# 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:

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

# 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

$$\begin{aligned}
P_{t-1} &= Q + A^T P_t A - A^T P_t B (R + B^T P_t B)^{-1} B^T P_t A \\
&= Q + A^T P_t \left( I - B(R + B^T P_t B)^{-1} B^T P_t \right) A \\
&= Q + A^T P_t \left( I - B((I + B^T P_t B R^{-1}) R)^{-1} B^T P_t \right) A \\
&= Q + A^T P_t \left( I - B R^{-1} (I + B^T P_t B R^{-1})^{-1} B^T P_t \right) A \\
&= Q + A^T P_t \left( I + B R^{-1} B^T P_t \right)^{-1} A \\
&= Q + A^T \left( I + P_t B R^{-1} B^T \right)^{-1} P_t A
\end{aligned}$$

or, in pretty, symmetric form:

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

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

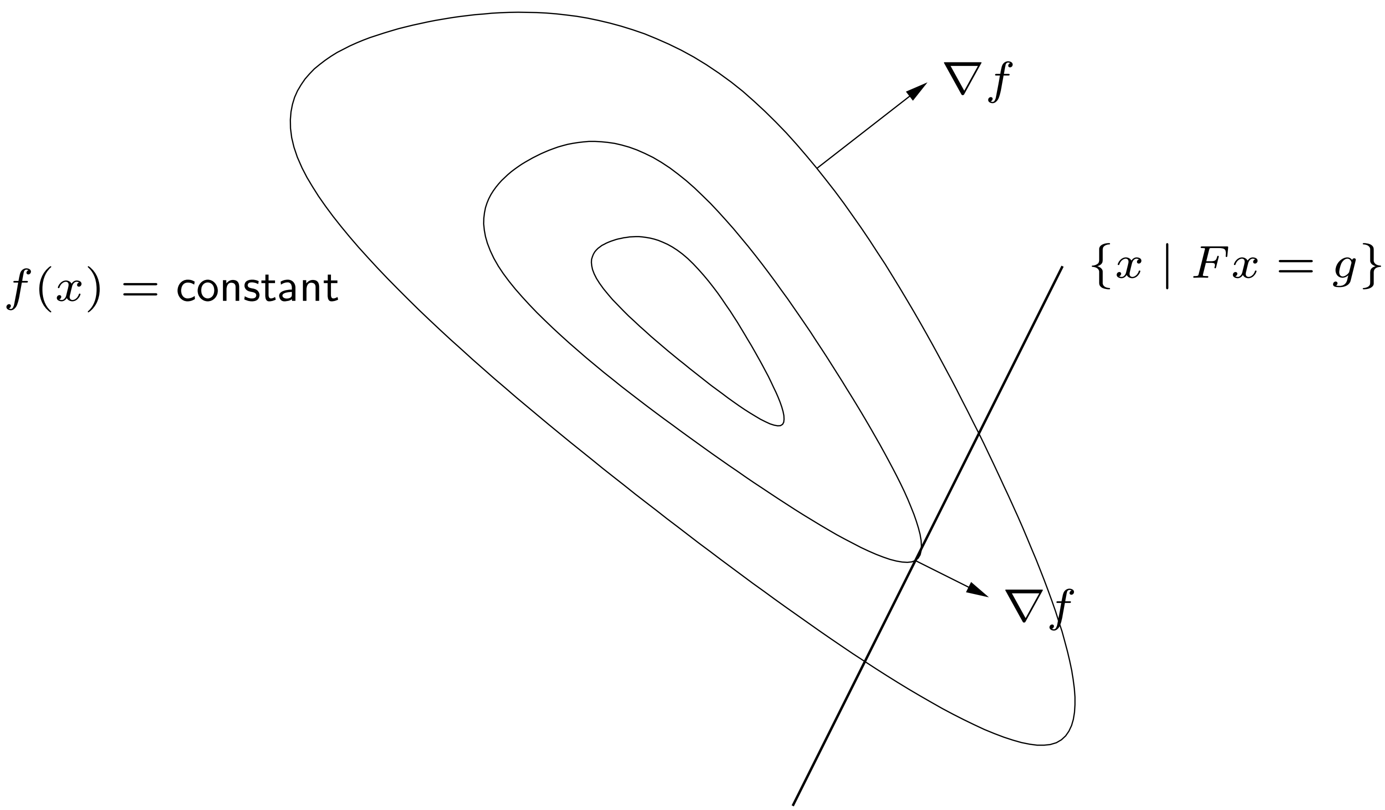
if $x$ is optimal, then

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

*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




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

# 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 $\nabla f(x)^T v = 0$

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

# LQR as constrained minimization problem

$$\begin{array}{ll} \text{minimize} & J = \frac{1}{2}\sum_{t=0}^{N-1}\left(x_t^T Q x_t + u_t^T R u_t\right) + \frac{1}{2} x_N^T Q_f x_N \\ \text{subject to} & x_{t+1} = A x_t + B u_t, \quad t = 0, \ldots, N-1 \end{array}$$

- variables are $u_0, \ldots, u_{N-1}$ and $x_1, \ldots, x_N$ ($x_0 = x^{\text{init}}$ 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

$$L = J + \sum_{t=0}^{N-1} \lambda_{t+1}^T \left(A x_t + B u_t - x_{t+1}\right)$$

# Optimality conditions

we have $x_{t+1} = Ax_t + Bu_t$ for $t = 0, \ldots, N-1$, $x_0 = x^{\mathrm{init}}$

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:

$$x_{t+1} = Ax_t + Bu_t, \qquad x_0 = x^{\mathrm{init}}$$

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 $\lambda$ runs backward in time, with final condition

- $\lambda$ is called *co-state*
- recursion for $\lambda$ 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