19 Linear Quadratic Regulator

You have your way. I have my way. As for the right way, the correct way, and the only way, it does not exist.
—Friedrich Nietzsche

You can always count on Americans to do the right thing - after they've tried everything else.
—Winston Churchill

19.1 Derivation of the LQR solution

Up to this point in our discussion of optimal control (and a lot of what we see in the next couple of chapters) we have (will) come up with problems in which we suppose that we have an infinitely precise knowledge of everything, i.e. we are taking the omniscient perspective which is a danger that all good mathematicians must take care to avoid. We are ignoring the fact that not only do we not know everything, but what we do know is often incorrect when we are trying to model a given physical problem (ideally you would address this via stochastic optimal control, but we don't anticipate having the time/space for that in this text). This supposition of omniscience will get a practical mathematician in trouble when they go to apply their knowledge to real life situations. In practice, it is very rare that any system can be described completely from the outset. This means that most optimal control problems are actually broken up into smaller time intervals over which the optimal control formulation is applied at each step. Each of these smaller optimal control problems must be solved very rapidly though, or the updated system will respond poorly to the previous inputs and controls. This is all well and good, but from what we have seen up to now, we are talking about solving a nonlinearly constrained system of ODEs at best, which is a hideously horrible mess (alliteration bonus points). Thus we really need something that will give us the optimal solution much faster.

This is where the linear quadratic regulator comes in. If you recall form previous chapters, we have repeatedly recognized that quadratic cost functionals are very nice, yielding a unique global minimizer. In addition, you may recall from ODEs that if we are evolving the state equation for only a short amount of time, then linearization about the current state is a reasonable approach for modeling the state evolution. With these two observations backing us up, we will consider the linear quadratic regulator. Linear, because the system equation is linear in both the control and the state variable, and quadratic because the cost function is quadratic in both the control and state variables. I have no idea why 'regulator' is used, but it sure would sound lame without it.

19.1.1 Derivation of LQR

With all of the wonderfully worded motivation above, we will turn to study of a system governed by a linear state equation, i.e.

$$\mathbf{x}' = A(t)\mathbf{x} + B(t)\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{19.1}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $A(t)$ is an $n \times n$ matrix and $B(t)$ is an $m \times n$ matrix. The cost functional we associate with this system is

$$J[\mathbf{u}] = \int_0^{t_f} L(t;, \mathbf{x}, \mathbf{u})dt + \phi(\mathbf{x}(t_f)), \tag{19.2}$$

where

$$L(t; \mathbf{x}, \mathbf{u}) = \mathbf{x}(t)^T Q(t)\mathbf{x}(t) + \mathbf{u}(t)^T R(t)\mathbf{u}(t) \tag{19.3}$$

and

$$\phi(\mathbf{x}(t_f)) = \mathbf{x}(t_f)^T M \mathbf{x}(t_f), \tag{19.4}$$

where M and Q are symmetric and positive semi-definite, and R is symmetric and positive definite. Note that because R is positive definite, its eigenvalues are strictly positive so R is invertible. For the time being we will consider a fixed endtime t_f, with variable end point $\mathbf{x}(t_f)$. The Hamiltonian in this setting is:

$$H = \mathbf{p}(t)^T A(t)\mathbf{x}(t) + \mathbf{p}(t)^T B(t)\mathbf{u}(t) - \mathbf{x}(t)^T Q(t)\mathbf{x}(t) - \mathbf{u}(t)^T R(t)\mathbf{u}(t).$$

Remark 19.1.1. Before we can compute the optimal control $\tilde{\mathbf{u}}(t)$ we first recall a few things from multivariable calculus, and taking the derivative (gradient) for such a scalar valued function as the Hamiltonian. First, note that because the Hamiltonian is a scalar, then each term is as well, and hence

$$\mathbf{p}^T B\mathbf{u} = (\mathbf{p}^T B\mathbf{u})^T = \mathbf{u}^T B^T \mathbf{p}.$$

In addition we recall that the gradient (derivative) of a dot product is given by

$$\frac{D(\mathbf{x}^T\mathbf{y})}{D\mathbf{x}} = \mathbf{y}.$$

Also, the chain rule in such a setting for vector valued functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ becomes:

$$\frac{D\mathbf{h}(\mathbf{f}, \mathbf{g})}{D\mathbf{x}} = \frac{D(\mathbf{g}(\mathbf{x})^T)}{D\mathbf{x}}\frac{D\mathbf{h}}{D\mathbf{g}} + \frac{D(\mathbf{f}(\mathbf{x})^T)}{D\mathbf{x}}\frac{D\mathbf{h}}{D\mathbf{f}}. \tag{19.5}$$

Using this we can see that

$$\frac{D(\mathbf{x}^T A\mathbf{x})}{D\mathbf{x}} = A^T\mathbf{x} + A\mathbf{x}. \tag{19.6}$$

Using all of this information, we first write the Hamiltonian as follows (suppressing the time dependence for now):

$$H = \mathbf{p}^T A\mathbf{x} + \mathbf{u}^T B^T \mathbf{p} - \mathbf{x}^T Q\mathbf{x} - \mathbf{u}^T R\mathbf{u}, \tag{19.7}$$

so that the optimality condition states that the optimal control $\tilde{\mathbf{u}}(t)$ satisfies:

$$\begin{aligned} \frac{DH}{D\mathbf{u}} &= 0, \\ &\Rightarrow B^T\mathbf{p} - R^T\mathbf{u} - R\mathbf{u} = 0 \end{aligned}$$

Then it follows that because R is symmetric, this condition becomes:

$$B^T\mathbf{p} - 2R\mathbf{u} = 0,$$

so that

$$\tilde{\mathbf{u}} = \frac{1}{2}R^{-1}B^T\mathbf{p}.$$

We note that optimality is ensured (globally) because $\frac{D^2H}{D\mathbf{u}^2} = -2R(t) < 0$.

Using the Hamiltonian, and recalling the form of the optimal control for this problem,

$$\begin{aligned} \mathbf{p}' &= -\frac{DH}{D\mathbf{x}} \\ &= -A^T\mathbf{p} + 2Q\mathbf{x}, \end{aligned}$$

with

$$\mathbf{p}(t_f) = -\frac{D\phi}{D\mathbf{x}(t_f)} = -2M\mathbf{x}(t_f),$$

where we freely took advantage of the symmetry of Q and M. The endpoint condition shows that $\mathbf{p}(t_f) = -2M\mathbf{x}(t_f)$, so the state and costate are linearly related at time t_f. Because $\mathbf{p}(t)$ and $\mathbf{x}(t)$ are governed by a system of linear equations, then $\mathbf{p}(t) = -2P(t)\mathbf{x}(t)$ for some matrix $P(t)$ so that $\mathbf{x}$ and $\mathbf{p}$ are linearly related for all time. Hence we only need to discover what the nature of this linear relationship is, i.e. we need to identify the matrix $P(t)$.

To see this, consider the dual evolution of state and costate:

$$\begin{pmatrix} \mathbf{x}' \\ \mathbf{p}' \end{pmatrix} = \mathscr{H}(t)\begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix},$$

where

$$\mathscr{H}(t) = \begin{pmatrix} A(t) & \frac{1}{2}B(t)R(t)^{-1}B(t)^T \\ 2Q(t) & -A(t)^T \end{pmatrix}.$$

Because this is a linear system, it has a fundamental solution matrix $\Phi^{-1}(t)$. In other words, if

$$\mathbf{y}(t) = \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{pmatrix}$$

had an initial condition $\mathbf{y}(0) = \mathbf{y}_0$ then $\mathbf{y}(t) = \Phi^{-1}\mathbf{y}_0$. Similarly, if we know $\mathbf{y}(t_f)$, then $\mathbf{y}(t) = \Phi(t)\mathbf{y}(t_f)\ \forall t \leq t_f$. Let

$$\Phi(t) = \begin{pmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{pmatrix},$$

so that

$$\begin{aligned} \mathbf{x}(t) &= \Phi_{11}\mathbf{x}(t_f) + \Phi_{12}\mathbf{p}(t_f) \\ \mathbf{p}(t) &= \Phi_{21}\mathbf{x}(t_f) + \Phi_{22}\mathbf{p}(t_f). \end{aligned}$$

Then using the endpoint condition on the co-state,

$$\begin{aligned}\mathbf{x}(t) &= \Phi_{11}\mathbf{x}(t_f) - 2\Phi_{12}M\mathbf{x}(t_f)\\ &= (\Phi_{11} - 2\Phi_{12}M)\mathbf{x}(t_f).\end{aligned}$$

Then $\mathbf{x}(t_f) = (\Phi_{11} - 2\Phi_{12}M)^{-1}\mathbf{x}(t)$ so that

$$\begin{aligned}\mathbf{p}(t) &= (\Phi_{21} - 2\Phi_{22}M)\mathbf{x}(t_f)\\ &= (\Phi_{21} - 2\Phi_{22}M)(\Phi_{11} - 2\Phi_{12}M)^{-1}\mathbf{x}(t)\\ &= -2P(t)\mathbf{x}(t).\end{aligned}$$

so that $\mathbf{p}(t) = -2P(t)\mathbf{x}(t)$ where $P(t_f) = M$ and

$$P(t) = -\frac{1}{2}(\Phi_{21} - 2\Phi_{22}M)(\Phi_{11} - 2\Phi_{12}M)^{-1}.$$

Thus the optimal control is

$$\tilde{\mathbf{u}} = \frac{1}{2}R^{-1}B^T\mathbf{p} = -R^{-1}B^TP\tilde{\mathbf{x}}.$$

19.1.2 Summary and implementation of LQR

All the derivation above has proven the following theorem.

Theorem 19.1.2. *For a linear state equation given by*

$$\mathbf{x}' = A(t)\mathbf{x} + B(t)\mathbf{u},\ \ \mathbf{x}(0) = x_0,$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, and $A(t)$ is an $n \times n$ matrix and $B(t)$ is an $n \times m$ matrix, the minimization of the cost functional,

$$J[\mathbf{u}] = \int_0^{t_f} \left[\mathbf{x}(t)^TQ(t)\mathbf{x}(t) + \mathbf{u}(t)^TR(t)\mathbf{u}(t)\right]dt + \mathbf{x}(t_f)^TM\mathbf{x}(t_f),$$

is satisfied by

$$\mathbf{p}(t) = -2P(t)\mathbf{x}(t).$$

So that the optimal control is

$$\tilde{\mathbf{u}}(t) = -R^{-1}B^TP\tilde{\mathbf{x}}(t),$$

where the matrix $P(t)$ is a solution of the Ricatti equation

$$P' = -Q - A^TP - PA + PBR^{-1}B^TP, \quad P(t_f) = M \tag{19.8}$$

Proof. All that we need to do is extend the definition of the matrix $P(t)$ obtained previously from the fundamental matrix, to see that it satisfies this matrix differential equation. This follows because the Hamiltonian is given by

$$H = \mathbf{p}^TA\mathbf{x} + \mathbf{p}^TB\mathbf{u} - \mathbf{x}^TQ\mathbf{x} - \mathbf{u}^TR\mathbf{u}$$

indicating that:

$$
\begin{aligned}
\mathbf{p} &= -2P\mathbf{x} \\
\Rightarrow -A^T\mathbf{p} + 2Q\mathbf{x} &= -\frac{DH}{D\mathbf{x}} \\
\Rightarrow \mathbf{p}'(t) &= -2P'\mathbf{x} - 2P\mathbf{x}' \\
&= -2P'\mathbf{x} - 2PA\mathbf{x} - 2PB\mathbf{u} \\
&= -2P'\mathbf{x} - 2PA\mathbf{x} + 2PBR^{-1}B^TP\mathbf{x} \\
&= 2(-P' - PA + PBR^{-1}B^TP)\mathbf{x} \\
\Rightarrow 2(A^TP + Q)\mathbf{x} &= 2(-P' - PA + PBR^{-1}B^TP)\mathbf{x} \\
\Rightarrow (A^TP + Q)\mathbf{x} &= (-P' - PA + PBR^{-1}B^TP)\mathbf{x},
\end{aligned}
$$

which is satisfied for nontrivial $\mathbf{x}(t)$ if and only if

$$P' = -Q - A^TP - PA + PBR^{-1}B^TP.$$

The end point condition comes from the previous derivation of $P(t)$ in terms of the fundamental solution. □

Remark 19.1.3. At first glance it seems that this doesn't make our life any easier because the Ricatti differential equation is a matrix differential equation whose right hand side is quadratic in $P(t)$. Solutions of such a system are certainly non-trivial, but it turns out that we can 'reduce' this $m \times m$ quadratic system to a linear $2m \times 2m$ system (see the exercises, where else would such a derivation take place?).

In the autonomous case, there are some solvers that will automatically calculate $K = R^{-1}B^TP$ (the response matrix) given the appropriate A, B, Q, R, M. Although this is certainly non-trivial it is actually quite fast, even for rather high dimensions, and hence gives the solution of the LQR problem not only in closed form, but algorithmically very fast.

The primary reason for using LQR is that there are remarkably fast solvers for Riccati equations, and these make our life much easier than we would expect to see from a standard optimal control problem. Implementing and using these solvers can be a mess unto itself, but is a critical aspect of any control problem.

19.2 A few examples

We set up a few examples to demonstrate the true utility of the LQR solution. In the next section we talk about setting up the numerical solution via the Riccatti equation. These examples will re-appear in later sections of the text, as we investigate other potential ways to model these key problems. We start off with a handful of medically oriented examples, before dedicating a very long discussion and derivation to the inverted pendulum, which is one of the most classical control problems of all time.

Example 19.2.1 (Insulin Pump). Let $x(t)$ be the glucose level in the blood stream. Suppose that for a diabetic, the body is unable to regulate glucose levels, and hence the glucose blood level builds at a rate proportional to the current amount at a given time, this means that $x' = \alpha x - u$ where $\alpha > 0$, and in order to maintain a certain 'safe' glucose level, insulin is injected at a rate $u(t)$. The problem is that insulin is expensive, but more so, too much insulin can dangerously lower the glucose level, and so we must have a cost attached to the actual insulin level. There are a lot of ways we could phrase this, but one of them is discussed in the exercises at great length, and demonstrates how we can set this up as an LQR problem (the evolution equation given here is already linear).

Example 19.2.2 (Insulin Pump version 2.0). In reality the blood glucose level $x_1(t)$ is actually dependent on the net hormonal concentration $x_2(t)$ which is how insulin is actually injected. This means that the true evolution is given by the coupled system:

$$x_1' = -ax_1 - bx_2, \tag{19.9}$$

$$x_2' = -cx_2 + dx_1 + u, \tag{19.10}$$

where the constants a, b, c, d are all nonnegative. A standard approximation for a diabetic patient is that $d \approx 0$. This provides a more realistic, coupled model of the blood glucose response, but the question (left as an exercise of course) is, how do we implement it as an LQR problem?

Example 19.2.3 (Chemotherapy). Let $x(t)$ be the number of cancerous tumor cells at time t (with an exponential growth rate) and let $u(t)$ be the chemotherapy drug concentration. The desired outcome is of course to minimize both the end state $x(t_f)$, but to also limit the potential damage caused by to much $u(t)$ i.e. keep $u(t)$ to a minimum throughout the entire treatment interval $[t_0, t_f]$.

19.2.1 The inverted pendulum, and an infinite horizon

Another method to reduce the complexity of solving the Riccatti equation is to consider the case where we are optimizing over an infinite time horizon, i.e. $t_f \to \infty$. This is practical when we are trying to optimize something either over a very long interval, or perhaps when we are trying to maintain the optimal solution for an indefinite amount of time. This is exemplified in the canonical example of the inverted pendulum, an example so important that we devote this entire section to it.

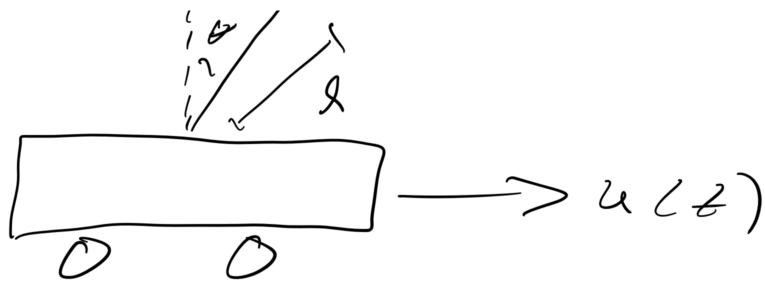

Figure 19.1: The setup for the inverted pendulum of Example 19.2.4.

Example 19.2.4. We will consider an inverted pendulum attached via a massless rod of length l to a cart that can move back and forth in a single horizontal direction. For now we will only let the pendulum have one degree of freedom, in the same direction as the motion of the cart. The control problem is to maintain the stability of the inverted pendulum purely by motion of the cart back and forth. Inherently we naturally notice that the system is entirely unstable, as the natural condition of the pendulum will be to fall as far as possible, and hence the cart's motion is the only thing that can keep the pendulum upright.

The derivation of the control problem follows these steps:

(i) Derivation of the equations of motion via a modified Hamilton's Principle for the pendulum with control (acceleration of the cart) $u(t)$.

(ii) Linearization of the resultant equations of motion.

(iii) Setup of the cost functional.

(iv) Solution of the resultant LQR problem.

We begin by considering the derivation of the equations of motion. Let

- v_1: velocity of the cart.
- v_2: Velocity of the pendulum.

The kinetic energy of this system will be $T = \frac{1}{2}mv_2^2 + \frac{1}{2}Mv_1^2$, where we are allowing m to be the mass of the pendulum (ignoring the mass of the rod for now) and M is the mass of the cart itself. If we specify the cart being at $y = 0$ in the plane, with horizontal position $x(t)$ then we can determine the position of the mass at the end of the rod as having coordinates $(x + l\sin\theta, l\cos\theta)$.

It follows that

$$\begin{aligned}
v_1^2 &= x'^2, \\
v_2^2 &= \left(\frac{d}{dt}[x + l\sin(\theta)]\right)^2 + \left(\frac{d}{dt}[l\cos(\theta)]\right)^2 \\
&= (x' + l\theta'\cos(\theta))^2 + (-l\theta'\sin(\theta))^2 \\
&= x'^2 + l^2\theta'^2 + 2lx'\theta'\cos(\theta).
\end{aligned}$$

The potential energy is given by

$$U = mgl\cos(\theta).$$

The difference between this problem and the standard setup with Hamilton's Principle is that in this setting we have the additional control variable u which is the acceleration/force that we apply to the cart with $u > 0$ indicating acceleration to the right, and $u < 0$ applying acceleration to the left.

As noted above, we need to modify the Lagrangian to account for the presence of our external forcing mechanism. This is because Hamilton's Principle as derived in these notes is applicable to systems in the absence of an external force. When the force is present, we need to include an additional potential that denotes its impact. The force $u(t)$ is induced by the potential $V(x,u) = -ux$, i.e. the force can be written as $u = -\partial_x V$ which can be included in the statement of the potential energy so that we will define the Lagrangian as

$$L = T - U - V = \frac{1}{2}(m+M)x'^2 + \frac{1}{2}ml^2\theta'^2 - mlx'\theta'\cos(\theta) - mgl\cos(\theta) + ux.$$

Note that our generalized coordinates here are $x(t)$ and $\theta(t)$, rather than $x(t)$ and $y(t)$. This leads to the Euler-Lagrange equations:

$$\begin{aligned}
\frac{\partial L}{\partial x} - \frac{d}{dt}\frac{\partial L}{\partial x'} &= 0 \\
\Rightarrow u &= \frac{d}{dt}\left[(m+M)x' - ml\theta'\cos\theta\right] \\
\Rightarrow u &= (m+M)x'' - ml\theta''\cos\theta + ml\theta'^2\sin\theta
\end{aligned}$$

and

$$\begin{aligned}
&\frac{\partial L}{\partial \theta} - \frac{d}{dt}\frac{\partial L}{\partial \theta'} = 0 \\
&\Rightarrow mlx'\theta'\sin(\theta) + mgl\sin(\theta) - \frac{d}{dt}\left[ml^2\theta' - mlx'\cos(\theta)\right] = 0 \\
&\Rightarrow (x'\theta' + g)\sin\theta = \frac{d}{dt}\left(l\theta' - x'\cos\theta\right) \\
&= l\theta'' - x''\cos\theta + x'\theta'\sin\theta \\
&\Rightarrow l\theta'' = g\sin\theta + x''\cos\theta
\end{aligned}$$

Hence the full nonlinear equations of motion for the inverted pendulum attached to a cart are:

$$u = (m+M)x'' - ml\theta''\cos\theta + ml\theta'^2\sin\theta, \tag{19.11}$$

$$l\theta'' = g\sin\theta + x''\cos\theta. \tag{19.12}$$

Very clearly this system is not linear, and hence no matter what the cost function we come up with, will not match the LQR designation. Because we are trying to maintain the pendulum about the point $\theta = 0$ then it is reasonable to linearize the evolution of the system around this point. This means that $\cos\theta \approx \theta$ and $\sin\theta \approx 0$. We also assume that $\theta' \approx 0$ meaning the system is nearly in total equilibrium and we are just providing small adjustments to keep the pendulum upright. Linearization leads to:

$$\begin{aligned} u &= (m+M)x'' - ml\theta'' \\ l\theta'' &= g\theta + x''. \end{aligned}$$

Now inserting the second equation into the first, we have

$$\begin{aligned} u &= (m+M)x'' - mg\theta - mx'' \\ &= Mx'' - mg\theta \end{aligned}$$

so that

$$x'' = \frac{1}{M}u + \frac{m}{M}g\theta. \tag{19.13}$$

Inserting this into the 2nd equation we have

$$l\theta'' = g\theta + \frac{1}{M}u + \frac{m}{M}g\theta$$

which gives us

$$\theta'' = \frac{1}{Ml}u + \frac{g}{l}\left(1 + \frac{m}{M}\right)\theta. \tag{19.14}$$

Thus, we have a linear system of equations, but we want to consider it as a first order system, so we let $x_1 = x$, $x_2 = x'$, $\theta_1 = \theta$, and $\theta_2 = \theta'$. Then

$$\begin{aligned} x_1' &= x_2, \\ x_2' &= \frac{1}{M}u + \frac{m}{M}g\theta_1, \\ \theta_1' &= \theta_2, \\ \theta_2' &= \frac{1}{Ml}u + \frac{g}{l}\left(1 + \frac{m}{M}\right)\theta_1. \end{aligned}$$

Let

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \theta_1 \\ \theta_2 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{m}{M} \end{pmatrix}, A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{l}(1+\frac{m}{M}) & 0 \end{pmatrix}$$

So $\mathbf{x}' = A\mathbf{x} + Bu$.

Let our cost functional be given by

$$J[u] = \int_0^\infty (q_1\theta_1^2 + q_2\theta_2^2 + q_3x_1^2 + q_4x_2^2 + ru^2)dt$$

- If $r >> q_k$ (for $k = 1, 2, 3, 4$) then the control is very expensive, so the response will be very slow
- If $r << q_k$ then the controller is relatively cheap and so the response is fast

We can write the cost functional as

$$J[u] = \int_0^\infty (\mathbf{x}^T Q\mathbf{x} + ru^2)dt$$

It takes some careful analysis to justify this limit $t_f \to \infty$. This is referred to as the infinite horizon problem.

Remark 19.2.5. The inverted pendulum attached to a cart seems like a rather boring example, but the generic application of these principles is quite broad. One may even consider how Yoda trained Luke on Dagobah. At one point in the training we see Luke balancing himself on one hand while holding several objects. This is the epitome of an inverted pendulum. Another example is how BB-8 can remain stationary without rolling on any type of an incline.

For these and other related examples, the applied force or control need not necessarily be a horizontal motion of the base of the pendulum. Sometimes this stability is maintained instead through oscillatory motion induced on the rod that holds the pendulum's mass (poor Bob remains out on a limb). In face this is really how humans and other bipeds walk on only two legs (if you think about how counter-intuitive such a concept is then this will make sense). No wonder it takes newborns so long to learn how to walk...it has only taken us most of two semesters to get there).

Remark 19.2.6. The real power in the inverted pendulum isn't so much in the pendulum itself, but in the principle of maintaining stability of an inherently unstable physical setting. This is exactly what 'hoverboards' and unicycles do, narrowly maintaining a stable configuration so that even a slight nudge of the control will produce a significant, rapid physical motion due to the underlying physical dynamics independent of the control itself, i.e. the inherent instability will drive the system to some very rapid motion. This type of design is fundamental to many different engineering systems, the Grumman X-29 fighter jet is perhaps the most publicized, but it is only one of several examples.

19.2.2 Infinite Horizon Problem

The infinite horizon LQR problem is defined as

$$\mathbf{x}' = A\mathbf{x} + B\mathbf{u},$$

with cost functional

$$J[\mathbf{u}] = \int_0^\infty (\mathbf{x}^T Q\mathbf{x} + \mathbf{u}^T R\mathbf{u})dt,$$

where now A, B, Q, R are independent of time, and the optimal control $\tilde{\mathbf{u}}(t) = -R^{-1}B^T P\mathbf{x}(t)$ where P satisfies the algebraic Riccati equation:

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \tag{19.15}$$

and

$$\tilde{\mathbf{x}}' = (A - BR^{-1}B^T P)\tilde{\mathbf{x}}$$

is the evolution of the optimal state vector.

Remark 19.2.7. The inverted pendulum and related discussion do not quite adequately express the true utility of using an infinite horizon LQR model. In practice there are very few control problems that are tractable, let alone with a closed form solution such as here, and the corresponding increase in performance and speed when an optimization problem is translated into LQR, is immeasurable. In fact, most optimization problems in practice are demanded to be LQR, i.e. this is a choice in the modeling phase of the problem. If the state equation is inherently nonlinear, then the optimization is performed over sufficiently small time intervals (infinite horizon probably wouldn't make as much sense here) so that linearizing the state equation is practical.

The truly remarkable thing is that even though an infinite horizon doesn't seem natural for many problems, it often works in practice despite being applied over a very short time interval.

19.3 Some computable examples

This section is dedicated to numerical solutions of the LQR problem both for finite and infinite horizons. We begin of course, by example:

Example 19.3.1 (Chemotherapy solution). We return to the chemotherapy example, where the population of cancer cells grows according to: $x' = \alpha x - u$ with $x(0) = 1$. We set this problem up as an LQR problem by specifying the cost functional

$$J[u] = \int_0^{t_f} u^2 dt + mx(t_f)^2,$$

which will minimize the effects of the chemotherapy throughout the treatment time $[0, t_f]$ and minimize the final population of cancer cells at the end of the treatment.

To identify the LQR solution of this problem, we note that according to the canonical symbolic derivation obtained previously, in this problem $A = \alpha,\ B = -1,\ Q = 0,\ R = 1$ and $M = m$. To make this explicit, we will only compute the solution for $\alpha = 2$ and $m = 1$. Variations of this setup are addressed in the exercises. We need to set up the solution of the Riccatti equation, but rather than trying to solve the full quadratic ODE, we use the trick derived in the exercises, and solve for $(x(t), y(t))^T$, and note that $p(t) = -\frac{1}{2}y(t)/x(t)$.

To proceed, first let

$$\mathscr{H}(t) = \begin{pmatrix} 2 & \frac{1}{2} \\ 0 & -2 \end{pmatrix}.$$

Then we know that the vector $\mathbf{x}(t) = (x(t), y(t))^T$ satisfies $\mathbf{x}' = \mathscr{H}\mathbf{x}$ with $\mathbf{x}(t_f) = (1, -2)^T$. This in turn leads to the optimal control $\tilde{u}(t) = p(t)\tilde{x}(t)$, i.e. the optimal state will satisfy the differential equation

$$x' = (2 - p)x,\ x(0) = 1.$$

The solution can be implemented using Scipy's ivp_solve functionality as demonstrated in Algorithm 19.1.

```python
import numpy as np
from matplotlib import pyplot as plt
from scipy.integrate import solve_ivp
#setup the evolution equation for the Riccatti
#equation (the linear 2D version)
def riccati(t,x):
    dxdt = [2*x[0]+.5*x[1],-2*x[1]]
    return dxdt

#Solve the Riccatti equation, and make sure to
#save the output densely so we can interpolate later on
xf = [1,-2]#solve backward from tf=1
p_sol = solve_ivp(riccati,[1,0],xf,dense_output=True)

#Assuming that we have an ODE solution to the Riccatti
#equation then this is the evolution of the cancer cell population
def cancer_evolve(t,x,p_sol):
    X,Y = p_sol.sol(t)
    p = -.5*Y/X
    return (2-p)*x

#Solve the forward model for the cancer cells,
#then compute the optimal control from the optimal state, and plot the results
final_sol = solve_ivp(cancer_evolve,[0,1],[1],args=[p_sol],dense_output=True)
t = np.linspace(0,.5,101)
x = final_sol.sol(t)
X,Y = p_sol.sol(t)
p = -.5*Y/X
u = p*x
plt.plot(t,np.squeeze(x))
plt.plot(t,np.squeeze(u))
plt.legend(['cancer cell population','chemotherapy concentration'])
```

Algorithm 19.1: Implementation for solving the cancer-chemotherapy problem as specified in Example 19.3.1.

With the setup used in Example 19.3.1 we note that the solution can be found quite efficiently. The solution of the two necessary ODE's is relatively straightforward and costs very little in terms of computational resources. This is of course an overly simplified version of everything as we are dealing with a one dimensional system. In practice we will be concerned with a larger dimensional state space which will then bog down the solution of the corresponding ODEs. This setting is exactly why we often consider the infinite horizon approximation.

Example 19.3.2 (Chemotherapy again). Returning to the previous example, we now suppose that we are working on the semi-infinite interval $[0, \infty)$. Although in practice this may seem extremely unreasonable, i.e. chemotherapy for a lifetime, we must recall that the infinite horizon is often just an approximation of the relative temporal scales. For instance, certain cancers react extremely rapidly to specific chemotherapy treatments, so that a few days may be reasonably approximated as an infinite time interval.

To justify using the infinite horizon, we necessarily need to include a quadratic cost of the cancer cells in the time integral, i.e. we now write the cost function as:

$$J[u] = \int_0^\infty \left[Qx^2 + Ru^2\right] dt.$$

This leads to the algebraic Riccati equation

$$2\alpha p + Q - \frac{1}{R}p^2 = 0,$$

which for this particular setting has the simple quadratic solution

$$p = \alpha R \pm R\sqrt{\alpha^2 + Q^2}.$$

Thus the optimal control will be $u(t) = \frac{1}{R}px(t)$ and the optimal state satisfies

$$x' = \left(\alpha - \frac{p}{R}\right) x, \quad x(0) = x_0.$$

This can be simplified to:

$$x' = \mp \left(\sqrt{\alpha^2 + Q^2}\right) x, \quad x(0) = x_0.$$

Clearly only one of these roots makes sense, i.e. even though the Riccati equation gives us two separate solutions, only the positive root (decaying solution) physically makes sense, giving us the exponentially decaying number of cancer cells $x(t) = x_0 e^{-\sqrt{\alpha^2+Q^2}t}$. The corresponding chemotherapy prescription should also follow $u(t) = \left(\alpha + \sqrt{\alpha^2 + Q^2}\right) x_0 e^{-\sqrt{\alpha^2+Q^2}t}$.

Remark 19.3.3. The models present thus far for the influence of chemotherapy on the growth of cancer cells is of course overly simplistic. Reality is much more complicated. In particular, one will notice rather readily that the first linear model with a finite horizon allows for a negative total number of cancer cells. In addition, both of these models completely ignore much of the basic physiology that defines how cancer grows, and how the immune system responds. It is not likely that the cancer can be completely eliminated by a single chemotherapy regimen so it is common to use several drugs that have differing negative effects (increasing the cost), but also have very different effects on the growth rate or decay of different types of cancers. There is a lot more to be done here, and this is clearly only the tip of a very large iceberg, but nevertheless the principle of balancing the need to eliminate cancerous cells against the need to not poison the patient to much is clearly laid out in this formalism. We just caution against taking any of these models to seriously without careful comparison with actual data, and input from trained oncologists.

We consider one final example wherein the Riccati equation is not just a quadratic equation in one dimension to demonstrate a numerical method of solution that is built in to scipy.

Example 19.3.4. [Driving a car with LQR] We return to the problem of driving a car along a straight line. The dynamics are given by:

$$\begin{aligned} x' &= y, \quad x(0) = -x_0 \\ y' &= u, \quad y(0) = 0, \end{aligned}$$

and we will seek to minimize the cost

$$J[u] = \int_0^\infty \left[q_0 x^2 + q_1 y^2 + Ru^2\right] dt,$$

where $q_0, q_1 \geq 0$ are trying to limit how fast we go (q_1) while also trying to drive us closer to the origin (q_0), and $R > 0$ indicates that fuel costs something (and perhaps we are uncomfortable with sudden acceleration or deceleration). In the language of the infinite horizon LQR setup, this identifies the following matrices:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad Q = \begin{pmatrix} q_0 & 0 \\ 0 & q_1 \end{pmatrix}.$$

From this, we need to solve the algebraic Riccati equation (19.15), and then insert the solution matrix P back into the solution for the optimal control. This is shown in Algorithm 19.2, where we use the built in continuous algebraic Riccati solver (*care*). This allows for a clean computation of the optimal state evolution for this type of problem.

```python
import numpy as np
from matplotlib import pyplot as plt
from scipy.integrate import solve_ivp
from scipy import linalg


def car_driving_LQR(q0,q1,R,x0):
    #First set up all the relevant matrices
    A=np.array([[0,1],[0,0]])
    B=np.array([[0],[1]])
    Q=np.array([[q0,0],[0,q1]])
    R=1
    #solve the continuous algebraic Riccati equation
    P=linalg.solve_continuous_are(A,B,Q,R)

    #setup the evolution equation with optimal control selected
    def car_evolve(t,y):
        return (A-(1/R)*B@B.T@P)@y

    #solve the optimal state evolution, and plot the position & velocity
    sol = solve_ivp(car_evolve,[0,10],[-x0,0],dense_output=True)
    t = np.linspace(0,10,1001)
    x,y = sol.sol(t)
    plt.plot(t,x)
    plt.plot(t,y)
```

Algorithm 19.2: Solution of the infinite horizon version of the car driving problem in Example 19.3.4.

Exercises

Note to the student: Each section of this chapter has several corresponding exercises, all collected here at the end of the chapter. The exercises between the first and second line are for Section 1, the exercises between the second and third lines are for Section 2, and so forth.

You should **work every exercise** (your instructor may choose to let you skip some of the advanced exercises marked with *). We have carefully selected them, and each is important for your ability to understand subsequent material. Many of the examples and results proved in the exercises are used again later in the text. Exercises marked with ⚠ are especially important and are likely to be used later in this book and beyond. Those marked with † are harder than average, but should still be done.

Although they are gathered together at the end of the chapter, we strongly recommend you do the exercises for each section as soon as you have completed the section, rather than saving them until you have finished the entire chapter.

19.1. Derive (19.6) from (19.5).

19.2. Let $X(t)$ and $Y(t)$ be $n \times n$ matrices satisfying the linear differential equation

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \mathscr{H}(t) \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix}$$

with the boundary condition $X(t_f) = I,\ Y(t_f) = -2M$ where

$$\mathscr{H}(t) = \begin{pmatrix} A(t) & \frac{1}{2}B(t)R^{-1}(t)B^T(t) \\ 2Q(t) & -A^T(t) \end{pmatrix}.$$

Prove that $P(t) = -\frac{1}{2}Y(t)X^{-1}(t)$ satisfies the Riccati differential equation

$$P'(t) = -P(t)A(t) - A^T(t)P(t) - Q(t) + P(t)B(t)R^{-1}(t)B^T(t)P(t)$$

with the endpoint condition $P(t_f) = M$.

19.3. Set up an LQR problem to find a solution to the car-driving problem (see Example 18.1.2 for instance) where the cost is now quadratic in both the control (acceleration) and the velocity of the car, and is weighted equally at all times throughout the time interval of interest. Your goal in this case is to minimize the time it takes to reach the destination, and to keep the fuel expended to a minimum.

19.4. Set up the solution (via LQR) for the blood glucose problem where the state evolution is $x' = -\alpha x + u$ and the cost is $J[u] = \int_0^{t_f} u^2 dt$, and $x(t_f) = l$.

19.5. Formulate the LQR solution (but don't worry about solving the Riccati equation explicitly) for the advanced blood-glucose problem, i.e.

$$\begin{aligned} x_1' &= -ax_1 - bx_2 \\ x_2' &= -cx_2 + u, \end{aligned}$$

$x_1(0) = x_0,\ x_2(0) = 0$, with cost functional given by

$$J[u] = \int_0^{t_f} \left[\alpha x_1^2 + u^2\right] dt.$$

19.6. Modify the inverted pendulum setup to include two horizontal directions of motion i.e. the pendulum is free to move in the x and y directions, and so is the cart. This will introduce two angles θ and ϕ that can dictate the motion as in spherical coordinates. Derive the full (nonlinear evolution equations) in this case.

19.7. *Linearize the system of equations from the previous example, and set up the LQR solution for the same. How large is the system now that you are concerned with both horizontal directions?

19.8. Modify Algorithm 19.1 so that it will work for any choice of the parameters α, and different weights R, Q, and M. Then create a plot of the optimal solution $x(t)$ and optimal control $u(t)$ for $\alpha = 3$, $R = 4$, and $Q = 1$ with $M = 1$ as well. Solve this on the time interval $[0, 1]$.

19.9. Create a combination of Algorithms 19.1 and 19.2 to create a solver for the car driving problem, but now with a finite horizon on the interval $[0, 5]$ and an additional endpoint cost to force the solution to reach the origin. Create plots of the solution in $x(t)$ (position) and $y(t)$ (velocity).

19.10. Create a solver for the updated glucose-insulin problem (Exercise 19.5 above) with an infinite horizon, i.e. use the algebraic Riccati equation. Calculate the solution and corresponding plots of the state for $a = 1$, $b = 2$, $c = 1.5$ and $\alpha = 2$.