

Introduction to Linear Quadratic Regulation

Robert Platt
Computer Science and Engineering
SUNY at Buffalo

February 13, 2013

1 Linear Systems

A linear system has dynamics that can be represented as a linear equation. Let $x_t \in \mathbb{R}^n$ denote the state ¹ of the system at time t . Let $u_t \in \mathbb{R}^m$ denote the action (also called the *control*) taken by the system at time t . Linear system dynamics can always be represented in terms of an equation of the following form:

$$x_{t+1} = Ax_t + Bu_t, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is a constant $n \times n$ matrix and $B \in \mathbb{R}^{n \times m}$ is a constant $n \times m$ matrix. Given that the system takes action u_t from state x_t , this equation allows us to predict the state of the next time step. The reason we call Equation 1 a *linear* equation is that it is linear in the variables, x_t and u_t .² Another thing to notice about the above equation is that it is written in terms of the states and actions taken at discrete time steps. As a result, we refer to this as a *discrete time* system.

A classic example of a linear system is the damped mass. Imagine applying forces to an object lying on a frictional surface. Denote the mass of the object by m and the coefficient of friction by b . Let $r_t, \dot{r}_t \in \mathbb{R}^n$, and $\ddot{r}_t \in \mathbb{R}^n$ denote the position, velocity, and acceleration of the object, respectively. There are three forces acting on the object. The inertial force at time t is $m\ddot{r}_t$. The frictional (viscous friction) force is $b\dot{r}_t$. Let the applied force at time t be denoted f_t . The motion of the object is described by the following second order differential equation known as the *equation of motion*:

$$m\ddot{r}_t + b\dot{r}_t = f_t.$$

¹State is assumed to be Markov in the sense that it is a sufficient statistic to predict future system behavior.

²If one of these variables were squared (for example, $x_{t+1} = Ax_t^2 + Bu_t$), then this equation would no longer be linear.

Suppose we want to write the equation of motion of the object in as a discrete time equation of the form of Equation 1. Choose (arbitrarily) a period between successive time steps as 0.1 seconds. The position at time $t + 1$ is:

$$r_{t+1} = r_t + 0.1\dot{r}_t. \quad (2)$$

The velocity at time $t + 1$ is:

$$\begin{aligned} \dot{r}_{t+1} &= \dot{r}_t + 0.1\ddot{r}_t \\ &= \dot{r}_t + 0.1\frac{1}{m}(f_t - b\dot{r}_t). \end{aligned} \quad (3)$$

Then Equations 2 and 3 can be written as a system of two equations as follows:

$$\begin{pmatrix} r_{t+1} \\ \dot{r}_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 0.1 \\ 0 & 1 - 0.1\frac{b}{m} \end{pmatrix} \begin{pmatrix} r_t \\ \dot{r}_t \end{pmatrix} + \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} u_t.$$

This can be re-written as:

$$x_{t+1} = Ax_t + Bu_t,$$

where

$$\begin{aligned} x_t &= \begin{pmatrix} r_t \\ \dot{r}_t \end{pmatrix}, \\ A &= \begin{pmatrix} 1 & 0.1 \\ 0 & 1 - 0.1\frac{b}{m} \end{pmatrix}, \end{aligned}$$

and

$$B = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix}.$$

2 Control Via Least Squares

Consider an initial state, x_1 , and a sequence of $T - 1$ actions, $\mathbf{u} = (u_1, \dots, u_{T-1})^T$. Using the system dynamics Equation 1, we can calculate the corresponding sequence of states. Let $\mathbf{x} = (x_1, \dots, x_T)^T$ be the sequence of T states such that Equation 1 is satisfied (the trajectory of states visited by the system as a result of taking the action sequence, \mathbf{u}). The objective of control is to take actions that cause the following cost equation to be minimized:

$$J(\mathbf{x}, \mathbf{u}) = x_T^T Q_F x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t, \quad (4)$$

where $Q \in \mathbb{R}^{n \times n}$ and $Q_F \in \mathbb{R}^{n \times n}$ determine state costs and $R \in \mathbb{R}^{m \times m}$ determines action costs.

One way to solve the control problem for linear systems with quadratic cost functions is to solve a least squares problem. Suppose that the sequence of actions, $\mathbf{u} = (u_1, \dots, u_{T-1})^T$, is taken starting in state x_1 at time $t = 1$. The state at time $t = 2$ is:

$$x_2 = Ax_1 + Bu_1.$$

The state of time $t = 3$ is:

$$\begin{aligned} x_3 &= A(Ax_1 + Bu_1) + Bu_2 \\ &= A^2x_1 + ABu_1 + Bu_2. \end{aligned}$$

The state of time $t = 4$ is:

$$\begin{aligned} x_4 &= A(A^2x_1 + ABu_1 + Bu_2) + Bu_3 \\ &= A^3x_1 + A^2Bu_1 + ABu_2 + Bu_3. \end{aligned}$$

The entire trajectory of states over time horizon T can be calculated in a similar way:

$$\begin{aligned} \mathbf{x} &= \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ B & 0 & 0 & 0 & \dots & 0 \\ AB & B & 0 & 0 & \dots & 0 \\ A^2B & AB & B & 0 & \dots & 0 \\ A^3B & A^2B & AB & B & \dots & 0 \\ \vdots & & & & & \vdots \\ A^{T-1}B & A^{T-2}B & A^{T-3}B & A^{T-4}B & \dots & B \end{pmatrix} \mathbf{u} + \begin{pmatrix} I \\ A \\ A^2 \\ A^3 \\ \vdots \\ A^T \end{pmatrix} x_1 \\ &= G\mathbf{u} + Hx_1. \end{aligned} \tag{5}$$

The cost function can also be “vectorized”:

$$\begin{aligned} J &= \mathbf{x}^T \begin{pmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & Q_F \end{pmatrix} \mathbf{x} + \mathbf{u}^T \begin{pmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & R \end{pmatrix} \mathbf{u} \\ &= \mathbf{x}^T Q\mathbf{x} + \mathbf{u}^T \mathcal{R}\mathbf{u} \\ &= (G\mathbf{u} + Hx_1)^T Q(G\mathbf{u} + Hx_1) + \mathbf{u}^T \mathcal{R}\mathbf{u} \end{aligned} \tag{6}$$

$$\tag{7}$$

The control problem can be solved by finding a \mathbf{u} that minimizes Equation 7. This is a least squares problem and can be solved using the pseudoinverse. For simplicity, assume that \mathcal{R} is zero.³ In this case, the objective is to find the \mathbf{u} that minimizes the following $L2$ norm:

$$\|Q^{1/2}G\mathbf{u} + Q^{1/2}Hx_1\|_2.$$

³The case where \mathcal{R} is non-zero can be solved by writing Equation 7 as a quadratic form by completing the square.

The least squares solution is:

$$\mathbf{u} = (Q^{1/2}G)^+ Q^{1/2}Hx_1, \quad (8)$$

where $(\cdot)^+$ denotes the pseudoinverse ⁴

3 Control Via Dynamic Programming

Equation 8 calculates an optimal control, \mathbf{u} , in the sense that no other control will achieve a lower cost, J . However, it can be inconvenient to use the direct least squares method to calculate control because of the need to create those big matrices. Moreover, the solution requires inverting a big matrix. In this section, I introduce an alternative method for calculating the same solution that uses the Bellman optimality equation ⁵. Let $V_t(x)$ denote the optimal value function at x at time t . Specifically, $V_t(x)$ is equal to the future cost that would be experienced by the system if the optimal policy is followed. This is similar to the way we defined the value function in Reinforcement Learning (RL). However, whereas the value function in the RL section denoted expected future rewards, the value function in this discussion denotes future costs.

The Bellman equation ⁶ is:

$$V_t(x) = \min_{u \in \mathbb{R}^m} (x^T Qx + u^T Ru + V_{t+1}(Ax + bu)). \quad (9)$$

Assume that we somehow know ⁷ that the value function at time $t + 1$ is a quadratic of the form,

$$V_{t+1}(x) = x^T P_{t+1}x, \quad (10)$$

where P_{t+1} is a positive semi-definite matrix ⁸. In this case, the Bellman equation becomes:

$$V_t(x) = x^T Qx \min_{u \in \mathbb{R}^m} (u^T Ru + (Ax + bu)^T P_{t+1}(Ax + bu)). \quad (11)$$

In the above, we are minimizing a quadratic function. As a result, we can calculate the global minimum by setting the derivative to zero:

$$\begin{aligned} 0 &= \frac{\partial}{\partial u} (u^T Ru + (Ax + bu)^T P_{t+1}(Ax + bu)) \\ &= 2u^T R + 2x^T A^T P_{t+1}B + u^T B^T P_{t+1}B. \end{aligned}$$

⁴Recall that for a non-singular matrix A , $A^+ = (A^T A)^{-1} A^T$ when A is tall (rows greater than columns) and $A^+ = A^T (A A^T)^{-1}$ when A is fat (columns greater than rows).

⁵The same bellman optimality equation we studied in the section on Reinforcement Learning.

⁶Often called the Hamilton-Jacobi-Bellman equation in the controls literature.

⁷Leprechauns told us?

⁸A matrix is positive semi-definite if all its eigenvalues are non-negative.

The optimal control is therefore:

$$u^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x. \quad (12)$$

Substituting u^* back into Equation 11, we have:

$$\begin{aligned} V_t(x) &= x^T (Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A) x \\ &= x^T P_t x. \end{aligned} \quad (13)$$

Notice that this equation is exactly of the form of Equation 10. We have just shown that if it is assumed that a linear system with quadratic costs has a value function of the form in Equation 10 at time $t + 1$, then it will have the same form at time t . Furthermore, notice that the value function on the final time step (at time T) is exactly equal to the cost function (there are no future rewards to take into account):

$$V_T(x) = x^T Q_F x.$$

As a result, we know that the value function for any linear system with quadratic costs is quadratic for all time. Equation 14 tells us that if we are given the matrix P_{t+1} that defines the value function at $t + 1$, then we can calculate P_t as follows:

$$P_t = Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A. \quad (14)$$

This is known as the Differential Riccati Equation and is relevant to finite time horizon problems.

These are all the equations that we need to use finite horizon discrete time LQR. It is used as follows. First, identify a time horizon, T , over which control will occur. Second, set $P_T = Q_F$. This reflects the knowledge that on the final time step, there are no future rewards and the value function is exactly equal to the cost function. Third, use the differential Riccati equation to calculate P_{T-1} , P_{T-2} , and so on back to P_1 . At this point, we have a policy that can be used to calculate control actions. On each time step, t , we take an action: $u_t^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x_t$.

Now, consider an infinite horizon problem. In this case, the Bellman equation (Equation 9) becomes:

$$V(x) = \min_{u \in \mathbb{R}^m} (x^T Q x + u^T R u + V(Ax + bu)).$$

Performing a similar derivation to what we did before, we obtain:

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A. \quad (15)$$

This is known as the Discrete Algebraic Riccati Equation (DARE). The key thing to notice about this equation is that P is no longer indexed by time. In order to use it, we solve Equation 15 for P ⁹. The optimal policy is: $u_t^* = -(R + B^T P B)^{-1} B^T P A x_t$. Notice that the policy no longer depends upon the time index as it did in the finite horizon case. An important connection exists between the finite and infinite horizon cases. As the time horizon, T , becomes large, the differential Riccati equation converges to the solution to the DARE, regardless of the value of the final cost matrix, Q_F .

⁹There are standard methods for solving the DARE. For example, Matlab has a function, `DLQR`, that does this.

4 LQR Controllability

In control, the term *controllable* has an important technical meaning. A system is controllable if for any initial state and any final state, a sequence of control actions exists that takes the system from between the two states in finite time. For linear systems, the following important test can be used to test controllability over a finite horizon, T : we check whether the following matrix has full (row) rank ¹⁰:

$$(B, AB, A^2B, A^3B, \dots, A^{T-1}B).$$

Notice that if this matrix is full rank, then the bottom row of the G matrix in Equation 5 has full rank. This means that for any x_1 and any x_T , a u exists that causes the system to reach x_T .

5 LQR Stability

Another word with important technical meaning in control is *stability*. There are many slightly different ways of defining stability. Here, we will use the following: a system is stable if, as time goes to infinity, state remains within a constant radius of the origin. A system is *closed-loop stable* if it is stable while being controlled by a given particular control policy. For linear systems, stability can be evaluated by considering the eigenvalues of the relevant matrix. Consider the following differential equation:

$$\dot{x}_t = \bar{A}x_t. \tag{16}$$

Since this is a first order differential equation, it can be integrated in closed form ¹¹:

$$x_t = \sum_{i=1}^n c_i v_i e^{\lambda_i t}, \tag{17}$$

where n is the number of eigenvalues in \bar{A} , c_i is a constant, and v_i and λ_i are the eigenvectors and eigenvalues of \bar{A} , respectively. Notice that the eigenvalues λ_i are in the exponent of the right side terms of Equation 17. Therefore, this solution decays to zero when all eigenvalues are negative. If one eigenvalue is positive, then there is one term in the trajectory that goes to infinity (exponentially) as time goes to infinity.

In the context of the current discussion, the question becomes: when is Equation 1 stable? Suppose we are given Equation 16 and we wish to construct the discrete time system where the time between successive discrete times is δ . Then we have:

$$x_{t+1} = (I + \delta\bar{A})x_t.$$

¹⁰Full row rank means that all rows are linearly independent.

¹¹This solution is found using the characteristic equation of the differential equation.

This system will be stable if all the eigenvalues of $I + \delta \bar{A}$ are less than one. Therefore, the system of Equation 1 is stable when all eigenvalues of A are less than one.