Lecture 2
LQR via Lagrange multipliers

- useful matrix identities
- linearly constrained optimization
- LQR via constrained optimization

Some useful matrix identities

let's start with a simple one:

$$Z(I+Z)^{-1} = I - (I+Z)^{-1}$$

(provided $I+Z$ is invertible)

to verify this identity, we start with

$$I = (I+Z)(I+Z)^{-1} = (I+Z)^{-1} + Z(I+Z)^{-1}$$

re-arrange terms to get identity

an identity that's a bit more complicated:

$$(I + XY)^{-1} = I - X(I + YX)^{-1}Y$$

(if either inverse exists, then the other does; in fact $\det(I + XY) = \det(I + YX)$)

to verify:

$$\begin{aligned}\left(I - X(I+YX)^{-1}Y\right)(I+XY) &= I + XY - X(I+YX)^{-1}Y(I+XY) \\ &= I + XY - X(I+YX)^{-1}(I+YX)Y \\ &= I + XY - XY = I\end{aligned}$$

another identity:

$$Y(I + XY)^{-1} = (I + YX)^{-1}Y$$

to verify this one, start with $Y(I + XY) = (I + YX)Y$

then multiply on left by $(I + YX)^{-1}$, on right by $(I + XY)^{-1}$

- note dimensions of inverses not necessarily the same
- mnemonic: lefthand Y moves into inverse, pushes righthand Y out . . .

and one more:

$$(I + XZ^{-1}Y)^{-1} = I - X(Z + YX)^{-1}Y$$

let's check:

$$\begin{aligned}\left(I + X(Z^{-1}Y)\right)^{-1} &= I - X\left(I + Z^{-1}YX\right)^{-1}Z^{-1}Y \\ &= I - X(Z(I + Z^{-1}YX))^{-1}Y \\ &= I - X(Z + YX)^{-1}Y\end{aligned}$$

Example: rank one update

- suppose we've already calculated or know A^{-1}, where $A \in \mathbf{R}^{n \times n}$
- we need to calculate $(A + bc^T)^{-1}$, where $b,\ c \in \mathbf{R}^n$
  ($A + bc^T$ is called a *rank one update* of A)

we'll use another identity, called *matrix inversion lemma*:

$$(A + bc^T)^{-1} = A^{-1} - \frac{1}{1 + c^T A^{-1} b}(A^{-1}b)(c^T A^{-1})$$

note that RHS is easy to calculate since we know A^{-1}

more general form of matrix inversion lemma:

$$(A+BC)^{-1} = A^{-1} - A^{-1}B\left(I+CA^{-1}B\right)^{-1}CA^{-1}$$

let's verify it:

$$\begin{aligned}
(A+BC)^{-1} &= \left(A(I+A^{-1}BC)\right)^{-1} \\
&= (I+(A^{-1}B)C)^{-1}A^{-1} \\
&= \left(I-(A^{-1}B)(I+C(A^{-1}B))^{-1}C\right)A^{-1} \\
&= A^{-1} - A^{-1}B(I+CA^{-1}B)^{-1}CA^{-1}
\end{aligned}$$

Another formula for the Riccati recursion

$$\begin{aligned}
P_{t-1} &= Q + A^TP_tA - A^TP_tB(R+B^TP_tB)^{-1}B^TP_tA \\
&= Q + A^TP_t\left(I - B(R+B^TP_tB)^{-1}B^TP_t\right)A \\
&= Q + A^TP_t\left(I - B((I+B^TP_tBR^{-1})R)^{-1}B^TP_t\right)A \\
&= Q + A^TP_t\left(I - BR^{-1}(I+B^TP_tBR^{-1})^{-1}B^TP_t\right)A \\
&= Q + A^TP_t\left(I+BR^{-1}B^TP_t\right)^{-1}A \\
&= Q + A^T\left(I+P_tBR^{-1}B^T\right)^{-1}P_tA
\end{aligned}$$

or, in pretty, symmetric form:

$$P_{t-1} = Q + A^TP_t^{1/2}\left(I+P_t^{1/2}BR^{-1}B^TP_t^{1/2}\right)^{-1}P_t^{1/2}A$$

Linearly constrained optimization

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Fx = g \end{array}$$

- $f : \mathbf{R}^n \to \mathbf{R}$ is smooth *objective function*
- $F \in \mathbf{R}^{m \times n}$ is fat

form *Lagrangian* $L(x, \lambda) = f(x) + \lambda^T(g - Fx)$ (λ is *Lagrange multiplier*)

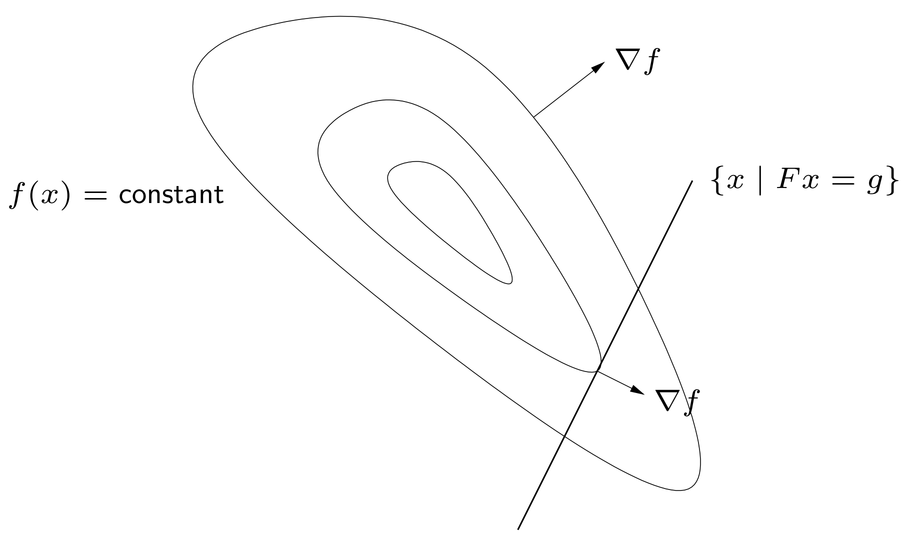
if x is optimal, then

$$\nabla_x L = \nabla f(x) - F^T \lambda = 0, \qquad \nabla_\lambda L = g - Fx = 0$$

i.e., $\nabla f(x) = F^T \lambda$ for some $\lambda \in \mathbf{R}^m$

(generalizes optimality condition $\nabla f(x) = 0$ for unconstrained minimization problem)

Picture



$\nabla f(x) = F^T \lambda$ for some $\lambda \iff \nabla f(x) \in \mathcal{R}(F^T) \iff \nabla f(x) \perp \mathcal{N}(F)$

Feasible descent direction

suppose x is current, feasible point ($i.e.$, $Fx = g$)

consider a small step in direction v, to $x + hv$ (h small, positive)

when is $x + hv$ better than x?

need $x + hv$ feasible: $F(x + hv) = g + hFv = g$, so $Fv = 0$

$v \in \mathcal{N}(F)$ is called a *feasible direction*

we need $x + hv$ to have smaller objective than x:

$$f(x + hv) \approx f(x) + h\nabla f(x)^T v < f(x)$$

so we need $\nabla f(x)^T v < 0$ (called a *descent direction*)

(if $\nabla f(x)^T v > 0$, $-v$ is a descent direction, so we need only $\nabla f(x)^T v \neq 0$)

x is not optimal if there exists a feasible descent direction

if x is optimal, every feasible direction satisfies $\nabla f(x)^T v = 0$

$$\begin{aligned}
Fv = 0 \;\Rightarrow\; \nabla f(x)^T v = 0 \quad &\Longleftrightarrow \quad \mathcal{N}(F) \subseteq \mathcal{N}(\nabla f(x)^T) \\
&\Longleftrightarrow \quad \mathcal{R}(F^T) \supseteq \mathcal{R}(\nabla f(x)) \\
&\Longleftrightarrow \quad \nabla f(x) \in \mathcal{R}(F^T) \\
&\Longleftrightarrow \quad \nabla f(x) = F^T \lambda \;\text{ for some }\; \lambda \in \mathbf{R}^m \\
&\Longleftrightarrow \quad \nabla f(x) \perp \mathcal{N}(F)
\end{aligned}$$

LQR as constrained minimization problem

$$\begin{array}{ll} \text{minimize} & J = \frac{1}{2}\sum_{t=0}^{N-1}\left(x(t)^T Q x(t) + u(t)^T R u(t)\right) + \frac{1}{2}x(N)^T Q_f x(N) \\ \text{subject to} & x(t+1) = Ax(t) + Bu(t), \quad t = 0, \ldots, N-1 \end{array}$$

- variables are $u(0), \ldots, u(N-1)$ and $x(1), \ldots, x(N)$ ($x(0) = x_0$ is given)
- objective is (convex) quadratic (factor $1/2$ in objective is for convenience)

introduce Lagrange multipliers $\lambda(1), \ldots, \lambda(N) \in \mathbf{R}^n$ and form Lagrangian

$$L = J + \sum_{t=0}^{N-1} \lambda(t+1)^T \left(Ax(t) + Bu(t) - x(t+1)\right)$$

Optimality conditions

we have $x(t+1) = Ax(t) + Bu(t)$ for $t = 0, \ldots, N-1$, $x(0) = x_0$

for $t = 0, \ldots, N-1$, $\nabla_{u(t)} L = Ru(t) + B^T\lambda(t+1) = 0$

hence, $u(t) = -R^{-1}B^T\lambda(t+1)$

for $t = 1, \ldots, N-1$, $\nabla_{x(t)} L = Qx(t) + A^T\lambda(t+1) - \lambda(t) = 0$

hence, $\lambda(t) = A^T\lambda(t+1) + Qx(t)$

$\nabla_{x(N)} L = Q_f x(N) - \lambda(N) = 0$, so $\lambda(N) = Q_f x(N)$

these are a set of linear equations in the variables

$$u(0), \ldots, u(N-1), \quad x(1), \ldots, x(N), \quad \lambda(1), \ldots, \lambda(N)$$

Co-state equations

optimality conditions break into two parts:

$$x(t+1) = Ax(t) + Bu(t), \qquad x(0) = x_0$$

this recursion for state x runs forward in time, with initial condition

$$\lambda(t) = A^T\lambda(t+1) + Qx(t), \qquad \lambda(N) = Q_f x(N)$$

this recursion for λ runs backwards in time, with final condition

- λ is called *co-state*
- recursion for λ sometimes called *adjoint system*

Solution via Riccati recursion

we will see that $\lambda(t) = P_t x(t)$, where P_t is the min-cost-to-go matrix defined by the Riccati recursion

thus, Riccati recursion gives clever way to solve this set of linear equations

it holds for $t = N$, since $P_N = Q_f$ and $\lambda(N) = Q_f x(N)$

now suppose it holds for $t+1$, $i.e.$, $\lambda(t+1) = P_{t+1}x(t+1)$

let's show it holds for t, $i.e.$, $\lambda(t) = P_t x(t)$

using $x(t+1) = Ax(t) + Bu(t)$ and $u(t) = -R^{-1}B^T\lambda(t+1)$,

$$\lambda(t+1) = P_{t+1}(Ax(t) + Bu(t)) = P_{t+1}(Ax(t) - BR^{-1}B^T\lambda(t+1))$$

so

$$\lambda(t+1) = (I + P_{t+1}BR^{-1}B^T)^{-1}P_{t+1}Ax(t)$$

using $\lambda(t) = A^T\lambda(t+1) + Qx(t)$, we get

$$\lambda(t) = A^T(I + P_{t+1}BR^{-1}B^T)^{-1}P_{t+1}Ax(t) + Qx(t) = P_tx(t)$$

since by the Riccati recursion

$$P_t = Q + A^T(I + P_{t+1}BR^{-1}B^T)^{-1}P_{t+1}A$$

this proves $\lambda(t) = P_tx(t)$

let's check that our two formulas for $u(t)$ are consistent:

$$\begin{aligned}
u(t) &= -R^{-1}B^T\lambda(t+1) \\
&= -R^{-1}B^T(I + P_{t+1}BR^{-1}B^T)^{-1}P_{t+1}Ax(t) \\
&= -R^{-1}(I + B^TP_{t+1}BR^{-1})^{-1}B^TP_{t+1}Ax(t) \\
&= -((I + B^TP_{t+1}BR^{-1})R)^{-1}B^TP_{t+1}Ax(t) \\
&= -(R + B^TP_{t+1}B)^{-1}B^TP_{t+1}Ax(t)
\end{aligned}$$

which is what we had before