Linear Quadratic Optimal Control

6.1: Introduction to Linear Quadratic Optimal Control

- In our study of dynamic optimization so far, we've focused on obtaining a control function $\boldsymbol{u}$ which optimizes a specified cost as a function of time.
  - In the course of solving the optimization, we also obtain $\boldsymbol{x}$ and $\boldsymbol{\lambda}$.
  - So, our optimal $\boldsymbol{u}(t) = \boldsymbol{u}^*(t)$ is completely pre-determined by the problem set-up and initial conditions.
- Typically, the problem is solved for a given set of *initial conditions* in order to proceed to a specified set of *terminal constraints*.
- So, provided we stay precisely on the optimal path stipulated by $\boldsymbol{u}^*(t)$ all we have to do is check our watch and apply the appropriate control!

$\Rightarrow$ OPEN-LOOP CONTROL!

- But what do we do if (i) disturbances knock us off the optimal path, or (ii) we don't know exactly where we will start from, or (iii) we don't know exactly when we will start?
  - EITHER, we must solve a new problem for each different situation,
  - OR we can calculate a family of optimal paths so that all possible starting conditions lie on or very close to one of the paths.

- In this second case, we can simply look at where we are and then decide what to do next.

$\Rightarrow$ FEEDBACK CONTROL

<u>*Example: Zermelo's Problem*</u>

- Recall the solution obtained previously:

$$\begin{aligned}\frac{x}{h} &= \frac{1}{2}\Big\{\sec\theta_f\left(\tan\theta - \tan\theta_f\right) + \tan\theta\left(\sec\theta_f - \sec\theta\right) \\ &\quad + \ln\left[\frac{\tan\theta + \sec\theta}{\tan\theta_f + \sec\theta_f}\right]\Big\} \\ \frac{y}{h} &= \sec\theta - \sec\theta_f\end{aligned}$$

- So, given the current x and y positional values, then θ and θ_f can be calculated:

$$\tan\theta = \tan\theta_f - \frac{V\left(t_f - t\right)}{h}$$

  – We can thus go backward in time to identify θ, x, and y for any given θ_f $\Rightarrow$ a *family of optimal paths*

- Obviously, the problem of generating feedback control solutions using the techniques suggested above will, in general, be an extremely tedious one.

  – One saving grace: a unique optimal control vector will, in general, be associated with each point; so we don't have to worry about selecting the proper solution from among many alternatives.

  – Still, the solution process is rather laborious.

- Is there another way? YES $\Rightarrow$ DYNAMIC PROGRAMMING

6.2: Dynamic Programming

- Basis for the approach: if we start from a given point and proceed optimally to the end, there will be a unique optimal value for the cost associated with this process (J^*)
  - This idea is known as Bellman's Principle of Optimality:

    "In place of determining the optimal sequence of decisions from the *fixed* state of the system, we wish to determine the optimal decision to be made at *any* state of the system. Only if we know the latter, do we understand the instrinsic structure of the solution." [1]
  - Bellman's ideas were developed in part to compete with the Pontryagin minimum principle during the same timeframe.
  - The central idea is to work backward in time from some desired goal states.
- Clearly, J^* is a function of the initial point; so J^* is often referred to as the OPTIMAL RETURN FUNCTION.
  - Using Hamilton-Jacobi theory, the solution of a special partial differential equation that is satisfied by J^* can be used to determine the optimal feedback control policy.
  - Furthermore, this theory has been generalized to include multistage systems and combinatorial problems by Bellman to produce the complete Dynamic Programming approach.

[1] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

- The complete development of this theory is beyond the scope of this course, but I do want to highlight the result, an important interpretation of the result, and the significance of (and difficulties with) the theory.
- An in-depth treatment of this approach is developed in ECE 5530, Multivariable Control Systems II.

RESULT: *Hamilton-Jacobi-Bellman partial differential equation*

- The optimal control policy is given by the solution of
$$-\frac{\partial J^*}{\partial t} = H^*\left(\boldsymbol{x},\ \frac{\partial J^*}{\partial \boldsymbol{x}},\ t\right)$$
where
$$H^*\left(\boldsymbol{x},\ \frac{\partial J^*}{\partial \boldsymbol{x}},\ t\right) = \min_u H\left(\boldsymbol{x},\ \frac{\partial J^*}{\partial \boldsymbol{x}},\ u,\ t\right)$$
  - This is an alternative approach to the Calculus of Variations for solving dynamic optimization problems.
  - The result is a first-order nonlinear partial differential equation that must be solved with appropriate boundary conditions.
  - The equation states that the optimal u^* minimizes globally the Hamiltonian H holding $\boldsymbol{x}$, ${}^{\partial J^*}/_{x}$, and t constant – this is another statement of Pontryagin's Minimum Principle.

INTERPRETATION: *The principle of optimality*

> "An optimal policy has the property that, no matter what the previous controls have been, the remaining controls must constitute an optimal policy with regard to the states resulting from the previous control"

SIGNIFICANCE:

1. Emphasizes the existence of optimal feedback control laws.
2. Provides a straightforward approach to solving discrete combinatorial problems (e.g., Bryson & Ho, pp. 136-141); these are multi-stage optimization problems in which there are only a small number of possible choices of the control at each stage.

DIFFICULTIES:

- It is not generally feasible to solve the Hamilton-Jacobi-Bellman partial differential equation for practical nonlinear systems.
  - So the development of *exact* feedback control schemes is typically out of reach.
  - But if we focus on *linear* dynamic systems and impose quadratic performance criteria and constraints, appropriate feedback controllers *can* be synthesized.

    $\Rightarrow$ LINEAR QUADRATIC OPTIMAL CONTROL
  - Furthermore, once the LQ techniques have been developed, they may be applied to nonlinear problems via "perturbation guidance" $\Rightarrow$ identify optimal feedback paths in the neighborhood of a previously-identified nominal optimal path.

6.3: Linear Quadratic Control Problem

- In this section, we examine the Linear Quadratic Control problem and develop techniques to generate optimal feedback control laws
- A general continuous-time dynamic system may be written:

$$\dot{\boldsymbol{x}}(t) = A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t)$$

$$\boldsymbol{y}(t) = C(t)\boldsymbol{x}(t) + D(t)\boldsymbol{u}(t)$$

  where here we allow for possibly time-varying system matrices.
- For simplicity, we shall assume *constant-valued* matrices to write:

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = C\boldsymbol{x}(t) + D\boldsymbol{u}(t)$$

  – Recall that vector-valued variables $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ allow for multiple inputs and multiple outputs.
- Types of Control Algorithms and Associated Costs:
  1. TERMINAL CONTROLLER $\Rightarrow$ designed to bring a system close to desired conditions at some specified (or unspecified) *terminal time*
     (a) "soft" end constraints

$$J = \frac{1}{2}\boldsymbol{x}^T(t_f)P_f\boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f}\left\{\boldsymbol{x}^T(t)\,Q\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\,R\boldsymbol{u}(t)\right\}dt$$

     where P_f, Q and R are constant-valued, positive definite, symmetric matrices that introduce weighting into the cost function.

     (b) "hard" end constraints

$$J = \frac{1}{2}\int_{t_0}^{t_f}\left\{\boldsymbol{x}^T(t)\,Q\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\,R\boldsymbol{u}(t)\right\}dt$$

$$\boldsymbol{x}_i(t_f) = 0; \qquad i = 1, 2, \ldots, q$$

2. REGULATOR $\Rightarrow$ designed to keep a stationary system within an acceptable deviation from a reference condition using acceptable amounts of control

$$J = \frac{1}{2}\int_{t_0}^{t_f} \left\{ \boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u} \right\} dt \quad \text{with} \quad \left(t_f - t_0\right) \to \infty$$

Here, the reference is set to zero, so we start from an initial condition and wish to return to the equilibrium state.

- We begin our investigation with the simplest form of the linear quadratic optimal control problem $\Rightarrow$ the "soft" end constraint problem presented above:

$$J = \frac{1}{2}\boldsymbol{x}^T(t_f) P_f \boldsymbol{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left\{ \boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u} \right\} dt$$

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$$

$$P_f \geq 0 \qquad Q \geq 0 \qquad R > 0$$

  – For the case where $t_f \to \infty$, is the soft end constraint necessary?
  – The selection of P_f, Q, and R is based on our desire to obtain "acceptable" levels for $\boldsymbol{x}(t_f)$, $\boldsymbol{x}(t)$, and $\boldsymbol{u}(t)$, respectively
  – Typically, these matrices will be diagonal – some rules of thumb for selecting these matrices are:
    1. $\dfrac{1}{P_f(i,i)} =$ max acceptable value of $x_i^2(t_f)$
    2. $\dfrac{1}{Q(i,i)} = \left(t_f - t_0\right) *$ max acceptable value of $x_i^2(t)$
    3. $\dfrac{1}{R(i,i)} = \left(t_f - t_0\right) *$ max acceptable value of $u_i^2(t)$

- How do we solve this optimization problem? Using the Calculus of Variations

$$\bar{J} = \frac{1}{2}\boldsymbol{x}_f^T P_f \boldsymbol{x}_f + \int_{t_0}^{t_f} \left\{ \frac{1}{2}\left[\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u}\right] + \boldsymbol{\lambda}^T \left[A\boldsymbol{x} + B\boldsymbol{u} - \dot{\boldsymbol{x}}\right] \right\} dt$$

$$\delta \bar{J} = \boldsymbol{x}_f^T P_f \delta \boldsymbol{x}(t_f) + \boldsymbol{\lambda}^T(t_0)\delta\boldsymbol{x}(t_0) - \boldsymbol{\lambda}^T(t_f)\delta\boldsymbol{x}(t_f)$$

$$+ \int_{t_0}^{t_f} \left\{ \left[\boldsymbol{x}^T Q + \boldsymbol{\lambda}^T A + \dot{\boldsymbol{\lambda}}^T\right]\delta\boldsymbol{x} + \left[\boldsymbol{u}^T R + \boldsymbol{\lambda}^T B\right]\delta\boldsymbol{u} \right\} dt$$

– Yielding the following equations:

$$\begin{aligned}
\dot{\boldsymbol{\lambda}}^T &= -\boldsymbol{x}^T Q - \boldsymbol{\lambda}^T A \\
\dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\
\boldsymbol{u}^T R + \boldsymbol{\lambda}^T B &= 0 \\
\boldsymbol{\lambda}^T(t_f) &= \boldsymbol{x}_f^T P_f
\end{aligned}$$

– Rearranging, we have

$$\begin{aligned}
\dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\
\dot{\boldsymbol{\lambda}} &= -Q\boldsymbol{x} - A^T\boldsymbol{\lambda} \\
\boldsymbol{u} &= -R^{-1}B^T\boldsymbol{\lambda}
\end{aligned}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \qquad \boldsymbol{\lambda}(t_f) = P_f \boldsymbol{x}_f$$

- We would have gotten the same result using the following equations:

$$\frac{\partial H}{\partial \boldsymbol{u}} = 0 \qquad \frac{\partial H}{\partial \boldsymbol{x}} = -\dot{\lambda}^T \qquad \frac{\partial H}{\partial \boldsymbol{\lambda}} = \dot{\boldsymbol{x}}$$

$$\boldsymbol{\lambda}^T(t_f) = \frac{\partial \varphi}{\partial \boldsymbol{x}(t_f)}$$

where

$$H = \frac{1}{2}\left\{\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u}\right\} + \boldsymbol{\lambda}^T\left\{A\boldsymbol{x} + B\boldsymbol{u}\right\}$$

$$\varphi = \frac{1}{2}\boldsymbol{x}_f^T P_f \boldsymbol{x}_f$$

- We can solve these equations by augmenting the state vector:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \left[\begin{array}{c|c} A & -BR^{-1}B^T \\ \hline -Q & -A^T \end{array}\right] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix}$$

- NOTE: This matrix is referred to as the HAMILTONIAN matrix, $\mathcal{H}$
  - So, we must find the solution to this set of $2n$ linear homogeneous differential equations with $\boldsymbol{x}(t_o)$ given and $\boldsymbol{\lambda}(t_f) = P_f \boldsymbol{x}_f$

$$\begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix} = e^{\mathcal{H}(t-t_0)} \begin{bmatrix} \boldsymbol{x}(t_0) \\ \boldsymbol{\lambda}(t_0) \end{bmatrix} = \Phi(t-t_0) \begin{bmatrix} \boldsymbol{x}(t_0) \\ \boldsymbol{\lambda}(t_0) \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix} = \begin{bmatrix} \phi_{11}(t-t_0) & \phi_{12}(t-t_0) \\ \phi_{21}(t-t_0) & \phi_{22}(t-t_0) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t_0) \\ \boldsymbol{\lambda}(t_0) \end{bmatrix}$$

$\Rightarrow$

$$\begin{aligned} \boldsymbol{x}(t) &= \phi_{11}\boldsymbol{x}(t_0) + \phi_{12}\boldsymbol{\lambda}(t_0) \\ \boldsymbol{\lambda}(t) &= \phi_{21}\boldsymbol{x}(t_0) + \phi_{22}\boldsymbol{\lambda}(t_0) \end{aligned}$$

- $\boldsymbol{x}(t_0)$ and $\boldsymbol{\lambda}(t_f)$ are known, but $\boldsymbol{\lambda}(t_0)$ and $\boldsymbol{x}(t_f)$ are not

$$\begin{aligned} \boldsymbol{x}(t_f) &= \phi_{11}\left(t_f - t_0\right)\boldsymbol{x}(t_0) + \phi_{12}\left(t_f - t_0\right)\boldsymbol{\lambda}(t_0) \\ \boldsymbol{\lambda}(t_f) &= \phi_{21}\left(t_f - t_0\right)\boldsymbol{x}(t_0) + \phi_{22}\left(t_f - t_0\right)\boldsymbol{\lambda}(t_0) = P_f \boldsymbol{x}(t_f) \end{aligned}$$

$\Rightarrow$

$$\left(P_f\phi_{11} - \phi_{21}\right)\boldsymbol{x}(t_0) = \left(\phi_{22} - P_f\phi_{12}\right)\boldsymbol{\lambda}(t_0)$$

$$\boldsymbol{\lambda}(t_0) = \left(\phi_{22} - P_f\phi_{12}\right)^{-1}\left(P_f\phi_{11} - \phi_{21}\right)\boldsymbol{x}(t_0)$$

- And now that we know $\boldsymbol{x}(t_0)$ and $\boldsymbol{\lambda}(t_0)$, we can calculate $\boldsymbol{x}(t)$, $\boldsymbol{\lambda}(t)$ and $\boldsymbol{u}(t)$

6.4: Linear Quadratic Control Problem Example

Example 6.1

$$\dot{x} = u$$

$$J = \frac{1}{2}px^2(t_f) + \frac{1}{2}\int_0^{t_f} \left(qx^2 + ru^2\right) dt$$

SOLUTION:

- The Hamiltonian is given by

$$H = \frac{1}{2}\left(qx^2 + ru^2\right) + \lambda u$$

- Writing the optimality equations gives,

$$\frac{\partial H}{\partial u} = ru + \lambda = 0 \quad \Rightarrow \quad u = -\frac{\lambda}{r}$$

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = -qx \qquad \dot{x} = -\frac{\lambda}{r}$$

- We can stack the state and co-state equations to write the augmented form

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{r} \\ -q & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix}$$

$$\dot{z} = \mathcal{H}z$$

- Using Laplace transforms gives the solution as

$$sZ(s) - z(0) = \mathcal{H}Z(s)$$

$$(sI - \mathcal{H})\, Z(s) = z(0)$$

$$Z(s) = (sI - \mathcal{H})^{-1} z(0)$$

- But we have

$$(sI - A)^{-1} = \left(\frac{1}{s^2 - 1/qr}\right) \begin{bmatrix} s & -\dfrac{1}{r} \\ -q & s \end{bmatrix}$$

- For the diagonal terms,

$$\frac{s}{s^2 - 1/qr} = \frac{A}{s + \sqrt{q/r}} + \frac{B}{s - \sqrt{q/r}}$$

where solving for A and B gives

$$A = 1/2$$
$$B = 1/2$$

and

$$\mathcal{L}^{-1}\left\{\frac{s}{s^2 - 1/qr}\right\} = \frac{1}{2}e^{-\sqrt{q/r}t} + \frac{1}{2}e^{+\sqrt{q/r}t} = \cosh\left(\sqrt{q/r}t\right)$$

- And for the off-diagonal terms,

$$\frac{1}{s^2 - 1/qr} = \frac{A}{s + \sqrt{q/r}} + \frac{B}{s - \sqrt{q/r}}$$

where solving for A and B here gives

$$A = -\frac{1}{2}\sqrt{r/q}$$
$$B = \frac{1}{2}\sqrt{r/q}$$

and

$$\mathcal{L}^{-1}\left\{\frac{1}{s^2 - 1/qr}\right\} = \frac{1}{2}\sqrt{r/q}\left(-e^{-\sqrt{q/r}t} + e^{+\sqrt{q/r}t}\right) = \sqrt{r/q}\sinh\left(\sqrt{q/r}t\right)$$

- So now we can write the state transition matrix as

$$e^{\mathcal{H}t} = \left[\begin{array}{c|c} \cosh\left(\sqrt{q/r}t\right) & -\left(\dfrac{1}{\sqrt{qr}}\right)\sinh\left(\sqrt{q/r}t\right) \\ \hline -\sqrt{rq}\sinh\left(\sqrt{q/r}t\right) & \cosh\left(\sqrt{q/r}t\right) \end{array}\right]$$

and the complete solution as

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \left[\begin{array}{c|c} \cosh\left(\sqrt{q/r}\,t\right) & -\left(\dfrac{1}{\sqrt{qr}}\right)\sinh\left(\sqrt{q/r}\,t\right) \\ \hline -\sqrt{rq}\sinh\left(\sqrt{q/r}\,t\right) & \cosh\left(\sqrt{q/r}\,t\right) \end{array}\right] \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix}$$

- So,

$$x(t) = x_0\cosh\left(\sqrt{q/r}\,t\right) - \lambda_0\left(\frac{1}{\sqrt{qr}}\right)\sinh\left(\sqrt{q/r}\,t\right)$$

$$\lambda(t) = -x_0\sqrt{rq}\sinh\left(\sqrt{q/r}\,t\right) + \lambda_0\cosh\left(\sqrt{q/r}\,t\right)$$

which gives,

$$u(t) = x_0\sqrt{q/r}\sinh\left(\sqrt{q/r}\,t\right) - \lambda_0\left(\frac{1}{r}\right)\cosh\left(\sqrt{q/r}\,t\right)$$

- But how do we find λ_0 ?
- Substituting $t = t_f$ we can write,

$$x_0\left\{p\cosh\left(\sqrt{q/r}\,t_f\right) + \sqrt{rq}\sinh\left(\sqrt{q/r}\,t_f\right)\right\}$$

$$= \lambda_0\left\{\cosh\left(\sqrt{q/r}\,t_f\right) + p\left(\frac{1}{\sqrt{qr}}\right)\sinh\left(\sqrt{q/r}\,t_f\right)\right\}$$

- Solving we obtain,

$$\lambda_0 = \left\{\frac{p + \sqrt{rq}\tanh\left(\sqrt{q/r}\,t_f\right)}{1 + p\left(\frac{1}{\sqrt{qr}}\right)\tanh\left(\sqrt{q/r}\,t_f\right)}\right\}x_0$$

- Substituting, we find that $u(t)$ is a function of x_0, q, r, p, t_f, and t.
  - NOTE: In the special case where $q = 0$ and $r = 1$, we can use L'Hopital's rule to find

$$\lambda_0 = \left\{\frac{p}{1+pt_f}\right\} x_0 = \lambda(t) \quad \Rightarrow \quad u(t) = -\left\{\frac{p}{(1+pt_f)}\right\} x_0$$

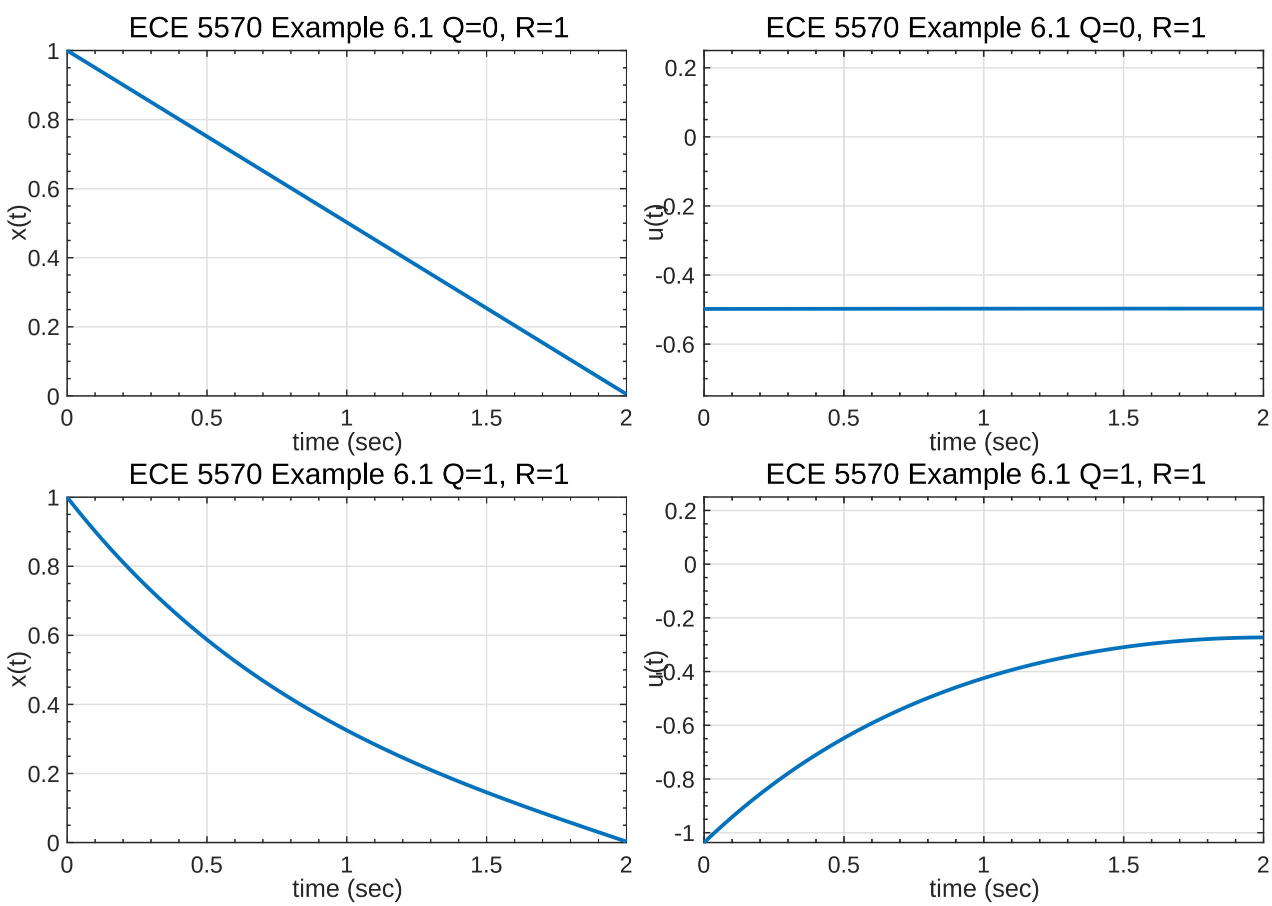
giving *constant* control input.

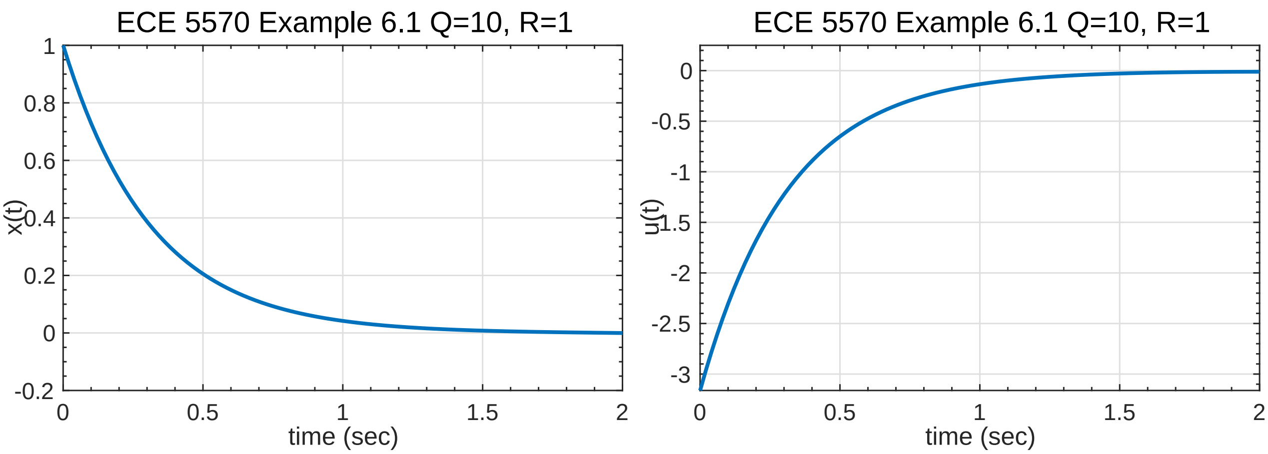
– Additionally,

$$x_f = \frac{x_0}{1+pt_f}$$

so that as $p \to \infty,\ x_f \to 0$

- The accompanying plots show results for the optimal output and input for various values of weighting factor q; $r = 1$ and $p = 100$ in all cases shown.





- One problem with this Calculus of Variations approach to solving this problem is that $\boldsymbol{u}(t)$ is a function of time and initial conditions.

- From our earlier solution we found,

$$\boldsymbol{u}(t) = -R^{-1}B^T\boldsymbol{\lambda}(t)$$

and the state and co-state in terms of the state transition matrix,

$$\boldsymbol{\lambda}(t) = \phi_{21}(t-t_0)\,\boldsymbol{x}(t_0) + \phi_{22}(t-t_0)\,\boldsymbol{\lambda}(t_0)$$
$$\boldsymbol{x}(t) = \phi_{11}(t-t_0)\,\boldsymbol{x}(t_0) + \phi_{12}(t-t_0)\,\boldsymbol{\lambda}(t_0)$$

$$e^{\mathcal{H}(t-t_0)} = \left[\begin{array}{c|c} \phi_{11} & \phi_{12} \\ \hline \phi_{21} & \phi_{22} \end{array}\right]$$

where we were able to solve for $\boldsymbol{\lambda}(t_0)$ in terms of $\boldsymbol{x}(t_0)$ as

$$\boldsymbol{\lambda}(t_0) = -M(t_f)\boldsymbol{x}(t_0)$$

$$M(t_f) = \left\{P_f\phi_{12} - \phi_{22}\right\}^{-1}\left\{P_f\phi_{11} - \phi_{21}\right\}$$

- It would be nice, however, to identify $\boldsymbol{u}(t)$ in terms of $\boldsymbol{x}(t) \Rightarrow$ this would give us a FEEDBACK LAW

- Can this be done? YES!

$$\begin{aligned}
\boldsymbol{u}(t) &= -R^{-1}B^T\left\{\phi_{21}\left(t-t_0\right)-\phi_{22}\left(t-t_0\right)M(t_f)\right\}\boldsymbol{x}\left(t_0\right)\\
\boldsymbol{x}\left(t\right) &= \left\{\phi_{11}\left(t-t_0\right)-\phi_{12}\left(t-t_0\right)M(t_f)\right\}\boldsymbol{x}\left(t_0\right)
\end{aligned}$$

$\Rightarrow$

$$\begin{aligned}
\boldsymbol{u}(t) &= -R^{-1}B^T\left\{\phi_{21}-\phi_{22}M\right\}\left\{\phi_{11}-\phi_{12}M\right\}^{-1}\boldsymbol{x}\left(t\right)\\
&= -R^{-1}B^T P(t)\boldsymbol{x}(t) \quad \left\{\boldsymbol{\lambda}(t) = P(t)\boldsymbol{x}(t)\right\}
\end{aligned}$$

- NOTE: this approach is not necessarily the best way to solve for $P(t)$, but it does demonstrate the construction of $P(t)$

- Is this result surprising? NO!
  - Since $\boldsymbol{u}(t)$ is a function of $\boldsymbol{x}(t_0)$, the Principle of Optimality indicates that any time t may be regarded as a new initial time, t_0
  - So our optimal control will always be a function of the current states!

6.5: Feedback Form of the Linear Quadratic Control Problem

- We've seen that the LQ optimal control problem with "soft" end constraints yields a full-state feedback control law of the form

$$\boldsymbol{u}(t) = -K(t)\boldsymbol{x}(t)$$

with a time-varying control law:

$$K(t) = R^{-1}B^T P(t)$$

- Question: How do we compute the time-varying matrix function $P(t)$?
- Find $e^{\mathcal{H}t}$ and compute $P(t)$ using the following relationships:

$$\left.\begin{aligned} \boldsymbol{x}(t) &= \phi_{11}\left(t-t_f\right)\boldsymbol{x}(t_f) + \phi_{12}\left(t-t_f\right)\boldsymbol{\lambda}\left(t_f\right) \\ \boldsymbol{\lambda}(t) &= \phi_{21}\left(t-t_f\right)\boldsymbol{x}(t_f) + \phi_{22}\left(t-t_f\right)\boldsymbol{\lambda}\left(t_f\right) \end{aligned}\right\} \quad e^{\mathcal{H}(t-t_f)} = \left[\begin{array}{c|c} \phi_{11} & \phi_{12} \\ \hline \phi_{21} & \phi_{22} \end{array}\right]$$

$$\boldsymbol{\lambda}\left(t_f\right) = P_f\boldsymbol{x}(t_f)$$

$$\boldsymbol{\lambda}(t) = \left(\phi_{21} + \phi_{22}P_f\right)\boldsymbol{x}(t_f) \qquad \boldsymbol{x}(t_f) = \left(\phi_{11} + \phi_{12}P_f\right)^{-1}\boldsymbol{x}(t)$$

$$\begin{aligned} \boldsymbol{\lambda}(t) = &\left\{\phi_{21}\left(t-t_f\right) + \phi_{22}\left(t-t_f\right)P_f\right\} \\ &\times \left\{\phi_{11}\left(t-t_f\right) + \phi_{12}\left(t-t_f\right)P_f\right\}^{-1}\boldsymbol{x}(t) \end{aligned}$$

  - Problems:
    - $e^{\mathcal{H}t}$ may be difficult to derive analytically
    - Numerical methods may be inaccurate due to different exponetial growth rates within $e^{\mathcal{H}t}$

- Integrate a "matrix Ricatti equation" (the "sweep" method)

$$\boldsymbol{\lambda}(t) = P(t)\boldsymbol{x}(t) \;\Rightarrow$$

$$\dot{\boldsymbol{\lambda}} = \dot{P}\boldsymbol{x} + P\dot{\boldsymbol{x}} = -A^T\boldsymbol{\lambda} - Q\boldsymbol{x}$$
$$\dot{\boldsymbol{x}} = A\boldsymbol{x} - BR^{-1}B^T\boldsymbol{\lambda}$$

$\Rightarrow$

$$\left\{\dot{P} + PA - PBR^{-1}B^TP\right\}\boldsymbol{x} = \left\{-A^TP - Q\right\}\boldsymbol{x}$$

$$\dot{P} = -PA - A^TP + PBR^{-1}B^TP - Q \quad P(t_f) = P_f$$

– Using a numerical integration method, we can let

$$\tau = t_f - t \qquad d\tau = -dt$$

$\Rightarrow$

$$dP = PA + A^TP - PBR^{-1}B^TP + Q$$

and let τ go from 0 to t_f $\left\{P(\tau = 0) = P_f\right\}$

Example 6.2

- First-order problem done previously with $q = 0$ and $r = 1$
- Putting into standard state-space form,

$$\dot{x}(t) = [0]\,x(t) + [1]\,u(t)$$

from which we have

$$A = 0$$
$$B = 1$$

- Therefore, the Ricatti equation simplifies...

$$\dot{P} + PA + A^TP - PBR^{-1}B^TP = -Q$$
$$\dot{P} - P^2 = 0$$

or,

$$\dot{P} = P^2$$

- Writing as

$$\frac{dP}{dt} = P^2$$

and rearanging, we can integrate to obtain

$$\int \frac{dP}{P^2} = \int_{t_f}^{t} dt$$

$$-P^{-1}\Big|_{t_f}^{t} = t - t_f$$

$$\frac{1}{P_f} - \frac{1}{P(t)} = t - t_f$$

$$\begin{aligned} P(t) &= \frac{1}{1/p + \left(t_f - t\right)} \\ &= \frac{p}{1 + p\left(t_f - t\right)} \end{aligned}$$

- Putting it all together, the state equation can be written,

$$\begin{aligned} \dot{x}(t) &= -P(t)\,x(t) \\ &= -\frac{p}{1 + p\left(t_f - t\right)} x(t) \end{aligned}$$

or as an ordinary first-order differential equation,

$$\dot{x}(t) + \frac{p}{1 + p\left(t_f - t\right)} x(t) = 0$$

- Here we can use a standard differential equation solver (e.g., Matlab's ode45) to solve, giving the same results as obtained earlier.

6.6: Linear Regulator Problem

- Remember from the last section that the standard form for the linear quadratic optimal control problem is:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$$

$$J = \frac{1}{2}\boldsymbol{x}^T(t_f)P_f\boldsymbol{x}(t_f) + \int_{t_0}^{t_f} \frac{1}{2}\left\{\boldsymbol{x}^T Q\boldsymbol{x} + \boldsymbol{u}^T R\boldsymbol{u}\right\} dt$$

  with

$$P_f \ge 0 \quad Q \ge 0 \quad R > 0$$

- Using the Calculus of Variations, we've shown that the optimal control can be written in state feedback form as:

$$\boldsymbol{u}(t) = -K(t)\boldsymbol{x}(t) = -R^{-1}B^T P(t)\boldsymbol{x}(t)$$

  - $\boldsymbol{u}(t)$ is time-varying (depending on time-to-go)
  - $P(t)$ can be identified either by appropriate manipulation of transition matrices or by integrating a matrix Ricatti equation

- Now, we want to focus on a special subset of this category of problems: THE REGULATOR PROBLEM
- What is a regulator? A feedback controller designed to keep a stationary system within an acceptable deviation from a reference condition using acceptable amounts of control
- Example: a satellite pointing problem
- Assumptions associated with the regulator problem:
  - The system is time-invariant (e.g., A and B are constant)
  - The Q and R matrices in J are constant
  - $t_f - t_0 \to \infty$

- What are the implications of these assumptions?
  - The cost function reduces to

$$J = \frac{1}{2}\int_{t_0}^{t_f} \left(\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u}\right) dt$$

- The term $1/2\boldsymbol{x}^T(t_f)P_f\boldsymbol{x}(t_f)$ in J can be eliminated since the terminal time is so far into the future; but there must be a running cost on at least some of the states (i.e., $Q \neq 0$) for this problem to be feasible
- Under these conditions, a *constant*, finite solution to the matrix Ricatti equation exists

$$\dot{P} = -PA - A^T P + PBR^{-1}B^T P - Q = 0$$

$$\Rightarrow \; P(t) \to P_0 \text{ as } t_f - t_0 \to \infty$$

- NOTE: this result assumes:
  - $[A,\; B]$ is stabilizable
  - $[A,\; C]$ is detectable
  - $Q = \rho C^T C$
- So, the feedback gain matrix, $K(t)$, will be *constant* and it can be shown that the controlled system will be stable if $P_0 > 0$
- But how do we find P_0 ?
  - If $\dot{P} = 0$, the matrix Ricatti equation becomes an *algebraic* equation that can be solved for the constant matrix P
  - But since the resulting equation is quadratic in P, more than one solution will appear $\Rightarrow$ the extraneous solutions can usually be eliminated using the fact that P_0 must be positive definite.

6.7: Linear Regulator Problem: Example

- This example is taken from Bryson & Ho, pg. 168:

$$\dot{x} = -\frac{1}{\tau}x + u \qquad J = \frac{1}{2}\int_0^{\infty}\left(qx^2 + ru^2\right)dt$$

SOLUTION:

- From the state-space equations, we have

$$A = -\frac{1}{\tau}, \quad B = 1$$

and from the cost function,

$$Q = q, \quad R- = r$$

- Utilizing our previous solution approach for the infinite-time regulator, we have

$$u(t) = -R^{-1}B^T P_0 x(t) = -\frac{1}{r}P_0 x(t)$$

where P_0 is found as the solution to the algebraic Ricatti equation

$$-P_0 A - A^T P_0 + P_0 B R^{-1} B^T P_0 - Q = 0$$

Upon subsitutiing for A, B, Q and R we have,

$$\frac{2P_0}{\tau} + \frac{P_0^2}{r} - q = 0$$

giving the quadratic equation,

$$P_0^2 + \frac{2r}{\tau}P_0 - qr = 0$$

and solving, we obtain

$$P_0 = -\frac{r}{\tau} \pm \sqrt{\left(\frac{r}{\tau}\right)^2 + qr}$$

- The quadratic gives two possible solutions... but we are only interested in the case where $P_0 > 0 \Rightarrow$

$$P_0 = \frac{r}{\tau}\left\{\sqrt{1 + q\frac{\tau^2}{r}} - 1\right\}$$

so the optimal input is given by

$$u^*(t) = -\left\{\sqrt{\frac{1}{\tau^2} - \frac{q}{r}} - \frac{1}{\tau}\right\} x(t)$$

- Another method of finding P_0 is to integrate the matrix Riccati equation backwards in time using numerical techniques until P settles down to a constant
  - Although $P_f = 0$, the fact that $Q \neq 0$ guarantees the existence of a nonzero P_0
  - Unfortunately, this approach is computationally expensive
- Yet another alternative can be found using the Calculus of Variations approach on a slightly different cost function:

$$J = \frac{1}{2}\int_0^{\infty} \left\{\boldsymbol{x}^T C^T Q_1 C \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u}\right\} dt$$

- Why do this? Because I may only be interested in controlling a *subset* of the states (i.e.., the system outputs)
  - Note here we define the system outputs via the linear output equation,

$$\boldsymbol{y} = C\boldsymbol{x}$$

  and re-define the weighting matrix Q_1 to apply weighting to the outputs contained in output vector $\boldsymbol{y}$

- Using the Calculus of Variations:

$$H = \frac{1}{2}\left\{\boldsymbol{x}^T C^T Q_1 C\boldsymbol{x} + \boldsymbol{u}^T R\boldsymbol{u}\right\} + \boldsymbol{\lambda}^T\left(A\boldsymbol{x} + B\boldsymbol{u}\right)$$

$$\dot{\boldsymbol{\lambda}}^T = -\frac{\partial H}{\partial \boldsymbol{x}} \;\Rightarrow$$

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$$

$$\dot{\boldsymbol{\lambda}} = -A^T\boldsymbol{\lambda} - C^T Q_1 C\boldsymbol{x}$$

$$\frac{\partial H}{\partial \boldsymbol{u}} = 0 \;\Rightarrow\; \boldsymbol{u} = -R^{-1}B^T\boldsymbol{\lambda} \;\Rightarrow\; \dot{\boldsymbol{x}} = A\boldsymbol{x} - BR^{-1}B^T\boldsymbol{\lambda}$$

- So, the optimal solution can be obtained by solving a set of homogeneous, first-order, linear differential equations as was done earlier, but here with a different interpretation:

$$\dot{\boldsymbol{z}} = \mathcal{H}\boldsymbol{z}$$

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} \qquad \mathcal{H} = \left[\begin{array}{c|c} A & -BR^{-1}B^T \\ \hline -C^T Q_1 C & -A^T \end{array}\right]$$

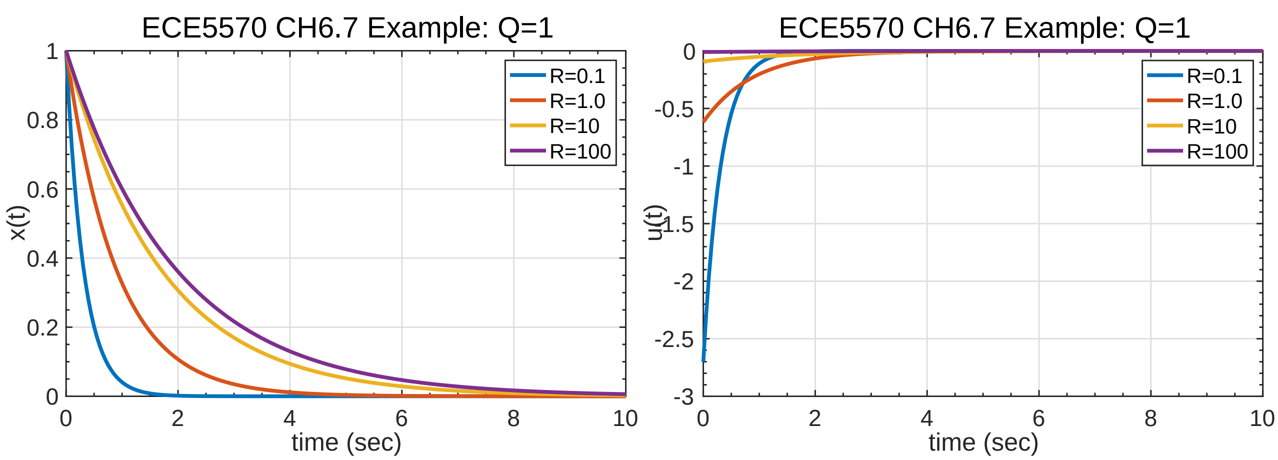
- This result is especially useful because it can be shown that the $2n$ eigenvalues of $\mathcal{H}$ are *symmetric* about both the imaginary the real axis
  - It turns out, this adjoined system has n *stable* roots and n *unstable* roots
  - n stable roots are associated with the state vector $\boldsymbol{x}$ and n unstable roots are associated with the co-state vector $\boldsymbol{\lambda}$
  - For J to remain finite as $t \to \infty$, $\boldsymbol{x}$ must approach zero
    $\Rightarrow$ The n stable eigenvalues of $\mathcal{H}$ must be the *closed-loop poles* of the system

– NOTE: When we optimize a controllable linear system using a quadratic cost, we will *always* generate a stable, closed-loop system!

- Having developed this result, we can now use it to generate the optimal full-state feedback gains.
- Introducing numbers to our example,

$$x_0 = 1$$
$$Q = 1$$

we present the optimal solution for R-values ranging from 0.1 to 100.



6.8: Single-Input-Single-Output Systems: Symmetric Root Locus

- In what we have developed so far, we are assuming that *all* state variables are available for feedback

  $\Rightarrow$ We have n control gains to select – one for each state.

- In general, if we know the optimal pole locations (by looking at the eigenvalues of $\mathcal{H}$), we can calculate the closed-loop characteristic equation and equate coefficients to identify the optimal control gains

- For SISO systems, there is an easier way to find the optimal pole locations than finding the eigenvalues of $\mathcal{H}$

- Beginning from the time-domain state and co-state equations, we apply Laplace transforms to write,

$$\left.\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\ \dot{\boldsymbol{\lambda}} &= -A^T\boldsymbol{\lambda} - C^TQ_1C\boldsymbol{x} \end{aligned}\right\} \Rightarrow \begin{aligned} \boldsymbol{x} &= (sI - A)^{-1}B\boldsymbol{u} \\ \boldsymbol{\lambda} &= -(sI - A)^{-1}C^TQ_1C\boldsymbol{x} \end{aligned}$$

  - From the stationarity condition we have

$$R\boldsymbol{u} + B^T\boldsymbol{\lambda} = 0$$

    and upon introducing from the Laplace transform variables above we can write

$$R\boldsymbol{u} + C^T\left(-sI - A^T\right)^{-1}C^TQ_1C\,(sI - A)^{-1}C\boldsymbol{u} = 0$$

- Now since $\boldsymbol{y} = C\boldsymbol{x}$, we have in Laplace transormed variables,

$$Y(s) = C^T(sI - A)^{-1}B \cdot \boldsymbol{U}(s)$$

- From matrix transpose properties, $\left(-sI - A^T\right)^{-1} = (-sI - A)^{-T}$

- So, if we define

$$G(s) = C^T(sI - A)^{-1}B$$

we get

$$\left[R + G^T(-s)Q_1G(s)\right]\boldsymbol{U}(s) = 0$$

- For a scalar, non-zero value of $u(t)$, $R + G^T(-s)Q_1G(s)$ is a scalar $2n^{th}$ -order polynomial that is *symmetric* in s and $-s$ and must equal zero
  - This polynomial is, in fact, the characteristic equation for the Euler-Lagrange equations developed for this problem
  - So, this polynomial can be used to identify the *optimal closed-loop poles* of the system
- For SISO systems, R and Q_1 are scalars, thus

$$R + G^T(-s)Q_1G(s) = 0 \;\Rightarrow\; 1 + G^T(-s)\frac{Q_1}{R}G(s) = 0$$

which is in ROOT LOCUS FORM (with Q_1/R as the variable gain)

- Root Locus is a technique used primarily in control systems to indicate graphically the locations of roots of a polynomial as a constant 'gain' value is varied.
  - So root locus techniques can be used here to find the optimal closed-loop poles for given ratio Q_1/R
  - The optimal steady-state control gains can then be found by equating coefficients in the closed-loop characteristic equation
  - For SISO systems, this process should be much easier than finding the stable eigenvalues of $\mathcal{H}$
- This technique is termed a *symmetric* root locus because the function $G(s)\,G(-s)$ gives rise to doubly-symmetric pole patterns in the complex plane.

Example 6.3:

$$
\begin{aligned}
x &= -by + u \\
y &= x
\end{aligned}
$$

- In this example, our state matrices are

$$
\begin{aligned}
A &= -b \\
B &= 1 \\
C &= 1
\end{aligned}
$$

- So, setting $Q_1 = q$ and recognizing that $y = Cx$, we can express the cost function in terms of the system output y as,

$$J = \frac{1}{2}\int_0^\infty \left(qy^2 + ru^2\right) dt$$

- Furthermore, we construct the system transfer function $G(s)$ per the foregoing development to write,

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

$$\Rightarrow \qquad 1 + \frac{1}{-s+b} \cdot \frac{q}{r} \cdot \frac{1}{s+b} = 0$$

$$\frac{q}{r}\left\{\frac{1}{(s-b)(s+b)}\right\} = 1$$

- Plotting the root locus gives us the closed-loop root locations as the factor q/r varies from $0 \to \infty$

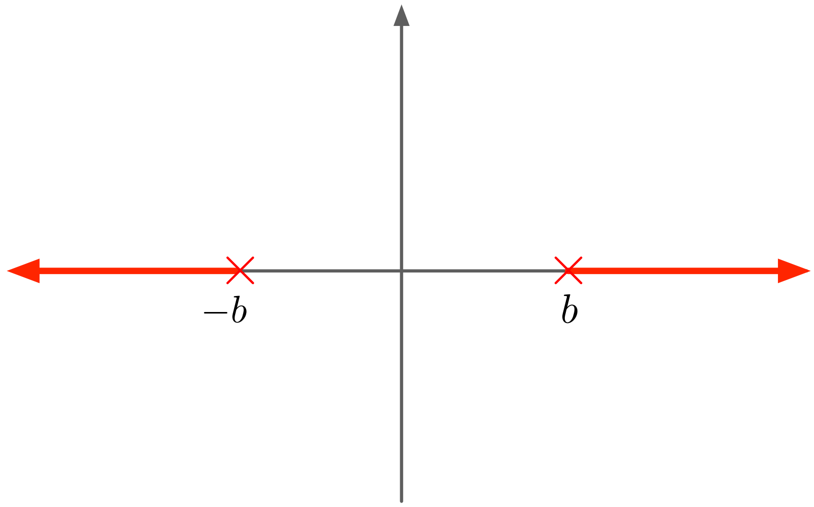


Figure 6.1 Root Locus Plot

- As expected, the root locus is symmetric about both real and imaginary axes; the optimal closed-loop system is described by the stable root in the left-half of the complex plane.
- Can we identify precisely where the closed-loop pole is located?
  - By constructing the characteristic polynomial, we solve analytically for its roots:

$$s^2 - b^2 - \frac{q}{r} = 0$$

$$s = -\sqrt{b^2 + \frac{q}{r}}$$

  - What is the corresponding optimal feedback gain, k?

$$u = -kx \;\Rightarrow\; s + b + k = 0$$

$$-b - k = -\sqrt{b^2 + \frac{q}{r}}$$

$$k = \sqrt{b^2 + \frac{q}{r}} - b$$

- COMMENT: In the case where $q = 0$ (i.e., no weight on the states), J will only be a function of the control $\Rightarrow$ We are only trying to minimize the control effort
  - $b > 0 \Rightarrow$ open-loop system is *stable* $\Rightarrow$ *no need to use control* and $k = 0$
  - $b < 0 \Rightarrow$ open-loop system is unstable $\Rightarrow$ f*inite control is required* to stabliize the system and $k = 2|b|$
- COMMENT: What if we increase q ?
  - Interpretation: we are trying to tighten our control on the system
  - So the system should become more stable (exactly what the root locus demonstrates)

Example 6.4

- Let's take a look at another symmetric root locus example, this one with two states:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u$$

$$y = x_1$$

- For this example, we define

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

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

- Generating the system transfer function,

$$G(s) = C(sI - A)^{-1}B$$

$$= \frac{\omega^2}{s^2 + \omega^2}$$

- Formulating the root locus form $1 + \left(\frac{q}{r}\right) G(s)\, G(-s) = 0$,

$$\Rightarrow \quad 1 + \frac{\omega^2}{(-s)^2 + \omega^2} \cdot \frac{q}{r} \cdot \frac{\omega^2}{s^2 + \omega^2} = 0$$

$$\frac{q}{r} \cdot \frac{\omega^4}{(s^2 + \omega^2)^2} = -1$$

- We may use this formulation to construct the symmetric root locus.

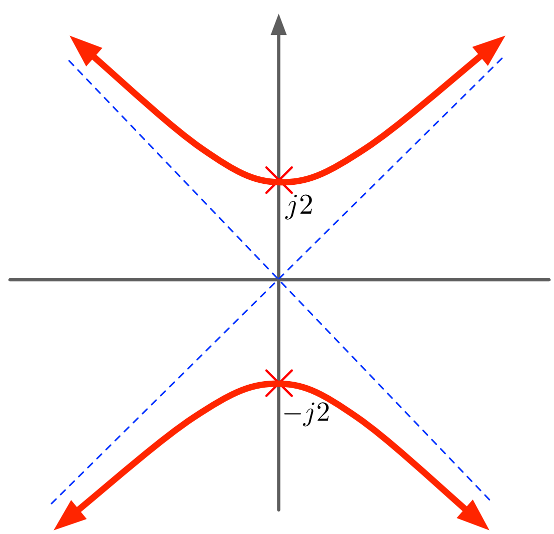


Figure 6.2 Root Locus Plot

CHARACTERISTIC EQUATIONS:

$$\left(s^2 + \omega^2\right)^2 + \frac{q}{r}\omega^4 = 0 \;\Rightarrow\; s^4 + 2\omega^2 s^2 + \omega^4\left(1 + \frac{q}{r}\right) = 0$$

$$u = -k_1 x_1 - k_2 x_2 \;\Rightarrow\; \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2(1 + k_1) & -\omega^2 k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$|sI - A|\,|-sI - A| = s^4 + \omega^2\left(2 + 2k_1 - \omega^2 k_2\right)s^2 + \omega^4\left(1 + k_1\right)^2 = 0$$

- Here we see that the optimal closed-loop poles appear as a complex conjugate pair that travel out into the left-half plane along symmetric asymptotes.
- For a specific set of optimal pole locations, we simply equate coefficients in this last equation with the characteristic polynomial corresponding to the desired pole locations obtained upon specific choice of q/r.

6.9: Multiple-Input-Multiple-Output Systems: Eigenvector Analysis

- For MIMO systems, MacFarlane and Potter developed an elegant eigenvector approach to identify the optimal steady-state feedback control gains
- Since $\boldsymbol{u} = -R^{-1}B^T\boldsymbol{\lambda}$,

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \left[\begin{array}{c|c} A & -BR^{-1}B^T \\ \hline -C^TQ_1C & -A^T \end{array}\right] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix}$$

  - and using the eigenvectors of $\mathcal{H}$, we can write

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = W \left[\begin{array}{c|c} \mathcal{S}_+ & 0 \\ \hline 0 & \mathcal{S}_- \end{array}\right] W^{-1} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix}$$

  where $\mathcal{S}_+ = \mathrm{diag}\{+s_i\}$ (stable eigenvalues) and $\mathcal{S}_- = \mathrm{diag}\{-s_i\}$ (unstable eigenvalues), and the columns of W are the eigenvectors of $\mathcal{H}$

- Now, let's define a new set of transformed states:

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_+ \\ \boldsymbol{z}_- \end{bmatrix} = W_{-1} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix}$$

$$\Rightarrow \quad \dot{\boldsymbol{z}}_+ = \mathcal{S}_+\boldsymbol{z}_+ \qquad \dot{\boldsymbol{z}}_- = \mathcal{S}_-\boldsymbol{z}_-$$

$$\Rightarrow \quad \begin{bmatrix} \boldsymbol{z}_+(t) \\ \boldsymbol{z}_-(t) \end{bmatrix} = \left[\begin{array}{c|c} e^{-\mathcal{S}_+(t_f-t)} & 0 \\ \hline 0 & e^{-\mathcal{S}_-(t_f-t)} \end{array}\right] \begin{bmatrix} \boldsymbol{z}_+(t_f) \\ \boldsymbol{z}_-(t_f) \end{bmatrix}$$

- Based on this solution, we can now solve for $\boldsymbol{x}(t)$ and $\boldsymbol{\lambda}(t)$

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = T\boldsymbol{z}(t) = \left[\begin{array}{c|c} \mathcal{X}_+ & \mathcal{X}_- \\ \hline \Lambda_+ & \Lambda_- \end{array}\right] \left[\begin{array}{c|c} e^{-\mathcal{S}_+(t_f-t)} & 0 \\ \hline 0 & e^{-\mathcal{S}_-(t_f-t)} \end{array}\right] \begin{bmatrix} \boldsymbol{z}_+(t_f) \\ \boldsymbol{z}_-(t_f) \end{bmatrix}$$

– Here, $[\mathcal{X}_+,\ \Lambda_+]$ and $[\mathcal{X}_-,\ \Lambda_-]$ represent the eigenvectors associated with $\mathcal{S}_+$and $\mathcal{S}_-$, respectively

$$\boldsymbol{x}(t) = \mathcal{X}_+ e^{-\mathcal{S}_+(t_f - t)} \boldsymbol{z}_+(t_f) + \mathcal{X}_- e^{-\mathcal{S}_-(t_f - t)} \boldsymbol{z}_-(t_f)$$

$$\boldsymbol{\lambda}(t) = \Lambda_+ e^{-\mathcal{S}_+(t_f - t)} \boldsymbol{z}_+(t_f) + \Lambda_- e^{-\mathcal{S}_-(t_f - t)} \boldsymbol{z}_-(t_f)$$

– Furthermore, as $t_f - t \to \infty$, $\ e^{-\mathcal{S}_+(t_f - t)} \to 0$

$$\boldsymbol{x}(t) = \mathcal{X}_- e^{-\mathcal{S}_-(t_f - t)} \boldsymbol{z}_-(t_f)$$

$$\boldsymbol{\lambda}(t) = \Lambda_- e^{-\mathcal{S}_-(t_f - t)} \boldsymbol{z}_-(t_f)$$

$$\Rightarrow \quad \boldsymbol{\lambda}(t) = \Lambda_- \mathcal{X}_-^{-1} \boldsymbol{x}(t)$$

$$S_0 = \Lambda_- \mathcal{X}_-^{-1} \ \Rightarrow\ \boldsymbol{u}(t) = -R^{-1} B^T \Lambda_- \mathcal{X}_-^{-1} \boldsymbol{x}(t)$$

- NOTE: For MIMO systems, this algorithm is significantly less computationally expensive than integrating the matrix Ricatti equation
- So far, we've discussed the application of optimal feedback control to systems with "soft" terminal constraints to perform the task of regulation (i.e., keeping the states "close" to zero)
- The next section highlights two alternative control problems:

  1. Zero Terminal Error
  2. Tracking

6.10: Zero Terminal Error Controller

- Consider the following linear quadratic control problem:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}, \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0$$

$$J = \frac{1}{2}\int_{t_0}^{t_f} \left\{\boldsymbol{x}^T Q\boldsymbol{x} + \boldsymbol{u}^T R\boldsymbol{u}\right\} dt$$

$$x_i(t_f) = c_i, \qquad (i = 1, 2, \ldots, q)$$

  - NOTE: A soft constraint on the other $n-q$ states at t_f could be used, and the following developments could be modified to handle this added complexity. But for now, we'll assume $\varphi = 0$.

- Standard Calculus of Variations approach:

$$\Phi = \sum_{i=1}^{q} \nu_i \left\{x_i(t_f) - c_i\right\} \qquad H = \frac{1}{2}\left\{\boldsymbol{x}^T Q\boldsymbol{x} + \boldsymbol{u}^T R\boldsymbol{u}\right\} + \boldsymbol{\lambda}^T\left\{A\boldsymbol{x} + B\boldsymbol{u}\right\}$$

$$\dot{\boldsymbol{\lambda}} = -Q\boldsymbol{x} - A^T\boldsymbol{\lambda}, \qquad \lambda_i(t_f) = \begin{cases} \nu_i, & i = 1, 2, \ldots, q \\ 0, & i = q+1, \ldots, n \end{cases}$$

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}, \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0$$

$$x_i(t_f) = c_i, \quad i = 1, 2, \ldots, q$$

$$\boldsymbol{u} = -R^{-1}B^T\boldsymbol{\lambda}$$

- The solution is obtained by solving the following equations:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix}$$

$$x_i(t_f) = c_i, \quad i = 1, 2, \ldots, q$$

$$\lambda_i(t_f) = 0, \quad i = q+1, q+2, \ldots, n$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x_0}$$

– For relatively simple problems, these equations can be solved using standard linear systems analysis methods

– For more difficult problems, the so-called “sweep” method provides a practical solution alternative.

6.11: Tracking Controller

- In this type of problem, we want to develop an optimal control law that will force the plant to track a desired *reference trajectory*, $\boldsymbol{r}(t)$, over a specified time interval

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$$
$$\boldsymbol{y} = C\boldsymbol{x}$$

$$J = \frac{1}{2}\left\{C\boldsymbol{x}(t_f) - \boldsymbol{r}(t_f)\right\}^T P_f \left\{C\boldsymbol{x}(t_f) - \boldsymbol{r}(t_f)\right\} + \frac{1}{2}\int_{t_0}^{t_f}\left\{(C\boldsymbol{x} - \boldsymbol{r})^T Q' (C\boldsymbol{x} - \boldsymbol{r}) + \boldsymbol{u}^T R\boldsymbol{u}\right\} dt$$

- Note here that the running cost penalizes the deviation of the output variable $\boldsymbol{y}(t) = C\boldsymbol{x}(t)$ from a time-varying reference trajectory $\boldsymbol{r}(t)$
- Likewise, the terminal cost penalizes the distance of the output variable $\boldsymbol{y}\left(t_f\right) = C\boldsymbol{x}\left(t_f\right)$ from a terminal reference point $r\left(t_f\right)$
- The Hamiltonian for this problem is written,

$$H = \frac{1}{2}\left\{(C\boldsymbol{x} - \boldsymbol{r})^T Q' (C\boldsymbol{x} - \boldsymbol{r}) + \boldsymbol{u}^T R\boldsymbol{u}\right\} + \boldsymbol{\lambda}^T (A\boldsymbol{x} + B\boldsymbol{u})$$

which gives the optimality equations,

$$\dot{\boldsymbol{\lambda}} = \left(-\frac{\partial H}{\partial \boldsymbol{x}}\right)^T = -C^T Q' C\boldsymbol{x} - A^T\boldsymbol{\lambda} + C^T Q'\boldsymbol{r}$$

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} = A\boldsymbol{x} - BR^{-1}B^T\boldsymbol{\lambda}$$
$$\boldsymbol{u} = -R^{-1}B^T\boldsymbol{\lambda}$$

- As before, the state and co-state equations can be stacked in augmented form to write,

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -C^TQ'C & -A^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ C^TQ' \end{bmatrix} \boldsymbol{r}$$

- We must account for the modifications introduced by the $\boldsymbol{r}$ term however
- For this development, we shall assume the input function is of the form,

$$\boldsymbol{u}(t) = -K(t)\,\boldsymbol{x}(t) + \boldsymbol{w}(t)$$

where

$$K(t) = -R^{-1}B^TP(t)$$

- We shall assume $w(t)$ is of similar form, i.e.,

$$\boldsymbol{w}(t) = -R^{-1}B^T\boldsymbol{v}(t)$$

- Therefore, since we know that

$$\boldsymbol{u}(t) = -R^{-1}B^T\boldsymbol{\lambda}(t),$$

then it's clear that

$$\boldsymbol{\lambda}(t) = P(t)\,\boldsymbol{x}(t) + \boldsymbol{v}(t)$$

- Subsituting these forms into the equation for $\dot{\boldsymbol{\lambda}}(t)$, we can write

$$\dot{\boldsymbol{\lambda}} = \dot{P}\boldsymbol{x} + P\dot{\boldsymbol{x}} + \boldsymbol{v} = -A^T\boldsymbol{\lambda} - C^TQ'C\boldsymbol{x} + C^TQ'\boldsymbol{r}$$

and thus,

$$\dot{P}\boldsymbol{x} = -PA\boldsymbol{x} - PBR^{-1}B^TP\boldsymbol{x} - PBR^{-1}B^T\boldsymbol{v} - \dot{\boldsymbol{v}}$$

$$-A^T(P\boldsymbol{x} + \boldsymbol{v}) - C^TQ'C\boldsymbol{x} + C^TQ'\boldsymbol{r}$$

- Extracting terms, we have a first-order differential equation in $\boldsymbol{v}(t)$,

$$\dot{\boldsymbol{v}} = \left(-A^T + PBR^{-1}B^T\right)\boldsymbol{v} + C^T Q' \boldsymbol{r}$$

- Therefore, we can construct the total solution by solving first, the matrix Ricatti differential equation for $P(t)$, and then the differential equation for $\boldsymbol{v}(t)$ to obtain the optimal solution,

$$\boldsymbol{u}^*(t) = -R^{-1}B^T\left(P(t)\,x(t) + v(t)\right)$$

Example 6.4: Tracking Controller

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= 2x_1 - x_2 + u \\
y &= x_1
\end{aligned}$$

$$J = \left(y_f - 1\right)^2 + \int_0^{t_f}\left\{(y-1)^2 + 0.0025u^2\right\}dt$$

- The state matrices are given by,

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

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$Q' = 2 \quad R = .005 \quad P_f = 2 \quad r(t) = 1$$

$$C^T Q' = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \qquad C^T Q' C = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\dot{P} = -PA - A^T P - C^T Q' C + PBR^{-1}B^T P$$

- Taking the terms of $P(t)$ one-at-a-time,

$$
\begin{aligned}
\dot{p}_{11} &= 2\left(100p_{12}^2 - 2p_{12} - 1\right) \\
\dot{p}_{12} = \dot{p}_{21} &= 200p_{12}p + p_{12} - p_{11} - 2p_{22} \\
\dot{p}_{22} &= 200p_{22}^2 + 2\left(p - p_{12}\right)
\end{aligned}
$$

where

$$
P\left(t\right) = \begin{bmatrix} p_{11}\left(t\right) & p_{12}\left(t\right) \\ p_{12}\left(t\right) & p_{22}\left(t\right) \end{bmatrix}
$$

$$
P(t_f) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}
$$

- It is now possible to use an ordinary differential equation solver (e.g., ode45 in MATLAB) to compute the solutions for $P\left(t\right)$, integrating backward from $P\left(t_f\right)$ and for $\boldsymbol{v}\left(t\right)$, integrating backward from $\boldsymbol{v}\left(t_f\right)$

$$
\dot{\boldsymbol{v}} = \left(-A^T + PBR^{-1}B^T\right)\boldsymbol{v} + C^TQ'\boldsymbol{r}
$$

$$
\begin{aligned}
\dot{v}_1 &= \left(200p_{12} - 2\right)v_2 + 2 \\
\dot{v}_2 &= \left(200p_{22} + 1\right)v_2 - v_1
\end{aligned}
$$

$$
\boldsymbol{v}(t_f) = \begin{bmatrix} -2 \\ 0 \end{bmatrix}
$$

$$
\begin{aligned}
u^*\left(t\right) &= -R^{-1}B^T\left(P\left(t\right)x\left(t\right) + v\left(t\right)\right) \\
&= -200\left\{p_{12}x_1 + p_{22}x_2 + v_2\right\}
\end{aligned}
$$