4 State Transition, Convolution, and Riccati Equations

4.1 State Transition Matrices

Consider the linear time-varying system

$$\dot{x} = A(t)x(t) \qquad x(t_0) = x_0.$$

The state transition matrix for this system is the matrix $\Phi(t, t_0)$ that satisfies the *matrix-valued* differential equation

$$\dot{\Phi} = A(t)\Phi \qquad \Phi(t_0, t_0) = Id_{n\times n}.$$

Amazingly,

$$x(t) = \Phi(t, t_0)x_0$$

holds for any linear system. In the special case where $A(t) = A$—the system is linear time-invariant (LTI)—it turns out that $\Phi(t, t_0) = e^{A(t-t_0)}$, the matrix exponential of $A(t - t_0)$.

4.2 Convolution equations

Consider the linear time-varying system

$$\dot{x} = A(t)x(t) + B(t)u(t) \qquad x(t_0) = x_0.$$

This does not have the same linear structure as the previous system, so the state transition matrix associated with $A(t)$ is not quite enough to compute the solution. Instead, we get the *convolution* equation

$$x(t) = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)d\tau.$$

Now, this equation is not quite as complex as it looks. Roughly speaking, it says that the control is coupled to what the system would have done based on its initial condition through the integral equation that weighs the input against the state transition matrix operating on the input. If $A = 0$, then we expect to be able to directly control the state; this is exactly what is predicted because for $A = 0$ we have $\Phi(t, t_0) = Id_{n\times n}\ \forall\ t$.

4.3 Using Linear Solutions to Find Descent Directions

Assume we have linear time-varying equations of motion. The dynamics are

$$\dot{x} = A(t)x + B(t)u \qquad x(0) = x_0$$

and

$$x(\tau) = \Phi(\tau, 0)x_0 + \int_0^{\tau} \Phi(\tau, s)B(s)u(s)ds.$$

Let's assume we have the simplest cost function imaginable–a cost that is quadratic in x and u:

$$J(u(\cdot)) = \frac{1}{2}\int_0^T x(t)^T Q(t)x(t) + u(t)^T R(t)u(t)dt + \frac{1}{2}x(T)^T P_1 x(T) \qquad (5)$$

where $Q = Q^T \geq 0$, $R = R^T > 0$, $P_1 = P_1^T \geq 0$. We now formally differentiate $J(\cdot)$ with respect to $u(\cdot)$ (assuming, potentially incorrectly, that everything is nicely differentiable[6]).

[6] Note that we cannot possibly know whether or not things are differentiable because we do not know what space we are working in.

Calculation of Necessary Conditions for Optimality

$$\frac{d}{d\epsilon} J(u(t) + \epsilon v(t))|_{\epsilon=0}$$
$$= \int_0^T x^T Q z + u^T R v dt + x(T)^T P_1 z(T)$$
$$\text{where } z(\cdot) = \frac{\partial x(\cdot)}{\partial u}$$
$$= \int_0^T \left[\underbrace{x(\tau)^T Q}_{a(\tau)^T} \underbrace{\left[\int_0^\tau \Phi(\tau, s) B(s) v(s) ds\right]}_{z(\tau)} + \underbrace{u(\tau)^T R(\tau)}_{b(\tau)^T} v(\tau) \right] d\tau + \underbrace{x(T)^T P_1}_{p_1^T} \int_0^T \Phi(T, s) B(s) v(s) ds$$
$$= \int_0^T a(\tau)^T \left[\int_0^\tau \Phi(\tau, s) B(s) v(s) ds\right] d\tau + p_1^T \int_0^T \Phi(T, s) B(s) v(s) ds + \int_0^T b(\tau)^T v(\tau) d\tau$$
$$= \int_0^T \left[\int_0^\tau a(\tau)^T \Phi(\tau, s) B(s) v(s) ds\right] d\tau + p_1^T \int_0^T \Phi(T, s) B(s) v(s) ds + \int_0^T b(\tau)^T v(\tau) d\tau$$

Now change the order of integration, noting in Fig. 10 the change in boundary conditions on the integrals.

$$= \int_0^T \left[\int_s^T a(\tau)^T \Phi(\tau, s) d\tau\right] B(s) v(s) ds + p_1^T \int_0^T \Phi(T, s) B(s) v(s) ds + \int_0^T b(\tau)^T v(\tau) d\tau$$
$$= \int_0^T \left[\underbrace{\int_s^T a(\tau)^T \Phi(\tau, s) d\tau + p_1^T \Phi(T, s)}_{p(s)^T} \right] B(s) v(s) ds + \int_0^T b(\tau)^T v(\tau) d\tau$$
$$= \int_0^T (p(\tau)^T B + b(\tau)^T) v(\tau) d\tau.$$

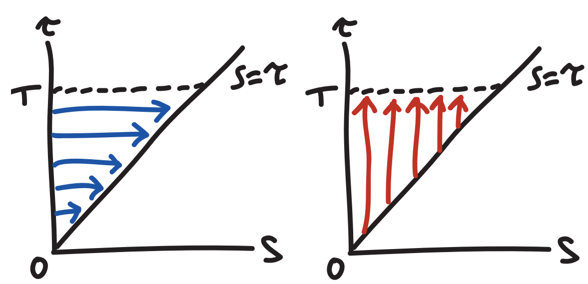

Figure 10: Changing order of integration.

Now, p as it is defined doesn't seem very easy to compute, but it turns out that it satisfies a nice but somewhat unexpected differential equation. You can think of the integral expression that defines $p(s)^T$ as being the convolution equation, except that in this case the state transition matrix $\Phi(\tau, s)$ operates on a boundary condition at the final time T instead of the initial time 0. Moreover, integration is being performed on the first argument of the state transition matrix $\Phi(\tau, s)$ instead of the second argument. (In order to formally get the convolution equation, we would need to reverse this.[7]) Nevertheless, as a consequence of this integral definition of p, we find (in the exercises) that

[7] Let us use the fact that $\Phi(t, t_0) = \Phi^{-1}(t_0, t)$ to rewrite the equation to look more like the convolution equation $p(s) = \int_s^T a(\tau)^T \Phi^{-1}(s, \tau) d\tau + p_1^T \Phi^{-1}(s, T)$. This is *almost* the convolution equation for a system with Φ^{-1} as the state transition matrix. However, since p flows *backwards* in time from p_1 at time T to "final" time s (treating s like a terminal time), the integration must also be reversed, producing the convolution equation $p(s) = -\int_T^s a(\tau)^T \Phi^{-1}(s, \tau) d\tau + p_1^T \Phi^{-1}(s, T)$. Due to the properties of the state transition matrix, it turns out that if Φ is the state transition matrix for LTV systems with linearization $A(t)$, Φ^{-1} is the state transition matrix for a system with linearization $-A(t)$—so the convolution equation indicates $p(s)$ can be computed from a linear affine ordinary differential equation.

$p(\tau)$ satisfies the differential equation

$$\dot{p} = -A^T p - Qx \quad p(T) = p_1$$

and we get

$$DJ(u) \cdot v = \frac{d}{d\epsilon} J(u(t) + \epsilon v(t))|_{\epsilon=0} = \int_0^T [p(\tau)B + b(\tau)]v(t)dt$$

so, to be optimal, $u(\cdot)$ must satisfy the equation

$$[p(t)^T B + u(t)^T R] = 0$$

or, more conveniently,

$$B^T p(t) + Ru(t) = 0 \tag{6}$$

which implies

$$u = -R^{-1} B^T p(t).$$

Note that if we write down the three conditions for optimality together, we get

$$\begin{aligned}
\dot{x} &= Ax + Bu \\
\dot{p} &= -A^T p - Qx \\
0 &= B^T p(t) + Ru(t).
\end{aligned}$$

This is the differential statement of the famous *Maximum Principle*, which we will discuss more later. It will turn out that there is a natural way to obtain these equations from an appropriately defined Hamiltonian; this will be very helpful to us later when we want to solve some LQR problems that have off-diagonal terms.

Rewriting the above equations by substituting in the value of u, we get

$$\begin{aligned}
\dot{x} &= Ax - BR^{-1} B^T p(t) \\
\dot{p} &= -A^T p - Qx \\
0 &= B^T p(t) + Ru(t).
\end{aligned}$$

which, in matrix form, looks like

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} \qquad \begin{bmatrix} x(0) \\ p(T) \end{bmatrix} = \begin{bmatrix} x_0 \\ p_1 \end{bmatrix} \tag{7}$$

Note this does not have the same boundary condition structure that we are used to (e.g., initial value problems) because it has an *initial* condition in x and a final condition in p. This is called a two-point boundary value problem (TPBVP). You can just solve this in MATLAB, but we are going to opt to be more clever than that.

The first-order optimality condition is equivalent to the *solvability* of (7). The question is whether or not we can solve this TPBVP? It turns out that something magical happens here; in particular, it is true that $p(t) = P(t)x(t)$ for some choice of $P(t)$ (at least near $t = T$).

4.4 Riccati Equations

Now, what differential equation does $P(t)$ satisfy, assuming it exists at all? We want to know because we would like to be working with lower-dimensional, better-conditioned systems if possible. We want $P(\cdot)$ so that $p(t) = P(t)x(t)$ for $t < T$ (hopefully $t \in [0, T]$, but we will see in the next section that we can't necessarily guarantee that). What do we do? What we always do: Differentiate!

$$\begin{aligned}
p &= Px \\
\Rightarrow \dot{p} &= \dot{P}x + P\dot{x} \\
&= \dot{P}x + P(Ax - BR^{-1}B^T Px) \\
\text{and } \dot{p} &= -Qx - A^T Px \\
\Rightarrow 0 &= (\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q)x
\end{aligned}$$

This equation has to hold for all possible trajectories $x(\cdot)$, so the matrix-valued differential equation

$$\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q = 0 \qquad P(T) = P_1$$

must hold. This is a *Riccati equation.* This Riccati Equation is a quadratic equation in P that runs backward in time. Since it is smooth, the soution will exist and be unique from T down to some $t^* < T$ which implies that $P(t)$ on $(t^*, T]$ is well defined. We *hope* that $t^* < t_0$. (You will see in the exercises that it does not need to be.)

Exercises

Given a two dimensional linear system (e.g., $\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$) and a quadratic objective function (e.g., $J = \int_0^1 \frac{1}{2} x^T x + u^2 dt + 10x(T)^T x(T)$), find the Riccati solution for the optimal control and numerically convince yourself that it is the optimizer. (Think about how to do this using tools we have discussed in class.)