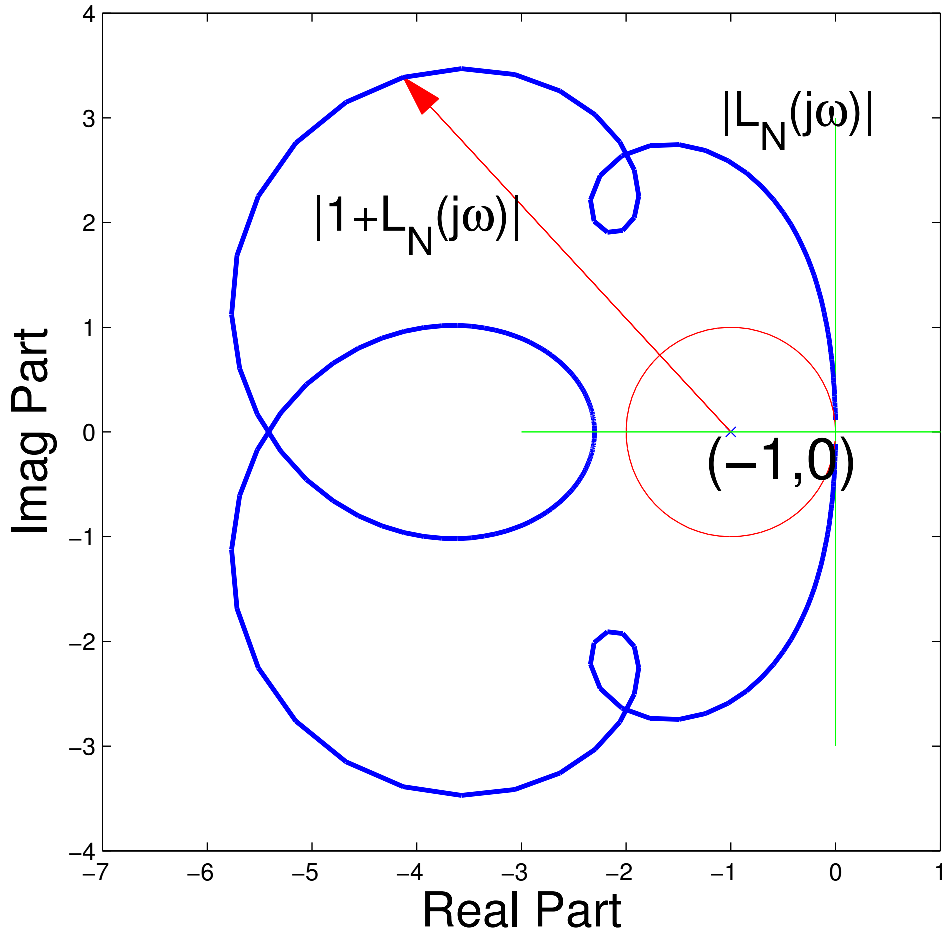
and

$$C^T(-\mathbf{j}\omega) C(\mathrm{j}\omega) = C_r^2 + C_i^2 = |C(\mathrm{j}\omega)|^2 \geq 0$$

- Thus the KFE becomes

$$|1 + L(\mathrm{j}\omega)|^2 = 1 + \frac{1}{\rho} |C(\mathrm{j}\omega)|^2 \geq 1$$

- **Implications:** The Nyquist plot of $L(j\omega)$ will always be outside the unit circle centered at (-1,0).



- Great, but why is this so significant? Recall the SISO form of the **Nyquist Stability Theorem:**

  If the loop transfer function $L(s)$ has P poles in the RHP s-plane (and $\lim_{s\to\infty} L(s)$ is a constant), then for closed-loop stability, the locus of $L(j\omega)$ for $\omega : (-\infty, \infty)$ must encircle the critical point (-1,0) P times in the **counterclockwise** direction (Ogata528)

- So we can directly prove stability from the Nyquist plot of $L(s)$. But what if the model is wrong and it turns out that the actual loop transfer function $L_A(s)$ is given by:

$$L_A(s) = L_N(s)[1 + \Delta(s)], \quad |\Delta(j\omega)| \le 1, \quad \forall \omega$$

- We need to determine whether these perturbations to the loop TF will change the decision about closed-loop stability

  ⇒ can do this directly by determining if it is possible to **change the number of encirclements of the critical point**

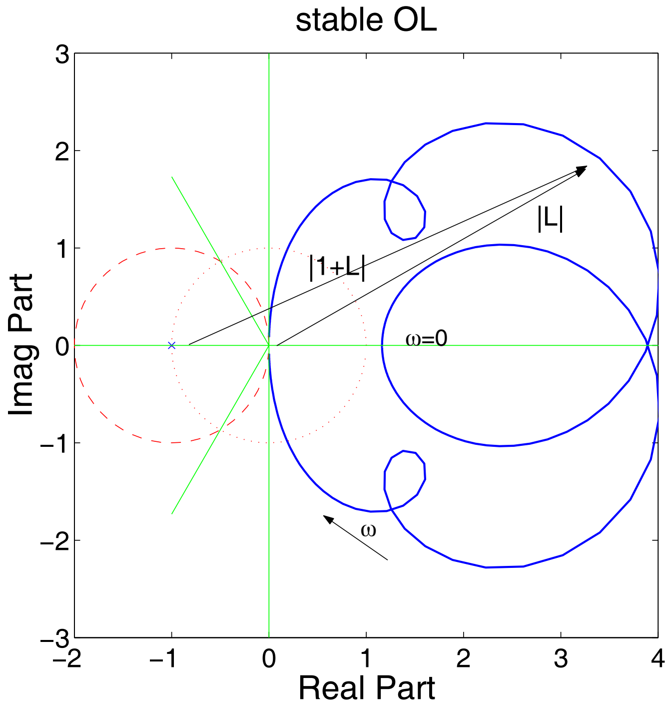
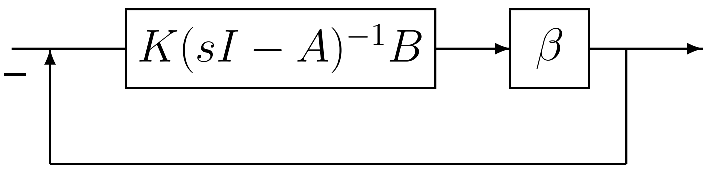


Figure 6.4: Example of LTF for an open-loop stable system

- Claim is that "since the LTF $L(j\omega)$ is guaranteed to be far from the critical point for all frequencies, then LQR is VERY robust."
  - Can study this by introducing a modification to the system, where nominally $\beta = 1$, but we would like to consider:
    - ◇ The gain $\beta \in \mathbb{R}$
    - ◇ The phase $\beta \in e^{j\phi}$



- In fact, can be shown that:
  - If open-loop system is stable, then any $\beta \in (0, \infty)$ yields a stable closed-loop system. For an unstable system, any $\beta \in (1/2, \infty)$ yields a stable closed-loop system $\Rightarrow$ gain margins are $(1/2, \infty)$
  - Phase margins of at least $\pm 60^\circ$

  $\Rightarrow$ which are both huge.

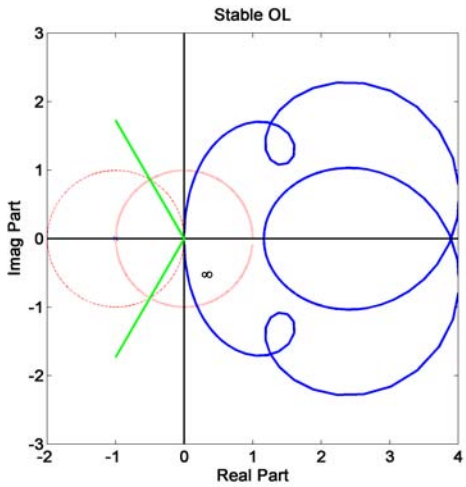


Figure 6.5: Example loop transfer functions for open-loop stable system.

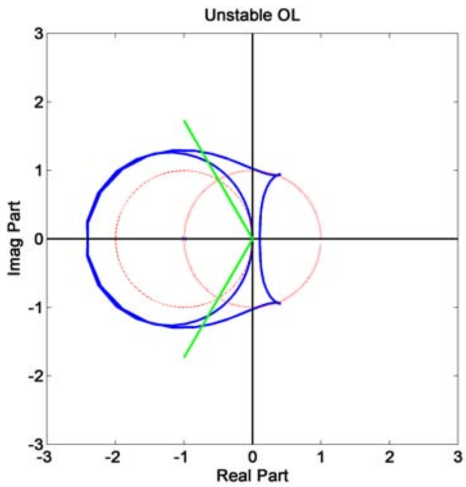


Figure 6.6: Example loop transfer functions for open-loop unstable system.

- While we have large margins, be careful because changes to some of the parameters in A or B can have a very large change to $L(s)$.

- Similar statements hold for the MIMO case, but it requires singular value analysis tools.

LTF for KDE

```
% Simple example showing LTF for KDE
% 16.323 Spring 2007
% Jonathan How
% rs2.m
%
clear all;close all;
set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontWeight','demi')
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontWeight','demi')

a=diag([-.75 -.75 -1 -1])+diag([-2 0 -4],1)+diag([2 0 4],-1);
b=[
       0.8180
    0.6602
    0.3420
    0.2897];
cz=[    0.3412    0.5341    0.7271    0.3093];
r=1e-2;
eig(a)
k=lqr(a,b,cz'*cz,r)
w=logspace(-2,2,200)';w2=-w(length(w):-1:1);
ww=[w2;0;w];
G=freqresp(a,b,k,0,1,sqrt(-1)*ww);

p=plot(G);
tt=[0:.1:2*pi]';Z=cos(tt)+sqrt(-1)*sin(tt);
hold on;plot(-1+Z,'r--');plot(Z,'r:','LineWidth',2);
plot(-1+1e-9*sqrt(-1),'x')
plot([0 0]',[-3 3]','k-','LineWidth',1.5)
plot([-3 6],[0 0]','k-','LineWidth',1.5)
plot([0 -2*cos(pi/3)],[0 -2*sin(pi/3)]','g-','LineWidth',2)
plot([0 -2*cos(pi/3)],[0 2*sin(pi/3)]','g-','LineWidth',2)
hold off
set(p,'LineWidth',2);
axis('square')
axis([-2 4 -3 3])

ylabel('Imag Part');xlabel('Real Part');title('Stable OL')
text(.25,-.5,'\infty')
print -dpng -r300 tf.png

%%%%%%%%%%%%%%%%%%%%%%

a=diag([-.75 -.75 1 1])+diag([-2 0 -4],1)+diag([2 0 4],-1);
r=1e-1;
eig(a)
k=lqr(a,b,cz'*cz,r)
G=freqresp(a,b,k,0,1,sqrt(-1)*ww);

p=plot(G);
hold on;plot(-1+Z,'r--');plot(Z,'r:','LineWidth',2);
plot(-1+1e-9*sqrt(-1),'x')
plot([0 0]',[-3 3]','k-','LineWidth',1.5)
plot([-3 6],[0 0]','k-','LineWidth',1.5)
plot([0 -2*cos(pi/3)],[0 -2*sin(pi/3)]','g-','LineWidth',2)
plot([0 -2*cos(pi/3)],[0 2*sin(pi/3)]','g-','LineWidth',2)
hold off
set(p,'LineWidth',2)
axis('square')
axis([-3 3 -3 3])

ylabel('Imag Part');xlabel('Real Part');title('Unstable OL')
print -dpng -r300 tf1.png
```