EE402 - Discrete Time Systems **Fall 2025**

Lecture 19

Lecturer: Eminalp Koyuncu

Discrete Time Linear Quadratic Regulator

Linear Quadratic Regulator control policy is one of the most used state feedback laws in the literature. The name of the control policy well defines its purpose. That is,

- **Linear** in the sense that it applies to the control of linear systems.
- **Quadratic** in the sense that it solves an optimal control problem with a quadratic cost function.
- **Regulator** in the sense that originally it applies to the regulation class of problems in the field of control theory.

Regulation vs. Tracking Problems

In the control theory, the feedback control can be applied to a system for two different purposes, namely, regulation and tracking. For the regulation problems, there is no reference input and the system tries to cancel any disturbance acting on it. Note that, the initial condition of the system can also be modeled as a disturbance and the regulator is expected to cancel the effect of the initial condition and drive the states into the origin. The performance of a regulator can be quantified by the asymptotic stability properties of the closed loop system. Let us define the discrete time system,

$$x[k+1] = Ax[k] + Bu[k] \tag{19.1a}$$
$$y[k] = \underbrace{I}_{C} x[k] = x[k] \tag{19.1b}$$

with the regulator state feedback law $u[k] = -Ky[k] = -Kx[k]$. So that, we obtain the autonomous system

$$x[k+1] = A_c x[k] \tag{19.2}$$

with the closed loop state transition matrix $A_c = (A - BK)$.

Definition 19.1 *The autonomous system given by (19.2) is asymptotically stable if the following conditions hold.*

- *For each $\epsilon > 0$, there exists $\delta > 0$ such that,*

$$||x[0]|| < \delta \implies ||x[k]|| < \epsilon \quad \textit{for all} \ \ k \geq 0 \tag{19.3}$$

- *δ can be chosen so that,*

$$||x[0]|| < \delta \implies \lim_{k\to\infty} ||x[k]|| = 0 \tag{19.4}$$

The meaning of the asymptotic stability is that for any initial condition $x[0]$, the state should be bounded and it eventually converges to the origin. In fact, for linear systems, it is sufficient to show this property for a single initial condition and the result can be generalizable.

Example: Consider the following closed loop system.

$$x[k+1] = A_c x[k] \tag{19.5}$$

For any initial condition $x[0]$, we can write the system solution as

$$x[k] = A_c^k x[0] \tag{19.6}$$

We know that the norm of the states is $||x[k]|| = x[k]^T x[k] = x[0]^T (A_c^T A_c)^k x[0]$, which is bounded if all the eigenvalues of $A_c^T A_c$ are either on the unit circle or inside the unit circle. In fact, it is the case when the eigenvalues of A_c satisfies the same condition (proof left as an exercise). Moreover, if the eigenvalues are strictly inside the unit circle, then the closed loop system is asymptotically stable.

$$\lim_{k\to\infty} A_c^k x[0] = 0 \implies \lim_{k\to\infty} ||x[k]|| = 0 \tag{19.7}$$

On the other hand, a tracking problem is the case when a nonzero reference trajectory is tried to be tracked autonomously with the feedback control. We can convert this problem into a regularization problem by defining the error state.

$$e[k] = x[k] - x_{ref}[k] \tag{19.8}$$

Next, we define an integrator state to cancel the steady state error for constant reference.

$$z[k+1] = z[k] + e[k] \tag{19.9}$$

Next, we redefine the state and the state transitions in terms of the error and the integrator state. Also, we define the state feedback law.

$$\begin{aligned}
\bar{x}[k] &= \begin{bmatrix} e_k \\ z_k \end{bmatrix} && (19.10a)\\
e[k+1] &= x[k+1] - x_{ref}[k+1] && (19.10b)\\
&= Ax[k] + Bu[k] - x_{ref}[k+1] && (19.10c)\\
&= Ae[k] + Bu_k + (Ax_{ref}[k] - x_{ref}[k+1]) && (19.10d)\\
u[k] &= -\begin{bmatrix} K_e & K_z \end{bmatrix} \bar{x}[k] && (19.10e)\\
\bar{x}[k+1] &= \underbrace{\begin{bmatrix} A - BK_e & -BK_z \\ I & I \end{bmatrix}}_{A_c} \bar{x}[k] + \begin{bmatrix} I \\ 0 \end{bmatrix} \underbrace{(Ax_{ref}[k] - x_{ref}[k+1])}_{d[k]} && (19.10f)
\end{aligned}$$

If the reference is constant, then we have a constant $d[k]$. In this case, if the error state is asymptotically stable, our system can perfectly track the reference input. Note that the stability definition for the reference input that varies over time is not trivial, which is not within the scope of this lecture. However, if the rate of change of the reference input is much slower than the error state dynamics, you can assume constant reference. Also, if the full state information is not available, you can reconstruct the state with an observer like the Luanberger Observer or the Kalman Filter if (A,C) is observable and use the observer output as the state.

Lyapunov Stability for Linear Systems

Before moving on to the LQR problem, we need another tool to establish the asymptotic stability of the closed loop system, which will be used to analyze the performance of LQR control law. This tool will be the Lyapunov's method applied to the linear systems. Note that Lyapunov's method is a vast topic that applies all kinds of systems; however, we will focus on linear systems for this lecture. Before moving on to the theory, lets focus on some useful definitions.

Definition 19.2 *Definiteness of Matrices:* A matrix G is said to be;

- Positive definite (shown as $G > 0$), if it satisfies $x^T Gx > 0$ for all x. This condition is satisfied if all the eigenvalues λ_i are strictly positive, i.e., $\lambda_i > 0 \; \forall i$.
- Positive semi-definite (shown as $G \geq 0$), if it satisfies $x^T Gx \geq 0$ for all x. This condition is satisfied if all the eigenvalues λ_i are non-negative, i.e., $\lambda_i \geq 0 \; \forall i$.
- Negative definite (shown as $G < 0$), if it satisfies $x^T Gx < 0$ for all x. This condition is satisfied if all the eigenvalues λ_i are strictly negative, i.e., $\lambda_i < 0 \; \forall i$.
- Negative semi-definite (shown as $G \leq 0$), if it satisfies $x^T Gx \leq 0$ for all x. This condition is satisfied if all the eigenvalues λ_i are non-positive, i.e., $\lambda_i \leq 0 \; \forall i$.
- Indefinite if neither of the above conditions hold.

Also, there are some useful identities.

- For some symmetric positive (or negative) definite matrix $G = G^T$, we have, $\lambda_{min}(G)||x||^2 \leq x^T Gx \leq \lambda_{max}(G)||x||^2$.
- For some symmetric positive definite matrix $G = G^T > 0$, the 2-norm can be expressed as, $||G|| = \max |\lambda_i(G)| = \lambda_{max}(G)$
- For some symmetric positive definite matrix $G = G^T > 0$, we have $x^T Gx \leq ||x|| \; ||G|| \; ||x|| \leq \lambda_{max}(G)||x||^2$
- If G is a negative definite matrix, since $\max |\lambda_i| = |\lambda_{min}|$, we have $||G|| = \max |\lambda_i| = |\lambda_{min}|$ and all the other relevant identities are modified respectively.
- For some symmetric positive definite matrix $G = G^T > 0$, $A^T GA$ is also positive definite given A is full column rank.
- For some symmetric positive definite matrix $G = G^T > 0$, $-G$ is negative definite.

Consider the autonomus discrete time system below.

$$x[k+1] = A_c x[k] \tag{19.11a}$$

$$y[k] = x[k] \tag{19.11b}$$

Theorem 19.3 *Lyapunov Stability for Linear Systems:* Let $V_k(x) = x[k]^T Px[k]$ be the "Lyapunov Function" candidate of the system with a positive definite symmetric matrix $P = P^T > 0$. That is, $V_k(x) > 0$ for all $x[k]$. Then, the system is asymptotically stable if there exists a positive definite symmetric matrix $L = L^T > 0$ such that $A_c P A_c - P = -L$. Note that, the existance of a single (P, L) pair guarantees the asymptotic stability of the system.

Proof: We know $V_k(x) = x[k]^T Px[k]$. Then,

$$V_{k+1}(x) = x[k+1]^T Px[k+1] = x[k]^T A_c^T P A_c x[x] \tag{19.12a}$$

$$\Delta V = V_{k+1} - V_k = x[k]^T (A_c^T P A_c - P)x[k] \tag{19.12b}$$

$$= -x[k]^T Lx[k] \tag{19.12c}$$

If ΔV is negative for all $x[k]$, we have a monotonically decreasing $V_k(x)$. Combining with the fact that $V_k(x) > 0$, we have

$$\Delta V < 0 \implies \lim_{k\to\infty} V_k(x) = 0 \implies \lim_{k\to\infty} x[k] = 0 \tag{19.13}$$

Which implies the system is asymptotically stable. From this point, we also have a decay rate bound for the norm of the state vector $||x[k]||$ following from the matrix identities.

$$V_{k+1} - V_k = -x[k]^T Lx[k] \leq -\lambda_{min}(L)||x[k]||^2 \tag{19.14a}$$

$$\leq -\frac{\lambda_{min}(L)}{\lambda_{max}(P)} V_k(x) \tag{19.14b}$$

$$V_{k+1} \leq \underbrace{\left[1 - \frac{\lambda_{min}(L)}{\lambda_{max}(P)}\right]}_{\beta<1} V_k(x) \tag{19.15a}$$

$$\implies V_k(x) \leq \beta^k V_0(x) \tag{19.15b}$$

$$\implies ||x[k]||^2 \leq \frac{1}{\lambda_{min}(P)} V_k(x) \leq \frac{\beta^k}{\lambda_{min}(P)} V_0(x) \tag{19.15c}$$

$$\implies ||x[k]||^2 \leq \frac{\lambda_{max}(P)}{\lambda_{min}(P)} \beta^k ||x[0]||^2 \tag{19.15d}$$

$$\implies ||x[k]|| \leq \sqrt{\frac{\lambda_{max}(P)}{\lambda_{min}(P)}} \left[\sqrt{1 - \frac{\lambda_{min}(L)}{\lambda_{max}(P)}}\right]^k ||x[0]|| \tag{19.15e}$$

Here, the term

$$\sqrt{1 - \frac{\lambda_{min}(L)}{\lambda_{max}(P)}} \tag{19.16}$$

is the bound on the decay rate of the norm of the state vector.

■

Corollary 19.4 *If all the eigenvalues of A_c is strictly inside the unit circle, there exists a positive definite symmetric matrix pair (P, L) such that $x^T Px > 0$ and $A_c^T PA_c - P = -L$.*

The LQR Policy

Remember the previously mentioned state feedback regulation problem.

$$x[k+1] = (A - BK)x[k] = A_c x[k] \tag{19.17}$$

If the pair (A, B) is reachable, we have an infinite number of state feedback constant K to stabilize the closed loop system. With the LQR problem, we are aiming to choose the "optimal" state feedback policy K with respect to some cost function.

LQR Optimal Control Problem

With LQR, we will select the inputs to the system that solves the following optimization problem.

$$\begin{aligned} \min_{u[.]} \quad & \sum_{k=0}^{N-1} (x[k]^T Qx[k] + u[k]^t Ru[k]) + x[N]^T Q_f x[N] && (19.18a) \\ \text{s.t.} \quad & x[k+1] = Ax[k] + Bu[k] && (19.18b) \\ & x[0] = x_0 && (19.18c) \end{aligned}$$

In this problem, N is the control horizon. The optimization problem is solved and the resulting optimal input sequence $u^*[k]$ is applied to the system for one control horizon. After the horizon ends, the initial condition is updated and the problem is solved again. The resulting control policy is a time-varying policy that minimizes the state $x[k]$ and the control effort $u[k]$. The tradeoff between the state decay rate and the input effort is determined by the positive semi-definite symmetric cost matrix Q and positive definite cost matrix R. Q_f is the final state cost function not necessarily equal to Q and might be important for some LQR applications, however, we will assume it is equal to the Q. Note that the cost function is a quadratic function and its minimum value is zero. So, if (A,B) are reachable, the LQR problem will drive the state to the origin eventually. We will prove this property in the following sections.

The Solution of the LQR Problem

19.0.0.1 Solution with Least Squares

One naive approach to solve the LQR problem is to use a least squares formulation. We know that the system difference equation has the solution of,

$$x[N] = A^N x[0] + \sum_{k=0}^{N-1} A^{N-1-k} Bu[k] \tag{19.19}$$

In matrix form,

$$\underbrace{\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \\ x[N] \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & \dots & B \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} u[0] \\ u[1] \\ \vdots \\ u[N-1] \end{bmatrix}}_{U} + \underbrace{\begin{bmatrix} I \\ A \\ \vdots \\ A^{N} \end{bmatrix}}_{H} + x_0 \tag{19.20}$$

Also, assume that $\bar{Q} = diag_N(Q, Q, \dots, Q)$ and $\bar{R} = diag_{N-1}(R, R, \dots, R)$. So the LQR problem becomes,

$$\min_U \quad X^T\bar{Q}X + U^T\bar{R}U \tag{19.21a}$$

$$= \min_U \quad U^T \underbrace{(G^T\bar{Q}G + \bar{R})}_{M} U + 2\underbrace{x_0H^TG}_{\alpha} U + \underbrace{x_0^2H^T\bar{Q}H}_{\beta} \tag{19.21b}$$

$$= \min_U \quad U^TMU + 2\alpha U + \beta \tag{19.21c}$$

Rewriting this problem to separate the constant part and U dependent part,

$$= \min_U \quad (U + M^{-1}\alpha)^TM(U + M^{-1}\alpha) + \underbrace{\beta - \alpha^TM^{-1}\alpha}_{Constant} \tag{19.22a}$$

The constant term has no effect on the solution of the optimization problem. So,

$$= \min_U \quad (U + M^{-1}\alpha)^TM(U + M^{-1}\alpha) \tag{19.23a}$$

$$U^* = -M^{-1}\alpha \tag{19.23b}$$

With this least squares solution, we need to invert the M matrix, which becomes huge as the control horizon increases. So, solving the LQR problem using the least squares formulation is not recommended. Next, we will see a recursive solution to the LQR problem, which is much more efficient and leads to the "infinite horizon" solution, which is not possible to get with the least squares solution.

Solution with Dynamic Programming

Dynamic programming is an optimization method that solves the optimization problem backwards in time. In this case, we define a so called cost-to-go function $J_k(x, u)$. This function represents the total future cost at timestep k. So, we can write any optimization problem as the cost at the current timestep plus the cost-to-go. This fact is illustrated below.

$$J_0(x, u) = \sum_{k=0}^{N-1} (x[k]^TQx[k] + u[k]^TRu[k]) + x[N]^TQ_fx[N] \tag{19.24a}$$

$$J_k(x, u) = x[k]^TQx[k] + u^T[k]Ru[k] + J_{k+1}(x, u) \tag{19.24b}$$

If we have a terminal cost, we can compute the cost-to-go backwards in time in an optimal way. By taking the optimal action at each timestep, we minimize the overall cost function J_0. This fact is called "Bellman Recursion" and Bellman himself explains this fact as: "An optimal policy has the property that, whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

Now lets assume our optimal cost-to-go has the form of $J_k^* = x[k]^T S_k x[k]$. We know that our terminal cost (the cost at final timestep) is $J_N = x[N]^T Q_f x[N]$. So, we can let $S_N = Q_F$. Then the recursion takes the form below.

$$\begin{aligned} J_{N-1} &= x[N-1]^T Q x[N-1] + u[N-1]^T R u[N-1] + x[N]^T S_N x[N] && (19.25a) \\ &= x[N-1]^T Q x[N-1] + u[N-1]^T R u[N-1] + (Ax[N-1] + Bu[N-1])^T S_N (Ax[N-1] + Bu[N-1]) && (19.25b) \end{aligned}$$

For the sake of notational simplicity, we will use $x[N-1] = x^-$, and $u[N-1] = u^-$, so that,

$$J_{N-1} = {x^-}^T Q x^- + {u^-}^T R u^- + (Ax^- + Bu^-)^T S_N (Ax^- + Bu^-) \tag{19.26}$$

To find the optimal input ${u^-}^*$ that minimizes the cost-to-go at timestep $N-1$, we take the derivative of the expression with respect to u^- and equate it to zero.

$$\frac{\partial J_{N-1}}{\partial u^-} = 2R{u^-}^* + 2B^T S_N A x^- + 2B^T S_N B {u^-}^* = 0 \tag{19.27}$$

$${u^-}^* = -\underbrace{(R + B^T S_N B)^{-1} B^T S_N A}_{K_{N-1}} x^- \tag{19.28}$$

We know $J_{N-1}^* = {x^-}^T S_{N-1} x^-$. If we plug-in the optimal input ${u^-}^*$, we obtain,

$${x^-}^T S_{N-1} x^- = {x^-}^T Q x^- + {x^-}^T K_N^T R K_N x^- + x^-(A - BK_N)^T S_N (A - BK_N) x^- \tag{19.29}$$

$${x^-}^T S_{N-1} x^- = {x^-}^T (Q + K_{N-1}^T R K_{N-1} + A^T S_N A - 2A^T S_N B K_{N-1} + K_{N-1}^T B^T S_N B K_{N-1}) x^- \tag{19.30}$$

We write plug-in the expression for K_N and make cancellations to obtain,

$$S_{N-1} = Q + A^T S_N A - A^T S_N B K_N \tag{19.31}$$

We can generalize this procedure to obtain the backwards LQR recursion.

$$u^*[k] = -\underbrace{(R + B^T S_{k+1} B)^{-1} B^T S_{k+1} A}_{K_k} x[k] \tag{19.32a}$$

$$\begin{aligned} S_k &= Q + A^T S_{k+1} A - A^T S_{k+1} B K_k && (19.32b) \\ &= Q + A^T S_{k+1} A - A^T S_{k+1} B (R + B^T S_{k+1} B)^{-1} B^T S_{k+1} A && (19.32c) \end{aligned}$$

The cost-to-go function $J_k^* = x[k]^T S_k x[k]$ is actually derived from the famous Hamilton-Jacobi-Bellman equation, however, the derivation is out of the scope of this course. The finite horizon LQR problem can be solved backward in time by letting $S_N = Q_f$ and using the recursion.

Infinite Horizon Case

The recursive nature of the dynamic programming procedure allows us to define an infinite horizon problem to obtain a time-independent feedback law. The intuition behind the infinite horizon case is that it minimizes the average effort to drive the state to the origin. In fact, it is guaranteed that the cost-to-go matrix S_k and the feedback law K_k converge to a stationary value S_∞ and K_∞ eventually, which can stabilize any system given (A, B) pair is reachable. After S_∞ and K_∞ converges, the Bellman Recursion takes the form,

$$S_\infty = Q + A^T S_\infty A - A^T S_\infty B(R + B^T S_\infty B)^{-1} B^T S_\infty A \tag{19.33}$$

Which is called the Discrete Time Algebraic Riccati Equation (DARE). Solving DARE, we obtain S_∞ and use it to find the infinite horizon gain K_∞.

$$u^*[k] = -\underbrace{(R + B^T S_\infty B)^{-1} B^T S_\infty A}_{K_\infty} x[k] \tag{19.34}$$

The Asymptotic Stability of Infinite Horizon LQR Feedback Law

From previous sections, we know the closed loop system is asymptotically stable if there exist a positive definite (P,L) pair such that $A_c P A_c - P = -L$. If we let $P = S_\infty$ and $L = Q + K_\infty R K \infty$,

$$\begin{aligned}
&(A - BK_\infty)^T S_\infty (A - BK) - S_\infty = -Q - K_\infty^T R K_\infty && (19.35a)\\
\Longrightarrow\ & A^T S_\infty A - 2A^T S_\infty B K_\infty + K_\infty^T B^T S_\infty B K_\infty - S_\infty = -Q - K_\infty^T R K_\infty && (19.35b)\\
\Longrightarrow\ & S_\infty = Q + K_\infty^T R K_\infty + A^T S_\infty A - 2A^T S_\infty B K_\infty + K_\infty^T B^T S_\infty B K_\infty && (19.35c)
\end{aligned}$$

With a similar arrangement of the equation done in Equation (19.30), we obtain,

$$S_\infty = Q + A^T S_\infty A - A^T S_\infty B(R + B^T S_\infty B)^{-1} B^T S_\infty A \tag{19.36}$$

which is the Discrete Time Algebraic Riccati Equation. Thus, we can say that, for a system with controllable (A, B) pair, the solution of DARE P_∞ and the resulting feedback law K_∞ makes the closed loop system asymptotically stable with a decay rate of

$$\sqrt{1 - \frac{\lambda_{min}(Q + K_\infty^T R K_\infty)}{\lambda_{max}(S_\infty)}} \tag{19.37}$$

Rules of Thumb for LQR Design

- If full state information is not available, using $Q = CC^T$ gives a satisfactory performance by minimizing the output signal.
- LQR is originally a regulator. To apply LQR to a tracking problem, consider the system given in Equation (19.10f) and use the cost function

$$\sum_{k=0}^{N-1} \left[\bar{x}[k]^T Q \bar{x}[k] + u[k]^T R u[k]\right] + \bar{x}[N]^T Q_f \bar{x}[N] \tag{19.38}$$

You can use the same DARE equation with the augmented A and B matrix of the integrator system. In this case, the control policy takes the name "Linear Quadratic Integrator (LQI)" instead of LQR. In this case, the addition of the integrator can make the system unreachable, especially for MIMO systems. Check the system's reachability before attempting to solve DARE.