Control Engineering SC42095

Linear quadratic control

Tamás Keviczky, Azita Dabiri
Delft Center for Systems and Control
Delft University of Technology

Lecture outline

1. Linear quadratic control
    - deterministic case
    - completing the squares
    - dynamic programming
    - example
    - stochastic case
2. Kalman filtering
3. Linear quadratic Gaussian control

Note: These slides are partly inspired by the slides for this course developed at the Department of Automatic Control, Lund Institute of Technology (see http://www.control.lth.se/˜kursdr)

Lecture outline (continued)

- Previous control methods: PID, pole placement
  $\rightarrow$ focus on SISO
- Now, more general: MIMO + process & measurement noise
  $\rightarrow$ optimization-based approach
- First we consider case with *full state information*
  = Linear Quadratic (LQ) control

Next lecture:

- Estimating state from measurements of noisy output
  = Kalman filtering
- Finally, combination based on *separation theorem*
  = Linear Quadratic Gaussian (LQG) control

1. Linear quadratic control

1.1 LQ control: Deterministic case

- *Discrete-time* LTI state space model:

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = Cx(k)$$

- We start at $k = 0$ with $x(0) = x_0$ and look N steps into the future, and determine optimal control sequence $u(0), \ldots, u(N-1)$

- Performance criterion (“loss function”) J over period $[0, N]$

  Inspired by “power” of the state: $\displaystyle\int_0^T \|x(t)\|^2 dt$

  or in discrete time: $\displaystyle\sum_{k=0}^{N} x^{\mathsf{T}}(k)x(k)$

1.1 LQ control: Deterministic case (continued)

- Components of state might have different units
  $\rightarrow$ weighting

$$J = \sum_{k=0}^{N} x^{\mathsf{T}}(k) Q_1 x(k)$$

  Control signal u can be penalized in similar way

- So in general:

$$J = \sum_{k=0}^{N-1} \Big(x^{\mathsf{T}}(k) Q_1 x(k) + 2x^{\mathsf{T}}(k) Q_{12} u(k) + u^{\mathsf{T}}(k) Q_2 u(k) \Big) + x^{\mathsf{T}}(N) Q_0 x(N)$$

  with Q-matrices symmetric and positive semi-definite
  i.e., $v^{\mathsf{T}} Q v \geqslant 0$ for all v

  Note: Separate penalty term for final state

Deterministic LQ: Problem formulation

- Minimize

  $$J = \sum_{k=0}^{N-1} \Big(x^{\mathsf{T}}(k) Q_1 x(k) + 2x^{\mathsf{T}}(k) Q_{12} u(k) + u^{\mathsf{T}}(k) Q_2 u(k) \Big) + x^{\mathsf{T}}(N) Q_0 x(N)$$

  subject to $x(k+1) = \Phi x(k) + \Gamma u(k)$ and $x(0) = x_0$

  Note: For simplicity of notation we sometimes drop argument k for x and u in the sequel

- Solution approach based on quadratic optimization and dynamic programming

1.2 Completing the squares

- Scalar case: Find u that minimizes $ax^2 + 2bxu + cu^2$ with $c > 0$
- Use derivative or completing the squares:

$$\begin{aligned} ax^2 + 2bxu + cu^2 &= ax^2 + c\left(2\frac{b}{c}xu + u^2\right) \\ &= ax^2 + c\left(u^2 + 2\frac{b}{c}xu + \frac{b^2}{c^2}x^2 - \frac{b^2}{c^2}x^2\right) \\ &= \left(a - \frac{b^2}{c}\right)x^2 + c\left(u + \frac{b}{c}x\right)^2 \end{aligned}$$

- First term independent of u, second term always nonnegative

  So minimum is reached for $u = -\frac{b}{c}x$, and minimum value is $\left(a - \frac{b^2}{c}\right)x^2$

1.2 Completing the squares (continued)

- Matrix case: Find u that minimizes $x^\mathsf{T} Q_x x + 2x^\mathsf{T} Q_{xu} u + u^\mathsf{T} Q_u u$ with Q_u positive definite
- Let L be such that $Q_u L = Q_{xu}^\mathsf{T}$. Then

$$x^\mathsf{T} Q_x x + 2x^\mathsf{T} Q_{xu} u + u^\mathsf{T} Q_u u =$$

$$x^\mathsf{T}(Q_x - L^\mathsf{T} Q_u L)x + (u + Lx)^\mathsf{T} Q_u (u + Lx)$$

  is minimized for $u = -Lx$

  and minimum value is $x^\mathsf{T}(Q_x - L^\mathsf{T} Q_u L)x$
- If Q_u is positive definite, then $L = Q_u^{-1} Q_{xu}^\mathsf{T}$

1.3 LQ control via least-squares

- The LQ control problem can be formulated (and solved) as a (large) least-squares problem.
- Note that $X = (x(0), \ldots, x(N))$ is a *linear function* of $x(0)$ and $U = (u(0), \ldots, u(N-1))$:

$$\underbrace{\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} \Gamma & 0 & \cdots & 0 \\ \Phi\Gamma & \Gamma & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Phi^{N-1}\Gamma & \Phi^{N-2}\Gamma & \cdots & \Gamma \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}}_{U} + \underbrace{\begin{bmatrix} \Phi \\ \Phi^2 \\ \vdots \\ \Phi^N \end{bmatrix}}_{H} x(0)$$

1.3 LQ control via least-squares (continued)

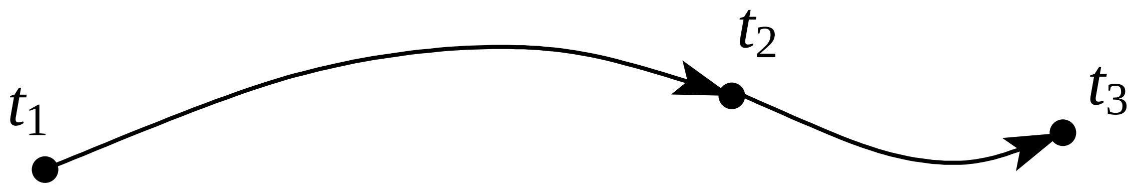
- Thus (assuming $Q_{12} = 0$ for simplicity) we can express the LQ cost function as

$$J(U) = \left\| \mathbf{diag}\left(Q_1^{1/2}, \ldots, Q_1^{1/2}, Q_0^{1/2}\right)(GU + Hx(0)) \right\|^2 + \left\| \mathbf{diag}\left(Q_2^{1/2}, \ldots, Q_2^{1/2}\right) U \right\|^2$$

- This is just a (big) least-squares problem!
- This solution method requires forming and solving a least-squares problem with size that grows with N!

1.4 Dynamic programming

- **Principle of optimality**: From any point on optimal trajectory, remaining trajectory is also optimal



$\rightarrow$ allows to determine best control law over period $[t_2, t_3]$ independent of how state at t_2 was reached

- Useful idea, has many applications beyond LQ control, e.g.
  - optimal flow control in communication networks
  - optimization in finance

LQ control: Dynamic programming solution

- Gives an efficient, recursive method to solve LQ control least-squares problem
- Define the value function J_k as

$$J_k(z) = \min_{u(k),\ldots,u(N-1)} \sum_{l=k}^{N-1} \left(x^{\mathsf{T}} Q_1 x + 2x^{\mathsf{T}} Q_{12} u + u^{\mathsf{T}} Q_2 u \right) + x^{\mathsf{T}}(N) Q_0 x(N)$$

  subject to $x(k) = z$ and $x(l+1) = \Phi x(l) + \Gamma u(l)$.
- We will write x and u instead of $x(l)$ and $u(l)$ for simplicity.
- $J_k(z)$ gives the minimum LQ cost-to-go, starting from state z at time k.
- $J_0(z)$ is the minimum LQ cost (from state $x(0)$ at time 0).

LQ control: Dynamic programming solution (continued)

We will find that

- J_k is quadratic, i.e., $J_k(z) = z^\mathsf{T} S(k) z$, where $S(k) = S(k)^\mathsf{T} \geq 0$
- $S(k)$ can be found recursively, working backwards from $k = N$
- the LQ optimal control u is easily expressed in terms of $S(k)$

LQ control: Dynamic programming solution (continued)

- Cost-to-go with no time left is just the final state cost:

$$J_N(z) = z^T Q_0 z$$

  thus we have $S(N) = Q_0$.

- Now suppose we know J_{k+1}, what is the optimal choice for $u(k)$?
- Choice of $u(k)$ affects
  - current cost incurred (through $u(k)^T Q_2 u(k)$)
  - where we end up, i.e., the min-cost-to-go from $x(k+1)$

Dynamic programming principle

$$J_k(x) = x^\mathsf{T} Q_1 x + \min_u \left(u^\mathsf{T} Q_2 u + J_{k+1}(\Phi x + \Gamma u) \right)$$

- Called DP, Bellman, or Hamilton-Jacobi equation
- Gives J_k recursively, in terms of J_{k+1}
- Any minimizing u gives optimal control action
- For N-step problem:
  - start from end at time $k = N$
  - now we can determine best control law for last step independent of how state at time $N-1$ was reached
  - iterate backward in time to initial time $k = 0$

Dynamic programming solution

- Define $S(k)$ as

$$J_k(x) = x^{\mathsf{T}} S(k) x$$

- Principle of optimality with $t_1 = k$, $t_2 = k+1$, $t_3 = N$ gives

$$\begin{aligned} x^{\mathsf{T}} S(k) x &= \min_u \; x^{\mathsf{T}} Q_1 x + 2x^{\mathsf{T}} Q_{12} u + u^{\mathsf{T}} Q_2 u + x^{\mathsf{T}}(k+1) S(k+1) x(k+1) \\ &= \min_u \; x^{\mathsf{T}} Q_1 x + 2x^{\mathsf{T}} Q_{12} u + u^{\mathsf{T}} Q_2 u + \\ &\qquad (\Phi x + \Gamma u)^{\mathsf{T}} S(k+1)(\Phi x + \Gamma u) \end{aligned}$$

Dynamic programming solution (continued)

- So

$$x^{\mathsf{T}}S(k)x = \min_u x^{\mathsf{T}} \underbrace{\left(Q_1 + \Phi^{\mathsf{T}}S(k+1)\Phi\right)}_{Q_x} x + 2x^{\mathsf{T}} \underbrace{\left(Q_{12} + \Phi^{\mathsf{T}}S(k+1)\Gamma\right)}_{Q_{xu}} u$$
$$+ u^{\mathsf{T}} \underbrace{\left(Q_2 + \Gamma^{\mathsf{T}}S(k+1)\Gamma\right)}_{Q_u} u$$

- Completing-of-squares solution then gives:

$$u(k) = -L(k)x(k)$$
$$L(k) = Q_u^{-1}Q_{xu}^{\mathsf{T}} = (Q_2 + \Gamma^{\mathsf{T}}S(k+1)\Gamma)^{-1}\left(Q_{12} + \Phi^{\mathsf{T}}S(k+1)\Gamma\right)^{\mathsf{T}}$$
$$S(k) = Q_x - L^{\mathsf{T}}Q_uL = Q_1 + \Phi^{\mathsf{T}}S(k+1)\Phi - L^{\mathsf{T}}(Q_2 + \Gamma^{\mathsf{T}}S(k+1)\Gamma)L$$

$\rightarrow$ Discrete-time Riccati recursion

Dynamic programming solution (continued)

- Discrete-time Riccati recursion:

$$S(k) = Q_1 + \Phi^{\mathsf{T}} S(k+1)\Phi$$
$$- \left(Q_{12} + \Phi^{\mathsf{T}} S(k+1)\Gamma\right)\left(Q_2 + \Gamma^{\mathsf{T}} S(k+1)\Gamma\right)^{-1}\left(Q_{12}^{\mathsf{T}} + \Gamma^{\mathsf{T}} S(k+1)\Phi\right)$$

- $S(N)$ determined via

$$x^{\mathsf{T}}(N)S(N)x(N) := x^{\mathsf{T}}(N)Q_0 x(N)$$

  So $S(N) = Q_0$

- Minimum of loss function over period $[0,N]$ is given by

$$\min J = x^{\mathsf{T}}(0)S(0)x(0)$$

Dynamic programming solution (continued)

- Solution of LQ problem is time-varying controller

$$u(k) = -L(k)x(k)$$

- Feedback matrix $L(k)$ does not depend on x and can be pre-computed for $k = N, N-1, \ldots, 0$ and stored on computer

- In practice only stationary controller is used, i.e., constant controller obtained when Riccati recursion is iterated until constant $S(k) = \bar{S}$ is obtained:

$$\bar{S} = Q_1 + \Phi^{\mathsf{T}}\bar{S}\Phi - \left(Q_{12} + \Phi^{\mathsf{T}}\bar{S}\Gamma\right)^{\mathsf{T}}\left(Q_2 + \Gamma^{\mathsf{T}}\bar{S}\Gamma\right)^{-1}\left(Q_{12}^{\mathsf{T}} + \Gamma^{\mathsf{T}}\bar{S}\Phi\right)$$

  This is called the discrete-time algebraic Riccati equation (ARE). The stationary controller is then:

$$L = \left(Q_2 + \Gamma^{\mathsf{T}}\bar{S}\Gamma\right)^{-1}\left(Q_{12} + \Phi^{\mathsf{T}}\bar{S}\Gamma\right)^{\mathsf{T}}$$

  Note: $\bar{S}$ does not depend on final state weight matrix Q_0

Properties of the LQ controller

- Consider LTI system and loss function with positive definite Q-matrices.
- If steady-state solution $\bar{S}$ exists and is positive definite, then steady-state control strategy

$$u(k) = -Lx(k)$$

gives asymptotically stable closed-loop system

$$x(k+1) = (\Phi - \Gamma L)x(k)$$

Proof: See book Åström & Wittenmark, pp. 423–424
Based on using $x^{\mathsf{T}}(k)\bar{S}x(k)$ as Lyapunov function

Properties of the LQ controller (continued)

- Discrete-time LQ controller has finite gain margin as opposed to continuous-time LQ controller! (Just like every digital controller.)
- LQ control is readily extended to handle time-varying systems $\Phi(k), \Gamma(k)$, and time-varying cost matrices $Q_1(k), Q_{12}(k), Q_2(k)$. The DP solution idea is the same, but there need not be a steady-state solution.
- Tracking problems can also be considered by replacing the state $x(k)$ and control $u(k)$ terms in the cost function with $x(k) - \bar{x}(k)$ and $u(k) - \bar{u}(k)$. The $\bar{x}(k)$ and $\bar{u}(k)$ are given state and input trajectories.

How to find weighting matrices?

- In principle derived from “natural” loss function
  But not always possible/applicable in practice
  $\rightarrow$ designer chooses/tunes weighting matrices
- So what is the difference from pole placement or direct search over $L(k)$?
  In theory: none
  In practice: LQ preferred since easy to use
- Sometimes one selects $Q_{12} = 0$ and Q_1, Q_2 diagonal with as diagonal entries the inverse value of square of allowed deviations in states and control signals:

$$(Q_{1/2})_{ii} = \frac{1}{(\text{allowed deviation in state/control input } i\,)^2}$$

How to solve LQ problem in matlab?

- Command `[K,S,E] = dlqr(A,B,Q,R,N)`
- Calculates optimal gain matrix `K` such that state-feedback law `u[n] = -Kx[n]` minimizes cost function

  ```
  J = Sum {x'Qx + u'Ru + 2*x'Nu}
  ```

  subject to state dynamics

  ```
  x[n+1] = Ax[n] + Bu[n]
  ```

- Also returns steady-state solution `S` of Riccati equation and closed-loop eigenvalues `E=eig(A-B*K)`

1.5 Example

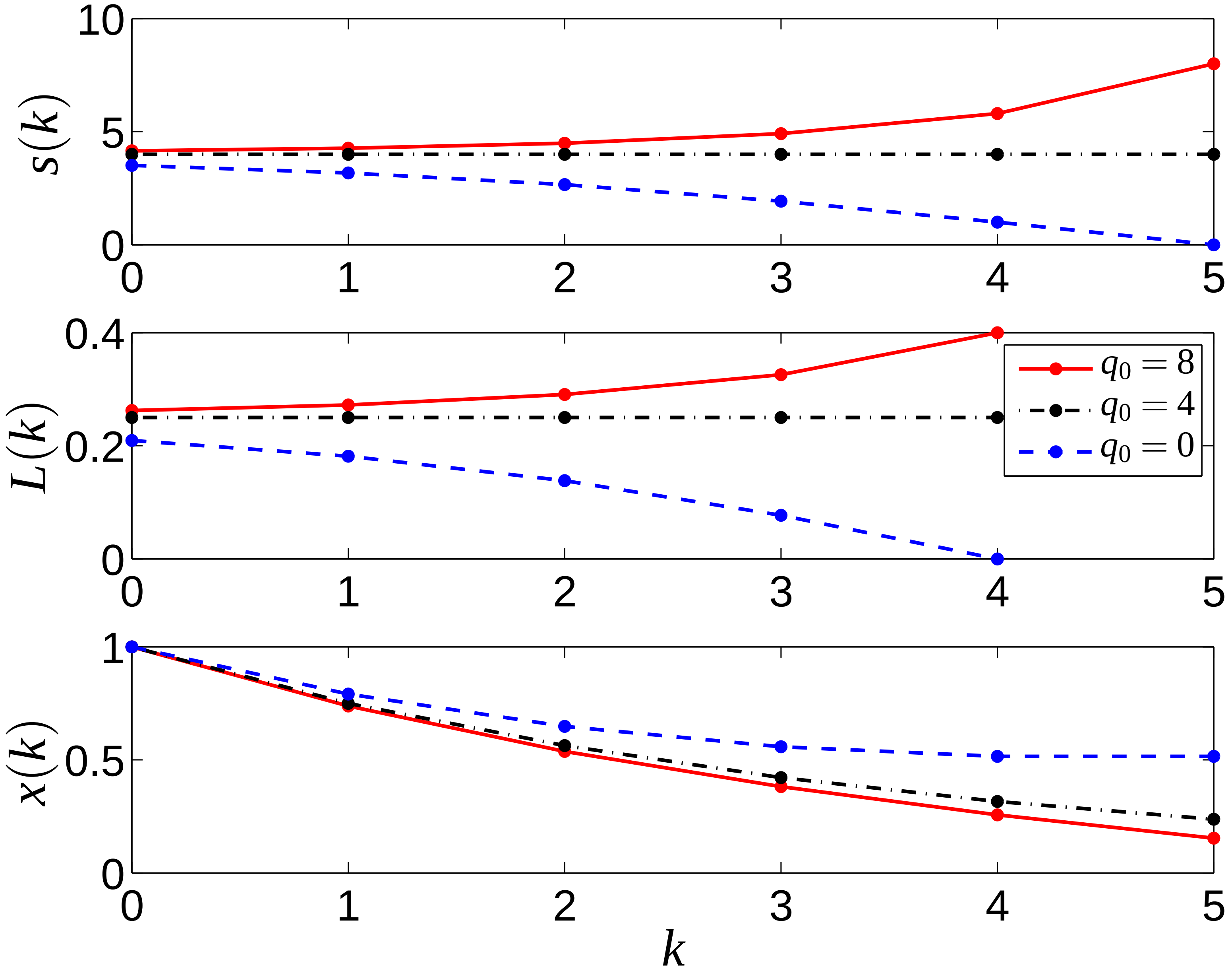
- Discrete-time system $x(k+1) = x(k) + u(k)$
- Loss function $\sum_{k=0}^{N-1}\left(x^2(k) + 12u^2(k)\right) + q_0 x^2(N)$ with $N = 5$
- Riccati equation:

$$s(k) = s(k+1) + 1 - \frac{s^2(k+1)}{s(k+1)+12} \qquad \text{with } s(N) = q_0$$

- Controller: $u(k) = -l(k)x(k) = -\dfrac{s(k+1)}{s(k+1)+12}x(k)$
- Stationary solution given by:

$$\bar{s} = \bar{s} + 1 - \frac{\bar{s}^2}{\bar{s}+12} \quad \rightarrow \quad \bar{s}^2 - \bar{s} - 12 = 0 \quad \rightarrow \quad \bar{s} = 4$$

1.5 Example (continued)



1.6 LQ control: Stochastic case

- Minimize

$$J = \mathsf{E}\left[\sum_{k=0}^{N-1}\left(x^{\mathsf{T}}Q_1x + 2x^{\mathsf{T}}Q_{12}u + u^{\mathsf{T}}Q_2u\right) + x^{\mathsf{T}}(N)Q_0x(N)\right]$$

  subject to $x(k+1) = \Phi x(k) + \Gamma u(k) + v(k)$ and $x(0) = x_0$ with v Gaussian zero-mean white noise process with

$$\mathsf{E}\left[v(k)v^{\mathsf{T}}(k)\right] = R_1$$

  and $x(0)$ has Gaussian distribution with

$$\mathsf{E}\left[x(0)\right] = m_0, \quad \mathrm{cov}(x(0)) = \mathsf{E}\left[(x(0) - m_0)^{\mathsf{T}}(x(0) - m_0)\right] = R_0$$

- Solution: also based on dynamic programming and Riccati equation

1.6 LQ control: Stochastic case (continued)

- Define $S(k)$ by Riccati equation and

$$V_k(x(k)) = \min_{u(k),\ldots,u(N-1)} \mathsf{E}\left[\sum_k^{N-1}\left(x^{\mathsf{T}}Q_1x + 2x^{\mathsf{T}}Q_{12}u + u^{\mathsf{T}}Q_2u\right) + x^{\mathsf{T}}(N)Q_0x(N)\right]$$

- Then with $x = x(N)$

$$V_N(x) = \mathsf{E}\left[x^{\mathsf{T}}Q_0x\right] = \mathsf{E}\left[x^{\mathsf{T}}S(N)x\right]$$

and with $x = x(N-1)$ and $u = u(N-1)$

$$V_{N-1}(x) = \min_u \mathsf{E}\left[(x^{\mathsf{T}}Q_1x + 2x^{\mathsf{T}}Q_{12}u + u^{\mathsf{T}}Q_2u) + \underbrace{x^{\mathsf{T}}(N)Q_0x(N)}_{V_N(x(N))}\right]$$

1.6 LQ control: Stochastic case (continued)

- Using principle of optimality:

$$
\begin{aligned}
&V_{N-1}(x) \\
&= \min_u \mathsf{E}\left[(x^\mathsf{T} Q_1 x + 2x^\mathsf{T} Q_{12} u + u^\mathsf{T} Q_2 u) + V_N(x(N))\right] \\
&= \min_u \mathsf{E}\left[(x^\mathsf{T} Q_1 x + 2x^\mathsf{T} Q_{12} u + u^\mathsf{T} Q_2 u) + (\Phi x + \Gamma u + v)^\mathsf{T} S(N)(\Phi x + \Gamma u + v)\right] \\
&= \min_u \mathsf{E}\left[(x^\mathsf{T} Q_1 x + 2x^\mathsf{T} Q_{12} u + u^\mathsf{T} Q_2 u) + (\Phi x + \Gamma u)^\mathsf{T} S(N)(\Phi x + \Gamma u)\right] \\
&\qquad\qquad + \mathsf{E}\left[v^\mathsf{T} S(N) v\right]
\end{aligned}
$$

  as v is independent of x and u

- Part to be minimized is independent of v, so

$$V_{N-1}(x) = \mathsf{E}\left[x^\mathsf{T} S(N-1) x\right] + \mathsf{E}\left[v^\mathsf{T} S(N) v\right]$$

1.6 LQ control: Stochastic case (continued)

- Results in

$$V_0(x) = \mathsf{E}\left[x^\mathsf{T} S(0) x\right] + \sum_{k=1}^{N-1} \mathsf{E}\left[v^\mathsf{T}(k) S(k+1) v(k)\right]$$

- Solution is then again given by $u(k) = -L(k)x(k)$ with $L(k)$ as defined in deterministic case and $S(k)$ given by Riccati equation

- Minimum cost equal to

$$V_0(x_0) = m_0^\mathsf{T} S(0) m_0 + \mathrm{tr}\left[S(0)R_0\right] + \sum_{k=1}^{N-1} \mathrm{tr}\left[S(k+1)R_1\right]$$

Proof: See book Åström and Wittenmark, pp. 418–419

Summary

- LQ control
  - minimize loss function over horizon of N steps
  - deterministic + stochastic case
  - full state information $\rightarrow$ state feedback controller
  - solution based on quadratic optimization and dynamic programming