Lecture 5. Dynamic Programming and LQR

Ivan Papusha

CDS270–2: Mathematical Methods in Control and System Engineering

April 27, 2015

Logistics

- hw4 due this **Wed, Apr 29**
  - do an easy problem or CYOA
- no midterm or final
- hw3 solutions posted online
- reading: Bertsekas, ch1 (Vol 1)
  - unfortunately, not available online (but great reference)
  - see also Bellman's 1957 book
- some very serious (hopefully doable) problems are starting to appear in the catalog

Dynamic Programming: why the name?

"Let's take a word that has an absolutely precise meaning, namely dynamic, in the classical physical sense. It also has a very interesting property as an adjective, and that it's impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It's impossible. Thus I thought dynamic programming was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities." —Richard Bellman[1]

[1] S. Drefus, "Richard Bellman on the Birth of Dynamic Programming," *Operations Research*, 50(6):48–51, 2002.

LQR optimization problem

Finite horizon *Linear Quadratic Regulator* problem:

$$
\begin{array}{ll}
\text{minimize} & \left(\sum_{t=0}^{T-1} x_t^T Q x_t + u_t^T R u_t\right) + x_T^T Q_T x_T \\
\text{subject to} & x_{t+1} = A x_t + B u_t, \quad t = 0, \ldots, T-1 \\
& x_0 = z
\end{array}
$$

- variables: $x_0, \ldots, x_T$, $u_0, \ldots, u_{T-1}$
- data: $Q \succeq 0$, $Q_T \succeq 0$, $R \succ 0$, initial state $z \in \mathbf{R}^n$, time horizon T
- equality constrained QP
- states and control inputs are unconstrained (aside from dynamics)

LQR optimization problem

minimize

$$\begin{bmatrix} x_0 \\ \vdots \\ x_{T-1} \\ x_T \end{bmatrix}^T \underbrace{\begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & Q_T \end{bmatrix}}_{\bar{Q}} \begin{bmatrix} x_0 \\ \vdots \\ x_{T-1} \\ x_T \end{bmatrix} + \begin{bmatrix} u_0 \\ \vdots \\ u_{T-1} \end{bmatrix}^T \underbrace{\begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix}}_{\bar{R}} \begin{bmatrix} u_0 \\ \vdots \\ u_{T-1} \end{bmatrix}$$

subject to

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_T \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ & & \ddots & \\ A^{(T-1)}B & A^{(T-2)}B & \cdots & B \end{bmatrix}}_{H} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{T-1} \end{bmatrix} + \underbrace{\begin{bmatrix} I \\ A \\ \vdots \\ A^{(T)} \end{bmatrix}}_{G} z$$

- variables: $\bar{x} = (x_0, \ldots, x_T)$, $\bar{u} = (u_0, \ldots, u_{T-1})$

LQR quadratic program

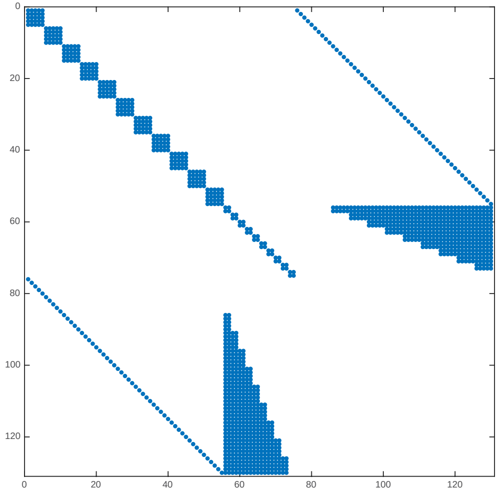
$$\begin{array}{ll} \text{minimize} & \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix}^T \begin{bmatrix} \bar{Q} & 0 \\ 0 & \bar{R} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} I & -H \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} = Gz \end{array}$$

- optimal primal and dual variables $(\bar{x}^\star, \bar{u}^\star)$ and $\bar{\nu}^\star$ satisfy

$$\begin{bmatrix} \begin{pmatrix} \bar{Q} + \bar{Q}^T & 0 \\ 0 & \bar{R} + \bar{R}^T \end{pmatrix} & \begin{pmatrix} I \\ -H^T \end{pmatrix} \\ \begin{pmatrix} I & -H \end{pmatrix} & 0 \end{bmatrix} \begin{bmatrix} \bar{x}^\star \\ \bar{u}^\star \\ \bar{\nu}^\star \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ Gz \end{bmatrix}$$

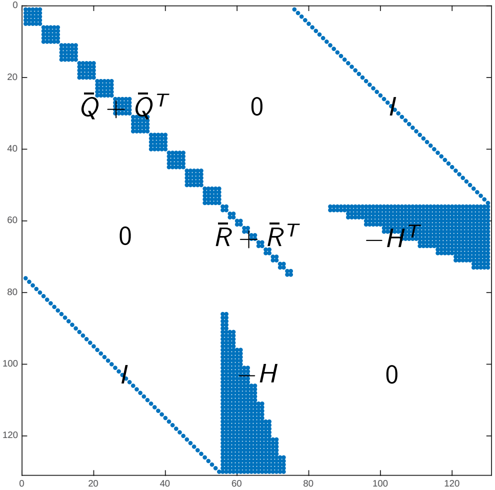
- dual variable $\bar{\nu}^\star$ is a sequence of *co-states*
- solution is given by (pseudo)inverse
- **bad solution technique** without sparse linear algebra

Example: LQR sparsity pattern



- $n = 5$ states
- $m = 2$ inputs
- $T = 10$ horizon

Example: LQR sparsity pattern



- $n = 5$ states
- $m = 2$ inputs
- $T = 10$ horizon

Identifying substructure

Define a function $V_t : \mathbf{R}^n \to \mathbf{R}$ to be the optimal value of the problem

$$\begin{array}{rll} V_t(z) := & \text{minimize} & \left(\sum_{\tau=t}^{T-1} x_\tau^T Q x_\tau + u_\tau^T R u_\tau\right) + x_T^T Q_T x_T \\ & \text{subject to} & x_{\tau+1} = Ax_\tau + Bu_\tau, \quad \tau = t, \dots, T-1 \\ & & x_t = z \end{array}$$

for every $t = 0, 1, \dots, T$

- V_t is a *value function* or *cost-to-go*
- $V_T(z) = z^T Q_T z$
- $V_0(x_0)$ is the optimal value of the original problem with initial condition $z = x_0$

Identifying substructure

$$V_{t-1}(z) = \left(\underset{\substack{u_{t-1}, u_t, \ldots, u_{T-1}, \\ x_{t-1}, x_t, \ldots, x_T, \\ x_T = Ax_{T-1} + Bu_{T-1} \\ \vdots \\ x_t = Ax_{t-1} + Bu_{t-1} \\ x_{t-1} = z}}{\text{minimize}} \left\{ \begin{array}{l} x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} \\ \quad + \sum_{\tau=t}^{T} x_\tau^T Q x_\tau + u_\tau^T R u_\tau \\ \quad + x_T^T Q_T x_T \end{array} \right\} \right)$$

$$= \underset{u_{t-1}=w, x_{t-1}=z}{\text{minimize}} \Bigg\{ z^T Q z + w^T R w + \underbrace{\underset{\substack{u_t, \ldots, u_{T-1}, \\ x_t, \ldots, x_T, \\ x_T = Ax_{T-1} + Bu_{T-1} \\ \vdots \\ x_t = Az + Bw}}{\text{minimize}} \left\{ \begin{array}{l} \sum_{\tau=t}^{T} x_\tau^T Q x_\tau + u_\tau^T R u_\tau \\ \quad + x_T^T Q_T x_T \end{array} \right\}}_{= V_t(Az + Bw)} \Bigg\}$$

$$= \underset{u_{t-1}=w, x_{t-1}=z}{\text{minimize}} \left\{ z^T Q z + w^T R w + V_t(Az + Bw) \right\}$$

Dynamic programming algorithm

1. initialize $V_T(z) := z^T Q_T z$ for all $z \in \mathbf{R}^n$
2. for each $t = T, T-1, \ldots, 1$ set

$$V_{t-1}(z) := \inf_w \left\{ z^T Q z + w^T R w + V_t(Az + Bw) \right\}, \qquad \text{for all } z \in \mathbf{R}^n$$

3. optimal value with initial condition $z = x_0$ is given by $V_0(x_0)$

Dynamic programming: LQR

LQR is special because the value functions are all quadratic.

$$V_t(z) = z^T P_t z, \quad \text{for all } z \in \mathbf{R}^n, \quad t = 0, \ldots, T$$

proof. (by induction)

- $V_T(z) = z^T Q_t z$ is quadratic in z
- assume $V_t(z) = z^T P_t z$ for all z, then

$$\begin{aligned}
V_{t-1}(z) &= \inf_w \left\{ z^T Q z + w^T R w + \underbrace{V_t(Az + Bw)}_{=(Az+Bw)^T P_t (Az+Bw)} \right\} \\
&= \inf_w \left\{ \begin{bmatrix} z \\ w \end{bmatrix}^T \begin{bmatrix} A^T P_t A + Q & A^T P_t B \\ B^T P_t A & B^T P_t B + R \end{bmatrix} \begin{bmatrix} z \\ w \end{bmatrix} \right\} \\
&= z^T \underbrace{\left(A^T P_t A + Q - (A^T P_t B)(B^T P_t B + R)^{-1}(B^T P_t A)\right)}_{=P_{t-1}} z
\end{aligned}$$

Riccati recursion

1. initialize $P_T := Q_T$
2. for each $t = T, T-1, \ldots, 1$ set

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

3. optimal value with initial condition $z = x_0$ is given by $x_0^T P_0 x_0$

at each step, the optimum input is given by

$$u_t^\star = \underbrace{-(B^T P_t B + R)^{-1} B^T P_t A}_{K_t} x_t$$

Continuous-time LQR

$$
\begin{array}{ll}
\text{minimize} & \int_0^T x(t)^T Q x(t) + u(t)^T R u(t)\, dt + x(T)^T Q_T x(T) \\
\text{subject to} & \dot{x}(t) = Ax(t) + Bu(t), \quad t \in (0, T) \\
& x(0) = z
\end{array}
$$

- variables: $x(\cdot)$, $u(\cdot)$
- data: $Q \succeq 0$, $Q_T \succeq 0$, $R \succ 0$, initial state $z \in \mathbf{R}^n$, time horizon T

Convert to discrete-time LQR

$$
\begin{array}{ll}
\text{minimize} & \left(\displaystyle\sum_{k=0}^{N-1} x_k^T Q x_k \delta + u_k^T R u_k \delta \right) + x_N^T Q_T x_N \\
\text{subject to} & \dfrac{x_{k+1} - x_k}{\delta} = A x_k + B u_k, \quad k = 0, \ldots, N-1 \\
& x_0 = z
\end{array}
$$

- split up $[0, T]$ into N time chunks, each of size $\delta = T/N$

$$
x_k = x(k\delta), \quad u_k = u(k\delta), \quad k = 0, 1, \ldots, N
$$

- take the left Riemann sums
- same as discrete time LQR with the replacements

$$
Q \leftrightarrow Q\delta, \quad R \leftrightarrow R\delta, \quad A \leftrightarrow (I + A\delta), \quad B \leftrightarrow B\delta
$$

Riccati recursion for continuous time LQR

- for each $k = N, N-1, \ldots, 1$ we have

$$\begin{aligned} P_{k-1} &= (I + A\delta)^T P_k (I + A\delta) + Q\delta \\ &- (I + A\delta)^T P_k B\delta \cdot ((B\delta)^T P_k (B\delta) + R\delta)^{-1} \cdot (B\delta)^T P_k (I + A\delta) \end{aligned}$$

$$\Downarrow$$

$$-\frac{P_k - P_{k-1}}{\delta} = A^T P_k + P_k A + Q - P_k B(\delta B^T P_k B + R)^{-1} B^T P_k + o(\delta)$$

- identify

$$-\frac{P_k - P_{k-1}}{\delta} \approx -\left.\frac{dP}{dt}\right|_{t=k\delta}$$

- and let $\delta \to 0$ and $N \to \infty$

Riccati differential equation for continuous LQR

The matrix $P(t) \in \mathbf{S}^n$ obeys the differential equation

$$-\frac{dP(t)}{dt} = A^T P(t) + P(t)A + Q - P(t)BR^{-1}B^T P(t), \quad t \in [0, T]$$
$$P(T) = Q_T.$$

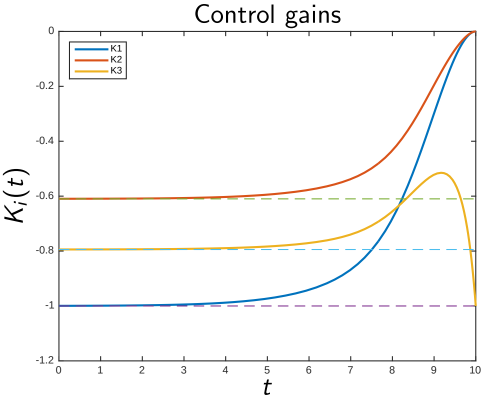
- the value function is a cost-to-go with initial condition $x(0) = z$

$$V_t(z) = z^T P(t) z$$

- optimum input at any time is a state feedback

$$u^\star(t) = \underbrace{-R^{-1}B^T P(t)}_{K(t)} x(t)$$

Example: LQR gains vs time



- $n = 3$ states
- $m = 1$ input
- $T = 10$ horizon
- dynamics:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & -1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- cost matrices:

$$Q = Q_T = I, \; R = 1$$

Infinite horizon LQR

$$\begin{array}{ll} \text{minimize} & \int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t)\, dt \\ \text{subject to} & \dot{x}(t) = Ax(t) + Bu(t), \quad t \geq 0 \\ & x(0) = z \end{array}$$

- variables: $x(\cdot)$, $u(\cdot)$
- data: $Q \succeq 0$, $Q_T \succeq 0$, $R \succ 0$, initial state $z \in \mathbf{R}^n$
- value function is time invariant $V_t(z) = z^T P z$, where $P \succ 0$ satisfies the Algebraic Riccati Equation,

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

- optimum input at any time is a time invariant state feedback

$$u^\star(t) = \underbrace{-R^{-1}B^T P}_{K} x(t)$$

Special terminal cost

Infinite horizon problem is like a finite horizon problem,

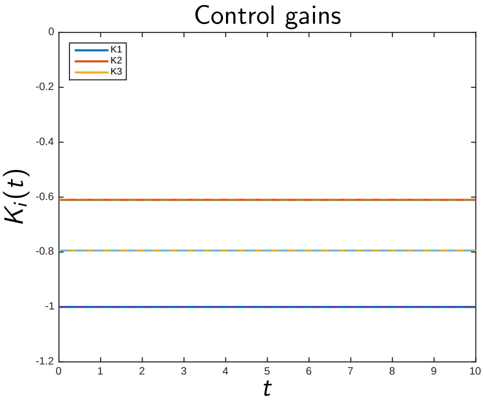
$$\begin{array}{ll} \text{minimize} & \int_0^T x(t)^T Q x(t) + u(t)^T R u(t)\, dt + x(T)^T Q_T x(T) \\ \text{subject to} & \dot{x}(t) = Ax(t) + Bu(t), \quad t \geq 0 \\ & x(0) = z, \end{array}$$

where the terminal matrix Q_T was chosen so that

$$x(T) Q_T x(T) = \int_T^\infty x(t)^T Q x(t) + u(t)^T R u(t)\, dt.$$

fact. in this case $Q_T = P$, where $A^T P + PA + Q - PBR^{-1}B^T P = 0$.

Example: LQR gains vs time



- $n = 3$ states
- $m = 1$ input
- $T = 10$ horizon
- dynamics:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- cost matrices:

$$Q = I, \; R = 1,$$

$$Q_T = \begin{bmatrix} 3.4042 & 1.7944 & 1.0000 \\ 1.7944 & 1.4985 & 0.6099 \\ 1.0000 & 0.6099 & 0.7944 \end{bmatrix}$$

Quadratic summary of future cost

In fact, we can find a matrix $P = P^T \succ 0$ such that

$$x(0)^T P x(0) = \int_0^\infty x(t)^T Q x(t) + u(t) R u(t)\, dt$$

for any initial condition $x(0) \in \mathbf{R}^n$.

fact. if the above integral exists, and we use time invariant state feedback $u(t) = Kx(t)$, then P satisfies the Lyapunov equation

$$(A + BK)^T P + P(A + BK) + (Q + K^T R K) = 0$$

- if $K = -R^{-1} B^T P$, this becomes the ARE

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

use case #1. use LQR to (attempt to) steer a linear system to a desired location $(x^{\mathrm{d}}, u^{\mathrm{d}})$ in state space.

LQR around a point

What if we minimize a cost around a point $(x^{\mathrm{d}}, u^{\mathrm{d}}) \neq (0,0)$?

- define new variables

$$\bar{x}(t) = x(t) - x^{\mathrm{d}}, \quad \bar{u}(t) = u(t) - u^{\mathrm{d}}$$

- cost in new variables

$$J = \left(\int_0^T \bar{x}(t)^T Q \bar{x}(t) + \bar{u}(t)^T R \bar{u}(t)\, dt \right) + \bar{x}(T)^T Q_T \bar{x}(T)$$

- new dynamics are affine, time-invariant

$$\dot{\bar{x}}(t) = \dot{x}(t) = A\bar{x}(t) + B\bar{u}(t) + \underbrace{(Ax^{\mathrm{d}} + Bu^{\mathrm{d}})}_{\text{fixed for all } t}$$

- optimal input is affine, $u(t) = u^{\mathrm{d}} + K_1(t)(x(t) - x^{\mathrm{d}}) + K_2(t)$

use case #2. use LQR to (attempt to) steer a real system to a desired trajectory in the presence of noise, disturbances...

(arguably the single most important method in robotics)

LQR around a trajectory

Suppose we have a nonlinear system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = z$$

and a method to generate a trajectory $x^{\mathrm{d}}(t)$, $u^{\mathrm{d}}(t)$ that satisfies the dynamics.

in practice. does implementing $u^{\mathrm{d}}(t)$ on a real system result in $x^{\mathrm{d}}(t)$?

no!

- physics modeling error
- discretization error
- unmodeled sensor noise, process noise...

LQR around a trajectory

Linearize nonlinear system around the desired trajectory

$$\underbrace{\dot{x}(t) - \dot{x}^{\mathrm{d}}(t)}_{\dot{\bar{x}}(t)} = \dot{x}(t) - f(x^{\mathrm{d}}(t), u^{\mathrm{d}}(t)) \approx$$

$$\underbrace{\frac{df}{dx}\Big|_{x^{\mathrm{d}}(t), u^{\mathrm{d}}(t)}}_{A(t)} \underbrace{(x(t) - x^{\mathrm{d}}(t))}_{\bar{x}(t)} + \underbrace{\frac{df}{du}\Big|_{x^{\mathrm{d}}(t), u^{\mathrm{d}}(t)}}_{B(t)} \underbrace{(u(t) - u^{\mathrm{d}}(t))}_{\bar{u}(t)}$$

- cost in new variables

$$J = \left(\int_0^T \bar{x}(t)^T Q \bar{x}(t) + \bar{u}(t)^T R \bar{u}(t) \; dt \right) + \bar{x}(T)^T Q_T \bar{x}(T)$$

- new dynamics are linear, time-varying
- optimal input is

$$u(t) = u^{\mathrm{d}}(t) + K(t)(x(t) - x^{\mathrm{d}}(t))$$

use case $\#3$. use LQR to (attempt to) find a nonlinear trajectory

Trajectory generation with LQR

Suppose we have a nonlinear system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = z$$

and we wish to find trajectory $x(t)$, $u(t)$ that satsifies the dynamics and minimizes cost

$$J = \int_0^T x(t)^T Q x(t) + u(t)^T R u(t) \, dt + x(T)^T Q_T x(T)$$

Iterative LQR

initialize $u^0(\cdot)$ to some guess

for each $k = 0, 1, 2, \ldots$

1. simulate $\dot{x}^k(t) = f(x^k(t), u^k(t))$ to obtain $x^k(\cdot)$

2. linearize nonlinear system around $(x^k(\cdot), u^k(\cdot))$

3. solve time-varying LQR with cost

$$J = \int_0^T (x^k(t) + \bar{x}(t))^T Q (x^k(t) + \bar{x}(t)) + (u^k(t) + \bar{u}(t))^T R (u^k(t) + \bar{u}(t))\, dt + (x^k(T) + \bar{x}(T))^T Q_T (x^k(T) + \bar{x}(T)),$$

and dynamics $\dot{\bar{x}}(t) = A(t)\bar{x}(t) + B(t)\bar{u}(t)$, where

$$A(t) = \left.\frac{df}{dx}\right|_{x^k(t), u^k(t)}, \quad B(t) = \left.\frac{df}{du}\right|_{x^k(t), u^k(t)}$$

4. update $u^{k+1}(t) := u^k(t) + \bar{u}(t)$

Example: simple car

- nonlinear dynamics (LaValle, Ch 13), set $L = 1$:

$$\dot{x} = u_s \cos(\theta)$$
$$\dot{y} = u_s \sin(\theta)$$
$$\dot{\theta} = \frac{u_s}{L} \tan(u_\phi)$$

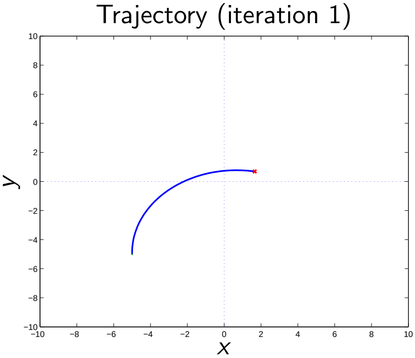
- linearization around a trajectory:

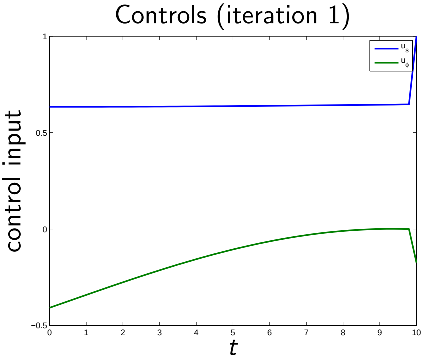
$$\begin{bmatrix} \dot{\bar{x}} \\ \dot{\bar{y}} \\ \dot{\bar{\theta}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & -u_s(t) \sin\theta(t) \\ 0 & 0 & u_s(t)\cos\theta(t) \\ 0 & 0 & 0 \end{bmatrix}}_{A(t)} \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ \frac{1}{L}\tan u_\phi(t) & \frac{u_s(t)}{L\cos^2 u_\phi(t)} \end{bmatrix}}_{B(t)} \begin{bmatrix} \bar{u}_s \\ \bar{u}_\phi \end{bmatrix}$$

- penalties

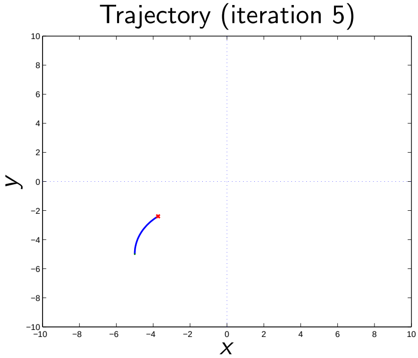
$$Q = 0, \quad Q_T = \mathbf{diag}(1, 1, 0), \quad R = \mathbf{diag}(0.1, 1)$$

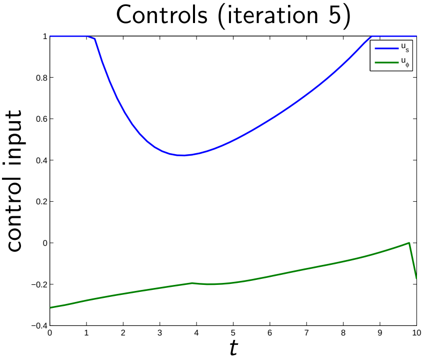
Example: simple car



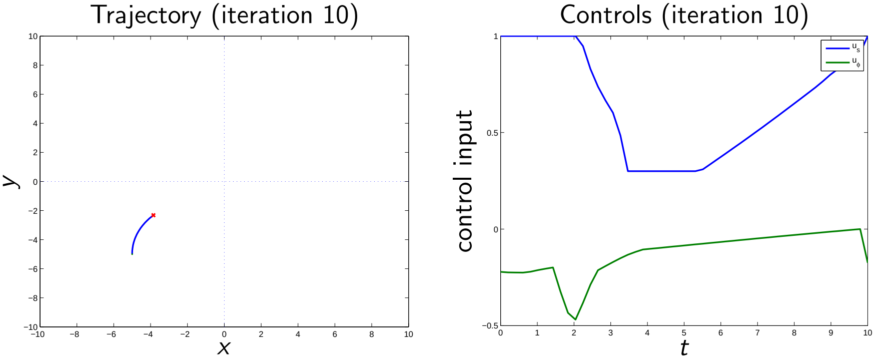


Example: simple car

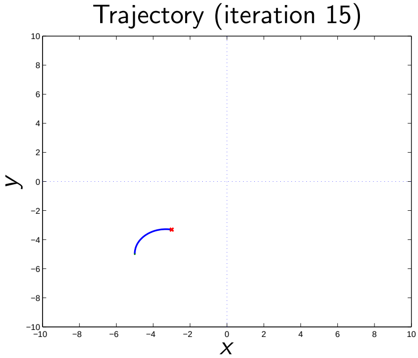


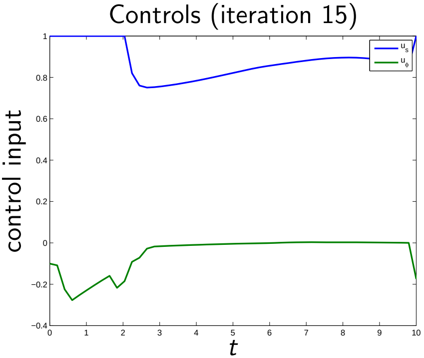


Example: simple car

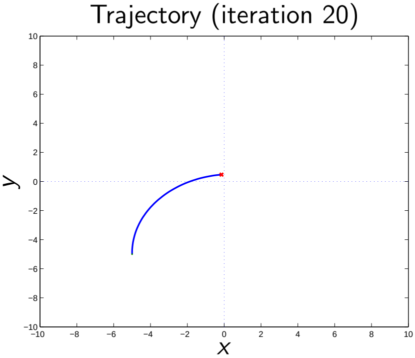


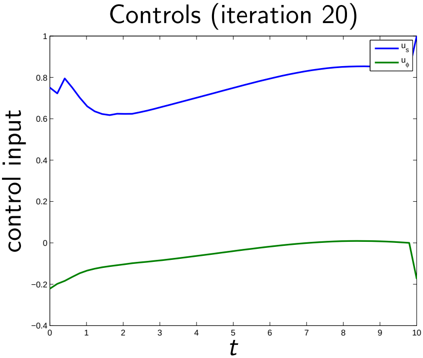
Example: simple car



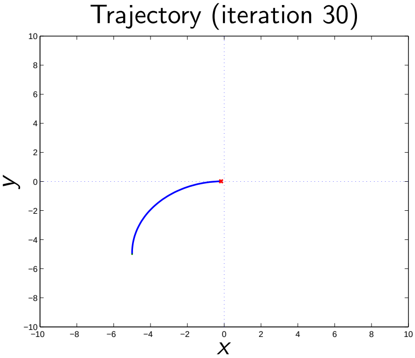


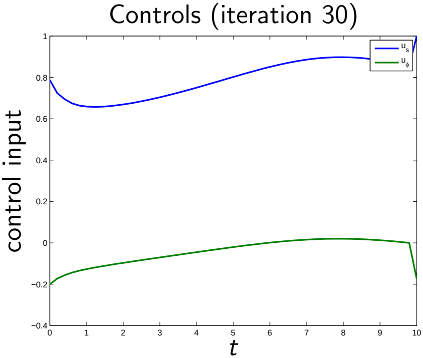
Example: simple car





Example: simple car





Summary

The linear quadratic regulator is a workhorse technique

- guaranteed stability margins (ex. 13)
- building block for LQE, LQG, stochastic control
- intimately connected with $\mathbf{H}_2$ optimal control
- online stabilization
- iterative LQR: nonlinear trajectory generation

However, it is not the entire story

- picking Q and R matrices is an art
- LQR does not handle constraints on state or input
- generically works only for linear systems