Linear Optimal Control

How does this guy remain upright?

Overview

1. expressing a linear system in state space form

2. discrete time linear optimal control (LQR)

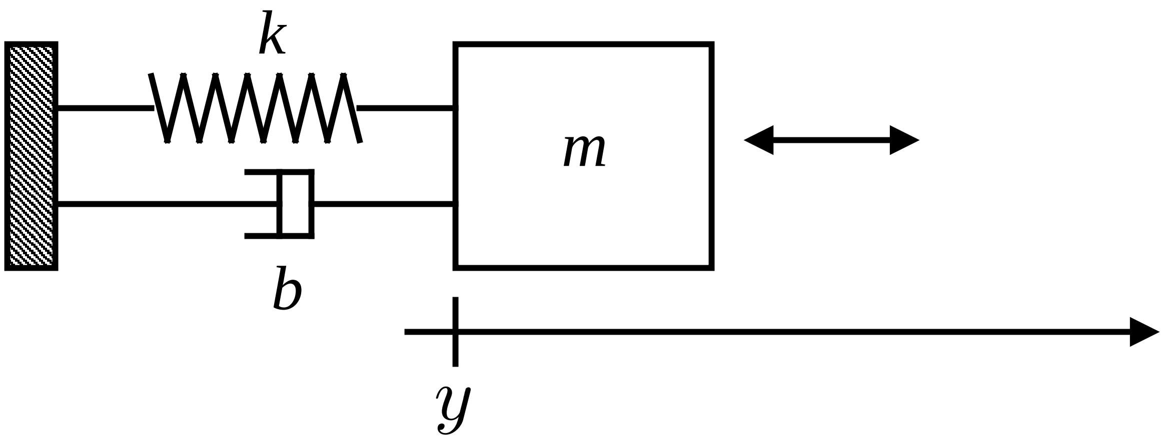
3. linearizing around an operating point

4. linear model predictive control

5. LQR variants

6. model predictive control for non-linear systems

A simple system

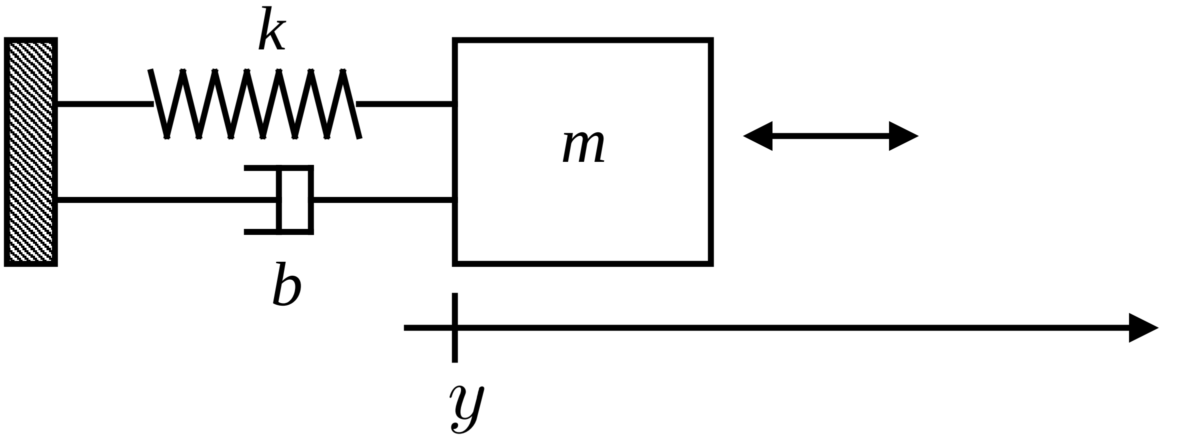


Force exerted by the spring: $f = ky$

Force exerted by the damper: $f = b\dot{y}$

Force exerted by the inertia of the mass: $f = m\ddot{y}$

A simple system



Consider the motion of the mass

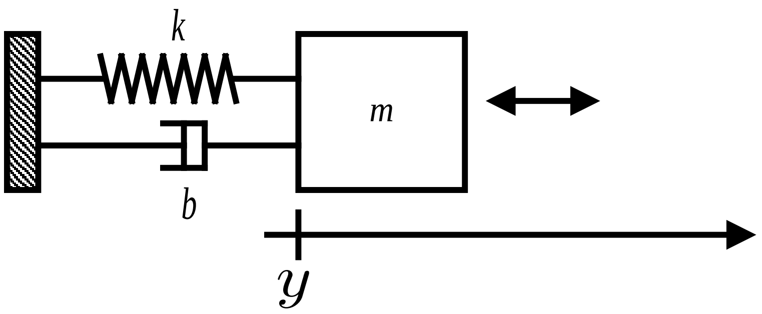
- there are no other forces acting on the mass
- therefore, the equation of motion is the sum of the forces:

$$0 = m\ddot{y} + b\dot{y} + ky$$

This is called a linear system. Why?

A simple system

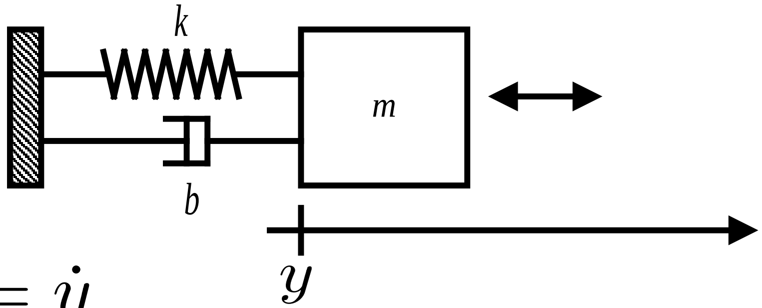
Let's express this in "state space form":



$$0 = m\ddot{y} + b\dot{y} + ky$$

A simple system

Let's express this in "state space form":

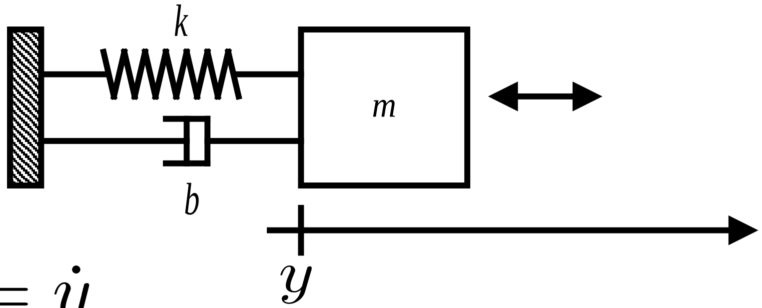


$$0 = m\ddot{y} + b\dot{y} + ky$$

$$\begin{cases} \dot{y} = \dot{y} \\ \ddot{y} = -\dfrac{1}{m}(b\dot{y} + ky) \end{cases}$$

A simple system

Let's express this in "state space form":



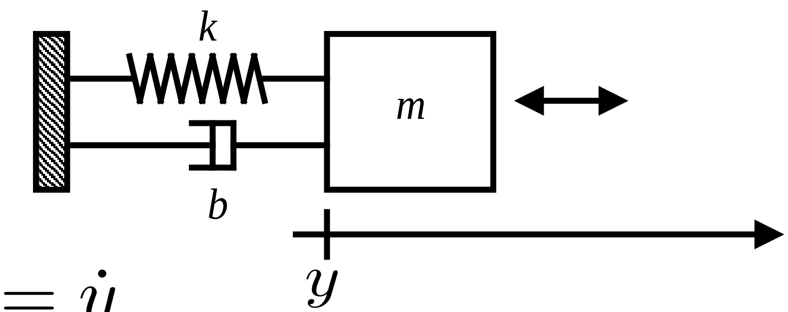
$$0 = m\ddot{y} + b\dot{y} + ky$$

$$\begin{cases} \dot{y} = \dot{y} \\ \ddot{y} = -\dfrac{1}{m}(b\dot{y} + ky) \end{cases}$$

$$\begin{pmatrix} \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -k/m & -b/m \end{pmatrix} \begin{pmatrix} y \\ \dot{y} \end{pmatrix}$$

A simple system

Let's express this in "state space form":



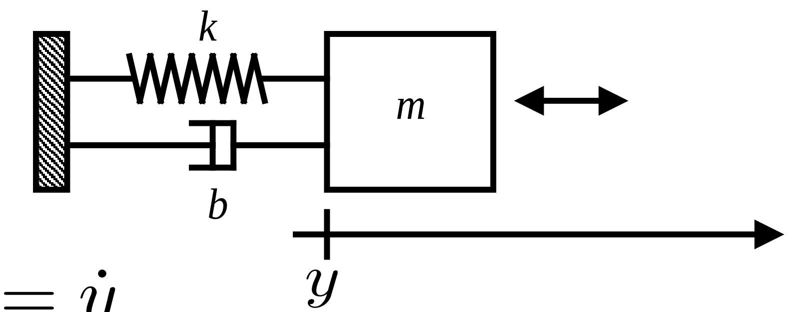
$$0 = m\ddot{y} + b\dot{y} + ky$$

$$\begin{cases} \dot{y} = \dot{y} \\ \ddot{y} = -\dfrac{1}{m}(b\dot{y} + ky) \end{cases}$$

$$\begin{pmatrix} \dot{y} \\ \ddot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -k/m & -b/m \end{pmatrix}}_{A} \begin{pmatrix} y \\ \dot{y} \end{pmatrix}$$

A simple system

Let's express this in "state space form":

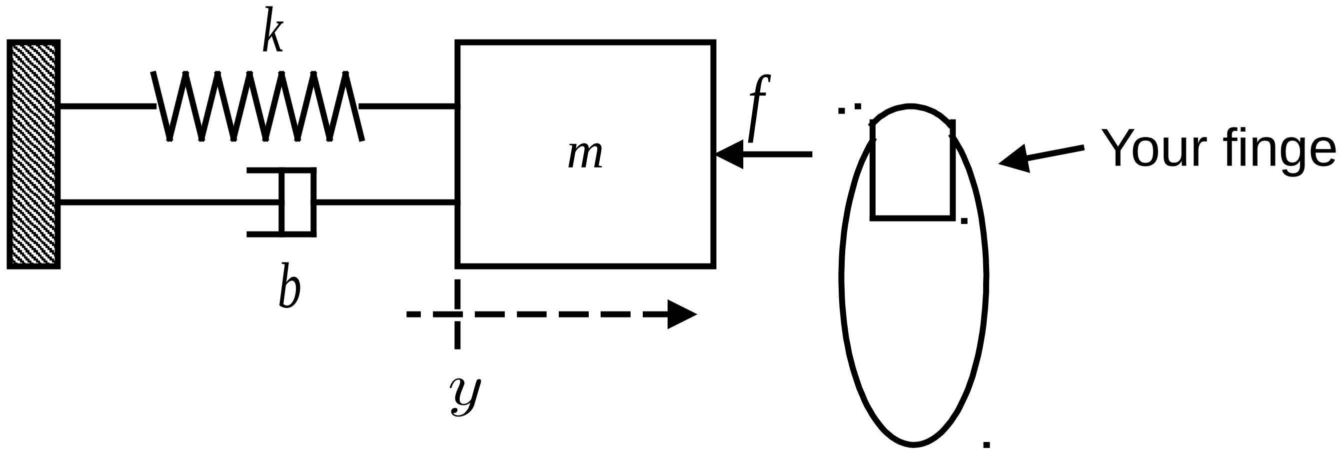


$$0 = m\ddot{y} + b\dot{y} + ky \quad \Rightarrow \quad \begin{cases} \dot{y} = \dot{y} \\ \ddot{y} = -\frac{1}{m}(b\dot{y} + ky) \end{cases}$$

$$\begin{pmatrix} \dot{y} \\ \ddot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -k/m & -b/m \end{pmatrix}}_{A} \begin{pmatrix} y \\ \dot{y} \end{pmatrix}$$

$\dot{x} = Ax$ where $x = \begin{pmatrix} y \\ \dot{y} \end{pmatrix}$

A simple system



Suppose that you apply a force:

$$u = m\ddot{y} + b\dot{y} + ky \quad \Rightarrow \quad \begin{cases} \dot{y} = \dot{y} \\ \ddot{y} = u - \frac{1}{m}(b\dot{y} + ky) \end{cases}$$

$$\begin{pmatrix} \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -k/m & -b/m \end{pmatrix} \begin{pmatrix} y \\ \dot{y} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

A simple system

Suppose that you apply a force:

$$u = m\ddot{y} + b\dot{y} + ky \quad \Rightarrow \quad \begin{cases} \dot{y} = \dot{y} \\ \ddot{y} = u - \dfrac{1}{m}(b\dot{y} + ky) \end{cases}$$

$$\begin{pmatrix} \dot{y} \\ \ddot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -k/m & -b/m \end{pmatrix}}_{A} \begin{pmatrix} y \\ \dot{y} \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{B} u$$

$$\Downarrow$$

$$\dot{x} = Ax + Bu$$

A simple system

Suppose that you apply a force:

$$u = m\ddot{y} + b\dot{y} + ky \quad \Rightarrow \quad \begin{cases} \dot{y} = \dot{y} \\ \ddot{y} = u - \dfrac{1}{m}(b\dot{y} + ky) \end{cases}$$

$$\begin{pmatrix} \dot{y} \\ \ddot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -k/m & -b/m \end{pmatrix}}_{A} \begin{pmatrix} y \\ \dot{y} \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{B} u$$

Canonical form for a linear system

$$\dot{x} = Ax + Bu$$

Continuous time vs discrete time

$$\dot{x} = Ax + Bu$$

Continuous time

$$x_{t+1} = Ax_t + Bu_t$$

Discrete time

Continuous time vs discrete time

$$\dot{x} = Ax + Bu$$

Continuous time

$$x_{t+1} = Ax_t + Bu_t$$

Discrete time

What are *A* and *B* now?

Continuous time vs discrete time

$$\dot{x} = Ax + Bu$$

Continuous time

$$x_{t+1} = Ax_t + Bu_t$$

Discrete time

$$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = A \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + Bu$$

What are *A* and *B* now?

Simple system in discrete time

We want something in this form: $$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = A \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + Bu$$

Simple system in discrete time

We want something in this form: $$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = A \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + Bu$$

$$y_{t+1} = y_t + \dot{y}_t dt$$

Simple system in discrete time

We want something in this form: $$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = A \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + Bu$$

$$y_{t+1} = y_t + \dot{y}_t dt$$

$$\dot{y}_{t+1} = \dot{y}_t + \ddot{y}_t dt$$

Simple system in discrete time

We want something in this form: $$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = A \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + Bu$$

$$y_{t+1} = y_t + \dot{y}_t dt$$

$$\dot{y}_{t+1} = \dot{y}_t + \ddot{y}_t dt$$

$$\dot{y}_{t+1} = \dot{y}_t - \frac{1}{m}(b\dot{y} + ky)dt + udt$$

Simple system in discrete time

We want something in this form: $$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = A \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + Bu$$

$$y_{t+1} = y_t + \dot{y}_t dt$$

$$\dot{y}_{t+1} = \dot{y}_t + \ddot{y}_t dt$$

$$\dot{y}_{t+1} = \dot{y}_t - \frac{1}{m}(b\dot{y} + ky)dt + udt$$

$$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & dt \\ -\frac{k}{m}dt & 1 - \frac{b}{m}dt \end{pmatrix} \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + \begin{pmatrix} 0 \\ dt \end{pmatrix} u$$

Simple system in discrete time

We want something in this form: $$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = A \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + Bu$$

$$y_{t+1} = y_t + \dot{y}_t dt$$

$$\dot{y}_{t+1} = \dot{y}_t + \ddot{y}_t dt$$

$$\dot{y}_{t+1} = \dot{y}_t - \frac{1}{m}(b\dot{y} + ky)dt + udt$$

$$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & dt \\ -\frac{k}{m}dt & 1 - \frac{b}{m}dt \end{pmatrix}}_{A} \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ dt \end{pmatrix}}_{B} u$$

Continuous time vs discrete time

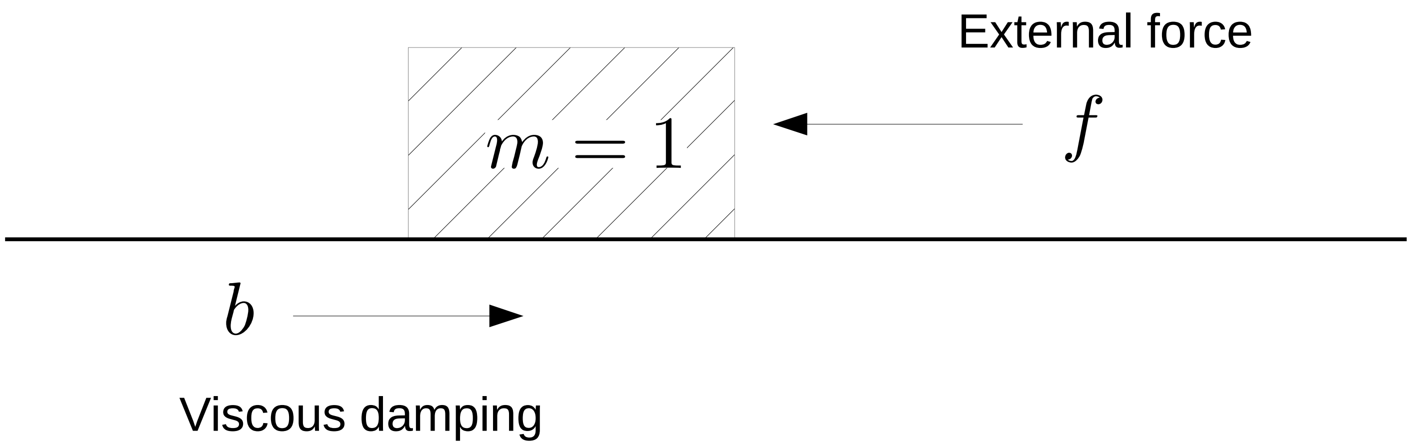
$$\dot{x} = Ax + Bu$$ ← CT

$$x_{t+1} = Ax_t + Bu_t$$ ← DT

$$\begin{pmatrix} \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -k/m & -b/m \end{pmatrix} \begin{pmatrix} y \\ \dot{y} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$ CT

$$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & dt \\ -\frac{k}{m}dt & 1 - \frac{b}{m}dt \end{pmatrix} \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + \begin{pmatrix} 0 \\ dt \end{pmatrix} u$$ DT

Exercise: write DT system dynamics



Exercise: write DT system dynamics

Something else...

Overview

1. expressing a linear system in state space form
2. discrete time linear optimal control (LQR)
3. linearizing around an operating point
4. linear model predictive control
5. LQR variants
6. model predictive control for non-linear systems

The linear control problem

<u>Given:</u>

System: $x_{t+1} = Ax_t + Bu_t$

The linear control problem

<u>Given:</u>

System: $$x_{t+1} = Ax_t + Bu_t$$

Cost function: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

where: $$X = (x_1, \ldots, x_T)$$

$$U = (u_1, \ldots, u_{T-1})$$

The linear control problem

<u>Given:</u>

$$Q = Q^T \geq 0$$

$$Q_T = Q_T^T \geq 0$$

System: $x_{t+1} = Ax_t + Bu_t$

Cost function: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

where: $$X = (x_1, \ldots, x_T)$$

$$U = (u_1, \ldots, u_{T-1})$$

The linear control problem

<u>Given:</u>

$$Q = Q^T \geq 0$$

$$Q_T = Q_T^T \geq 0$$

System: $x_{t+1} = Ax_t + Bu_t$

Cost function: $J(X, U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$

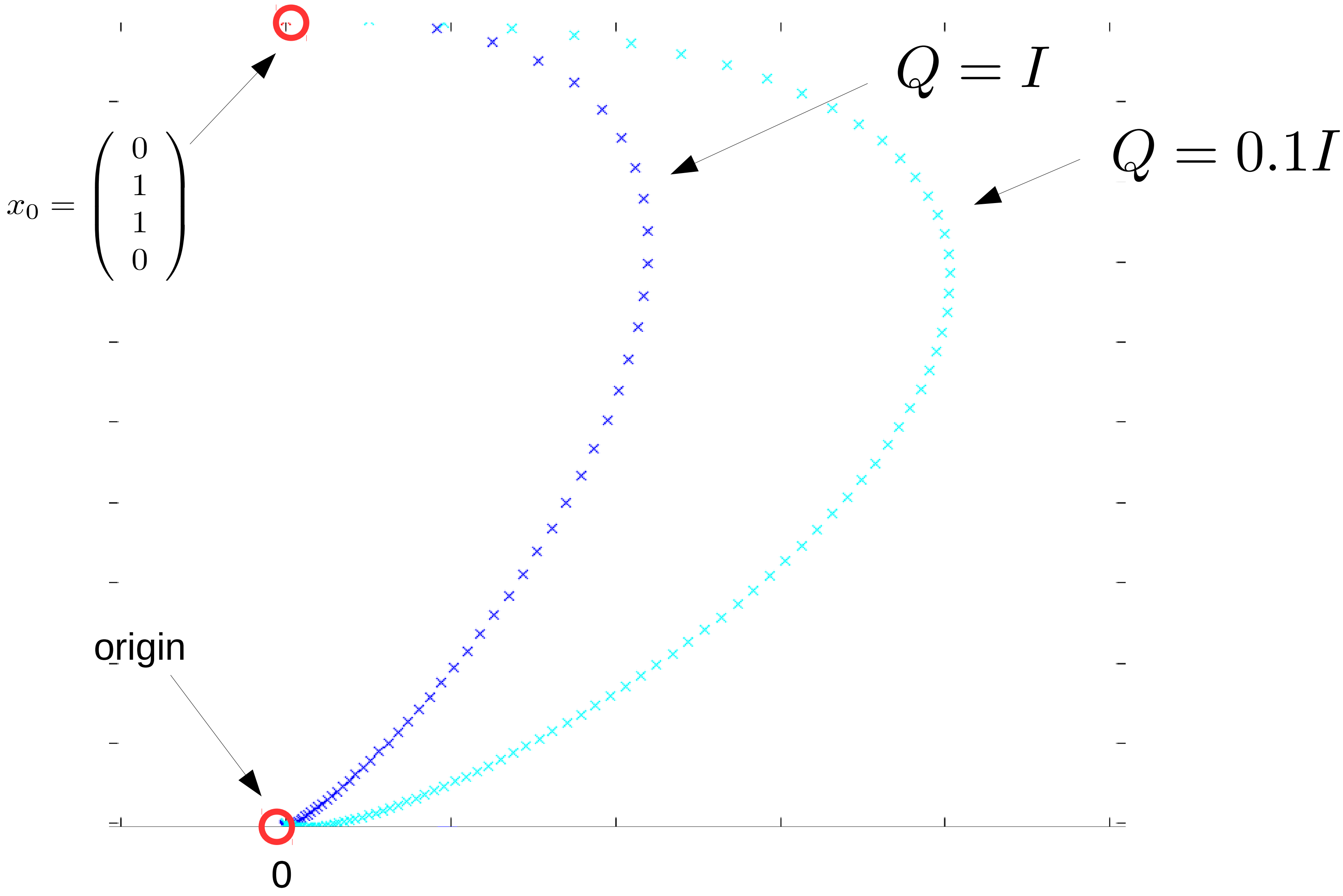
where: $X = (x_1, \ldots, x_T)$

$U = (u_1, \ldots, u_{T-1})$

Initial state: x_1

<u>Calculate:</u> *U* that minimizes J(*X*,*U*)

Examples of solutions for double integrator



The linear control problem

$$Q = Q^T \geq 0$$

$$Q_T = Q_T^T \geq 0$$

<u>Given:</u>

System:

Cost functi

$$x_t^T Q x_t + u_t^T R u_t$$

$$x_T)$$

$$u_{T-1})$$

Important problem!

How do we solve it?

Initial state: x_1

<u>Calculate:</u> *U* that minimizes J(*X*,*U*)

One solution: least squares

$$x_1 = x_1$$

$$x_2 = Ax_1 + Bu_1$$

$$x_3 = A(Ax_1 + Bu_1) + Bu_2 = A^2x_1 + ABu_1 + Bu_2$$

$$x_4 = \ldots$$

One solution: least squares

$$x_1 = x_1$$

$$x_2 = Ax_1 + Bu_1$$

$$x_3 = A(Ax_1 + Bu_1) + Bu_2 = A^2 x_1 + ABu_1 + Bu_2$$

$$x_4 = \ldots$$

$$\begin{pmatrix} x_1 \\ \vdots \\ x_T \end{pmatrix} = \begin{pmatrix} 0 & \ldots & & & \\ B & 0 & \ldots & & \\ AB & B & 0 & \ldots & \\ A^2B & AB & B & 0 & \ldots \\ \ldots & & & & \\ A^{T-1}B & A^{T-2}B & \ldots & \ldots & B \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_{T-1} \end{pmatrix} + \begin{pmatrix} I \\ A \\ A^2 \\ \vdots \\ A^T \end{pmatrix} x_1$$

One solution: least squares

$$\begin{pmatrix} x_1 \\ \vdots \\ x_T \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \dots & & & \\ B & 0 & \dots & & \\ AB & B & 0 & \dots & \\ A^2B & AB & B & 0 & \dots \\ \dots & & & & \\ A^{T-1}B & A^{T-2}B & \dots & \dots & B \end{pmatrix}}_{G} \begin{pmatrix} u_1 \\ \vdots \\ u_{T-1} \end{pmatrix} + \underbrace{\begin{pmatrix} I \\ A \\ A^2 \\ \vdots \\ A^T \end{pmatrix}}_{H} x_1$$

$$X = GU + Hx_1$$

where

$$X = (x_1, \dots, x_T)$$

$$U = (u_1, \dots, u_{T-1})$$

One solution: least squares

$$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

$$J(X,U) = X^T \mathbb{Q} X + U^T \mathbb{R} U$$

where:

$$X = (x_1, \ldots, x_T)$$

$$U = (u_1, \ldots, u_{T-1})$$

$$\mathbb{Q} = \begin{pmatrix} Q & 0 & \ldots & & \\ 0 & Q & 0 & \ldots & \\ \vdots & & & & \\ 0 & 0 & 0 & \ldots & Q_T \end{pmatrix}$$

$$\mathbb{R} = \begin{pmatrix} R & 0 & \ldots & & \\ 0 & R & 0 & \ldots & \\ \vdots & & & & \\ 0 & 0 & 0 & \ldots & R \end{pmatrix}$$

One solution: least squares

Given:

System: $$x_{t+1} = Ax_t + Bu_t$$

Cost function: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

where: $$X = (x_1, \dots, x_T)$$
$$U = (u_1, \dots, u_{T-1})$$

Initial state: x_1

Calculate: *U* that minimizes J(*X*,*U*)

One solution: least squares

Given:

System: $$X = GU + Hx_1$$

Cost function: $$J(X,U) = X^T \mathbb{Q} X + U^T \mathbb{R} U$$

Initial state: x_1

Calculate: *U* that minimizes J(*X*,*U*)

One solution: least squares

Substitute *X* into *J*:

$$J(X,U) = (GU + Hx_1)^T \mathbb{Q}(GU + Hx_1) + U^T \mathbb{R} U$$

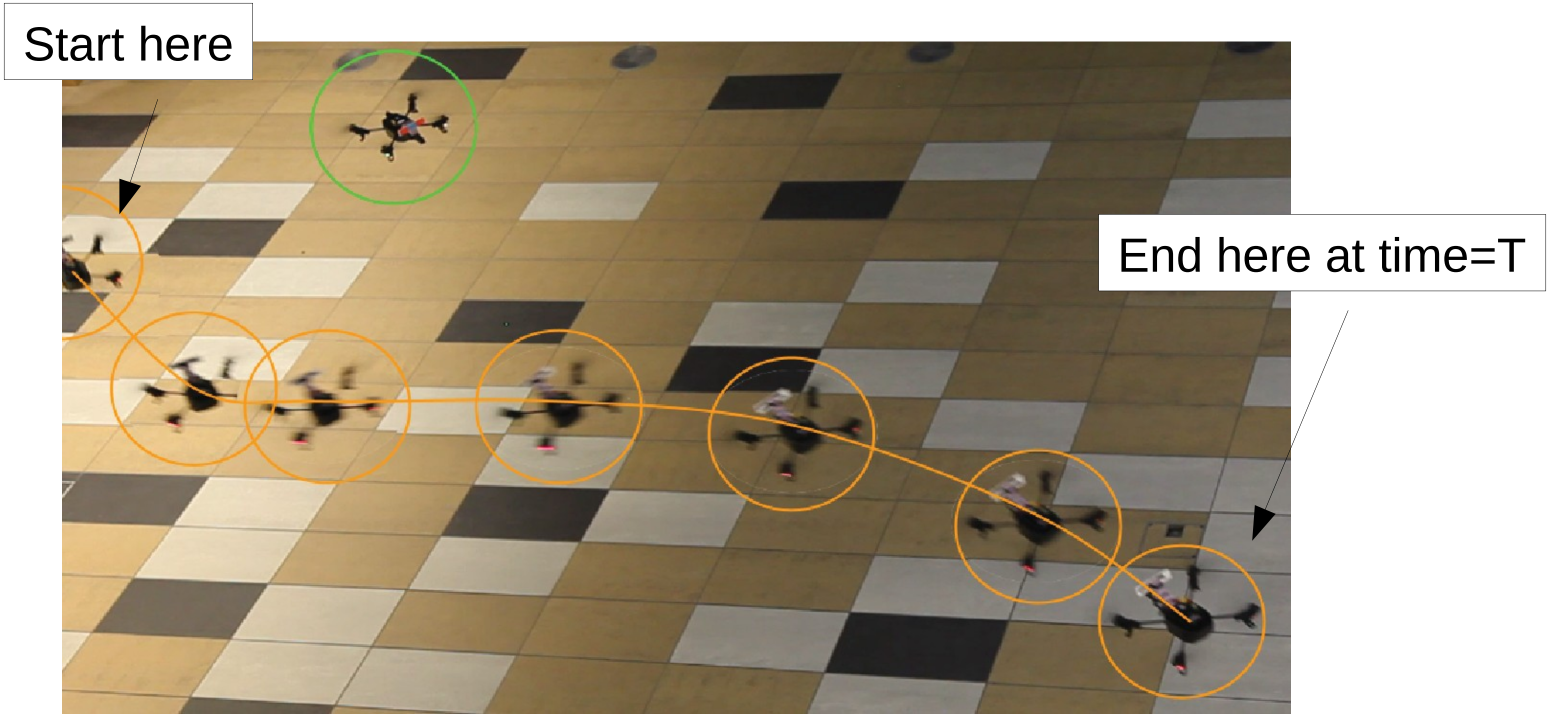
$$J(X,U) = U^T (G^T \mathbb{Q} G) U + U^T \mathbb{R} U + 2H^T x_1^T \mathbb{Q} G U$$

Minimize by setting d*J*/*dU*=0:

$$\frac{\partial J(X,U)}{\partial U} = 2(G^T \mathbb{Q} G)U + 2\mathbb{R}U + 2H^T x_1^T \mathbb{Q} G = 0$$

Solve for *U*: $$U = -(G^T \mathbb{Q} G + \mathbb{R})^{-1} G^T \mathbb{Q} H x_1$$

What can this do?



Solve for optimal trajectory: $U = -(G^T \mathbb{Q} G + \mathbb{R})^{-1} G^T \mathbb{Q} H x_1$

Image: van den Berg, 2015

What can this do?

$$U = -(G^T \mathbb{Q} G + \mathbb{R})^{-1} G^T \mathbb{Q} H x_1$$

This is cool, but...

- only works for finite horizon problems
- doesn't account for noise
- requires you to invert a big matrix

Let's try to solve this another way

Bellman optimality principle:

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

Why is this equation true?

Let's try to solve this another way

Bellman optimality principle:

Cost-to-go from state x at time t

Cost-to-go from state $(Ax+Bu)$ at time $t+1$

$$V_t(x) = \min_u \left[\underbrace{x^T Q x + u^T R u}_{\text{Cost incurred on this time step}} + \underbrace{V_{t+1}(Ax + Bu)}_{\text{Cost incurred after this time step}} \right]$$

Cost incurred **on** this time step

Cost incurred **after** this time step

Let's try to solve this another way

For the sake of argument, suppose that the cost-to-go is always a quadratic function like this: →

$$V_t(x) = x^T P_t x$$

where: $P_t = P_t^T \geq 0$

Let's try to solve this another way

For the sake of argument, suppose that the cost-to-go is always a quadratic function like this: → $V_t(x) = x^T P_t x$

where: $P_t = P_t^T \geq 0$

Then:

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

$$= x^T Q x + \min_u \left[u^T R u + (Ax + Bu)^T P_{t+1} (Ax + Bu) \right]$$

Let's try to solve this another way

For the sake of argument, suppose that the cost-to-go is always a quadratic function like this: → $V_t(x) = x^T P_t x$

where: $P_t = P_t^T \geq 0$

Then:

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

$$= x^T Q x + \underbrace{\min_u \left[u^T R u + (Ax + Bu)^T P_{t+1} (Ax + Bu) \right]}$$

How do we minimize this term?
– take derivative and set it to zero.

Let's try to solve this another way

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

$$= x^T Q x + \underbrace{\min_u \left[u^T R u + (Ax + Bu)^T P_{t+1} (Ax + Bu) \right]}$$

How do we minimize this term?
– take derivative and set it to zero.

$$\frac{\partial V_t(x)}{\partial u} = \left[u^T R + u^T B^T P_{t+1} B + x^T A^T P_{t+1} B \right] = 0$$

$$u^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x$$

optimal control as a function of state
– but: it depends on P_{t+1}...

Let's try to solve this another way

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu)\right]$$

$$= x^T Q x + \underbrace{\min_u \left[u^T R u + (Ax + Bu)^T P_{t+1}(Ax + Bu)\right]}$$

How do we minimize this term?
– take derivative and set it to zero.

$$\frac{\partial V_t(x)}{\partial u} = \left[u^T R + u^T B^T P_{t+1} B + x^T A^T P_{t+1} B\right] = 0$$

How solve for P_{t+1}???

$$u^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x$$

optimal control as a function of state
– but: it depends on P_{t+1}...

Let's try to solve this another way

Substitute *u* into *V_t(x)*:

$$u^* = \underbrace{-(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x}$$

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

Let's try to solve this another way

Substitute *u* into *V_t(x)*:

$$u^* = \underbrace{-(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x}$$

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

$$V_t(x) = x^T \left[Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A \right] x$$

Let's try to solve this another way

Substitute *u* into *V_t(x)*:

$$u^* = \underbrace{-(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x}$$

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

$$V_t(x) = x^T \underbrace{\left[Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A \right]}_{P_t} x$$

Let's try to solve this another way

Substitute *u* into *V_t(x)*:

$$u^* = \underbrace{-(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x}$$

$$V_t(x) = \min_u \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

$$V_t(x) = x^T \underbrace{\left[Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A \right]}_{P_t} x$$

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

Let's try to solve this another way

Substitute *u* into *V_t(x)*:

$$u^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x$$

$$V_t(x) = \min \left[x^T Q x + u^T R u + V_{t+1}(Ax + Bu) \right]$$

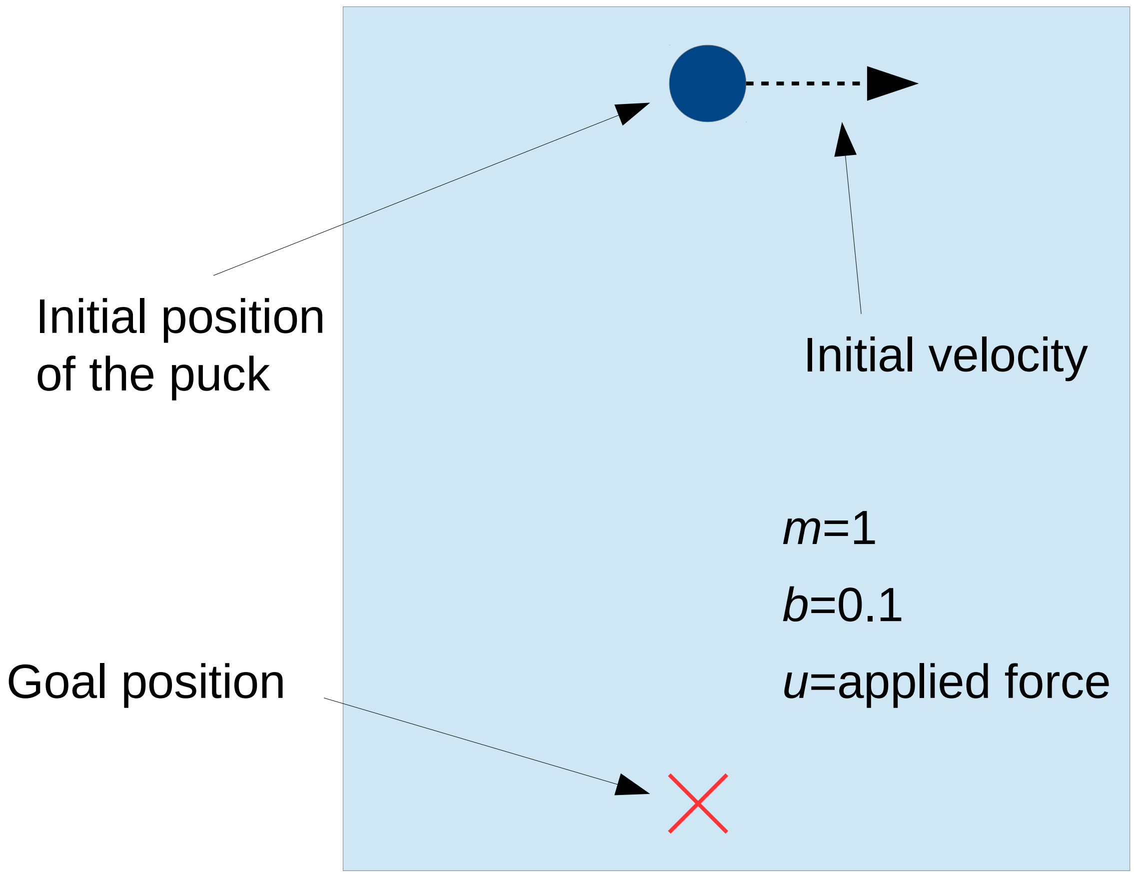
$$V_t(x) = x^T \left[Q \ldots P_{t+1} A \right] x$$

Dynamic Riccati Equation

P_t

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

Example: planar double integrator



Air hockey table

<u>Build the LQR controller for:</u>

Initial state: $x_0 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$

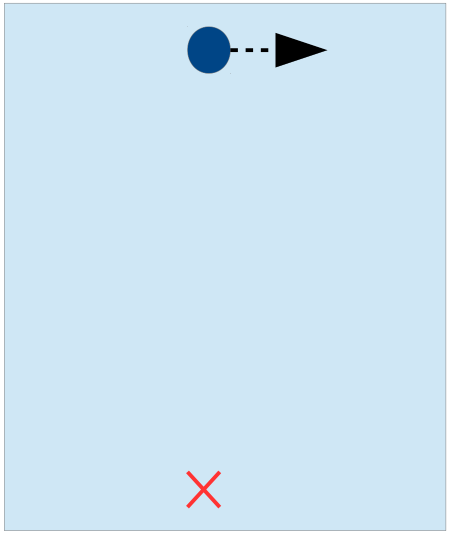
Time horizon: $T = 100$

Cost fn: $Q_T = 1000I$

$Q = I$

$R = I$

Example: planar double integrator

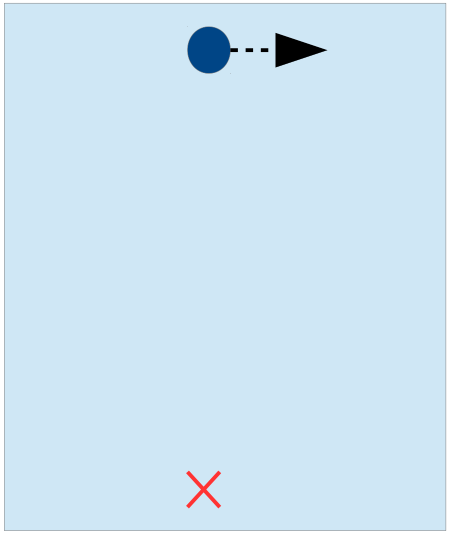


Air hockey table

Step 1:
Calculate P backward from T: P_100, P_99, P_98, … , P_1

HOW?

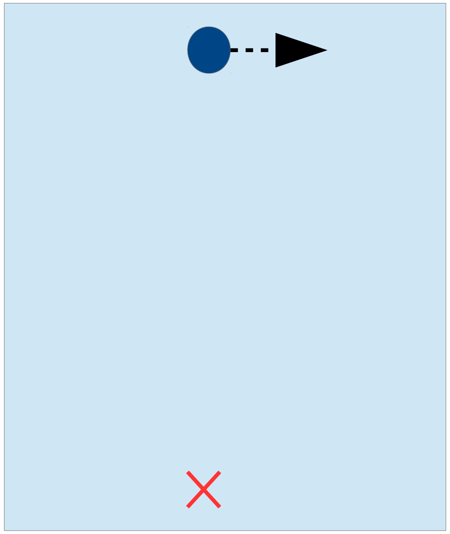
Example: planar double integrator


Air hockey table

Step 1:
Calculate P backward from T: P_100, P_99, P_98, … , P_1

$$P_{100} = 1000I$$

Example: planar double integrator



Air hockey table

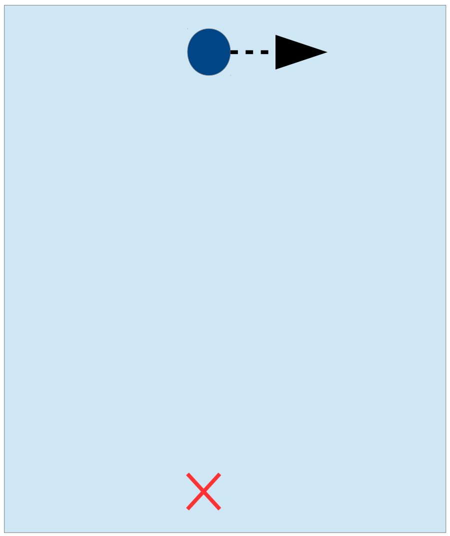
<u>Step 1:</u>

Calculate P backward from T: P_100, P_99, P_98, … , P_1

$$P_{100} = 1000I$$

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

Example: planar double integrator



Air hockey table

<u>Step 1:</u>

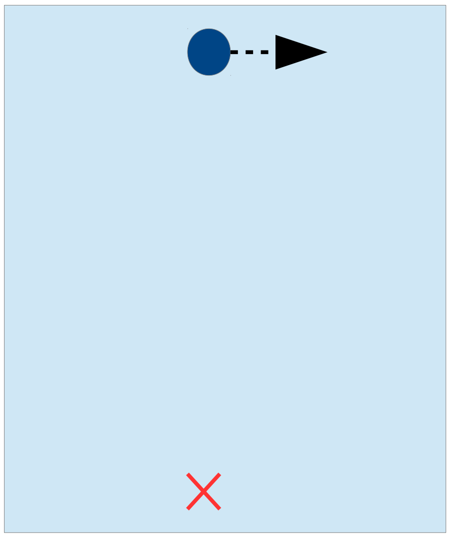
Calculate P backward from T: P_100, P_99, P_98, … , P_1

$$P_{100} = 1000I$$

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

$$P_{99} = \begin{pmatrix} 1001 & 0 & 1000 & 0 \\ 0 & 1001 & 0 & 1000 \\ 1000 & 0 & 1001 & 0 \\ 0 & 1000 & 0 & 1001 \end{pmatrix}$$

Example: planar double integrator


Air hockey table

<u>Step 1:</u>
Calculate P backward from T: P_100, P_99, P_98, … , P_1

$$P_{100} = 1000I$$

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

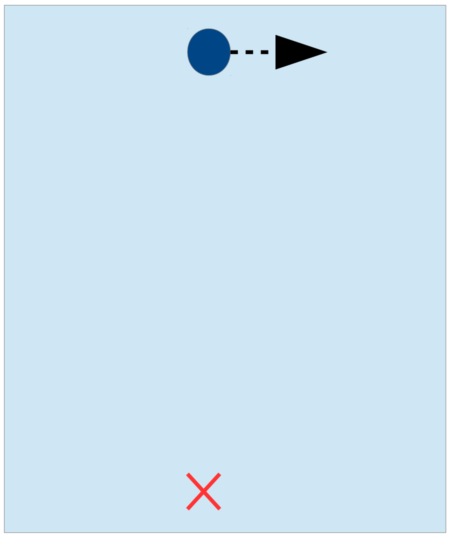
$$P_{99} = \begin{pmatrix} 1001 & 0 & 1000 & 0 \\ 0 & 1001 & 0 & 1000 \\ 1000 & 0 & 1001 & 0 \\ 0 & 1000 & 0 & 1001 \end{pmatrix}$$

$$\vdots$$

$$P_{90} = \begin{pmatrix} 119.7293 & 0 & 59.9348 & 0 \\ 0 & 119.7293 & 0 & 59.9348 \\ 59.9348 & 0 & 43.1627 & 0 \\ 0 & 59.9348 & 0 & 43.1627 \end{pmatrix}$$

$$\vdots$$

Example: planar double integrator



Air hockey table

Step 2:

Calculate u starting at t=1 and going forward to t=T-1

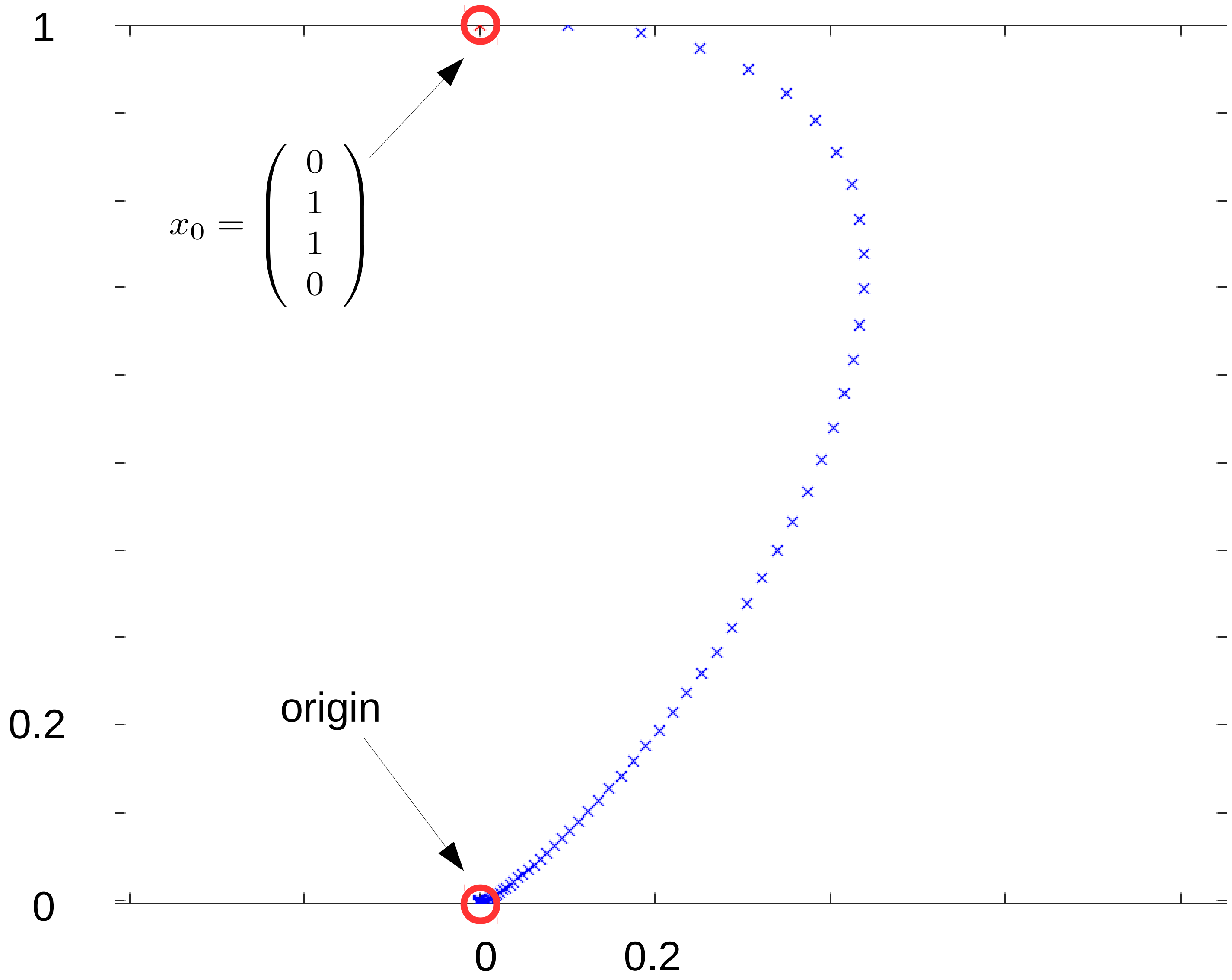
$$u_1 = -(R + B^T P_1 B)^{-1} B^T P_1 A x$$

$$\vdots$$

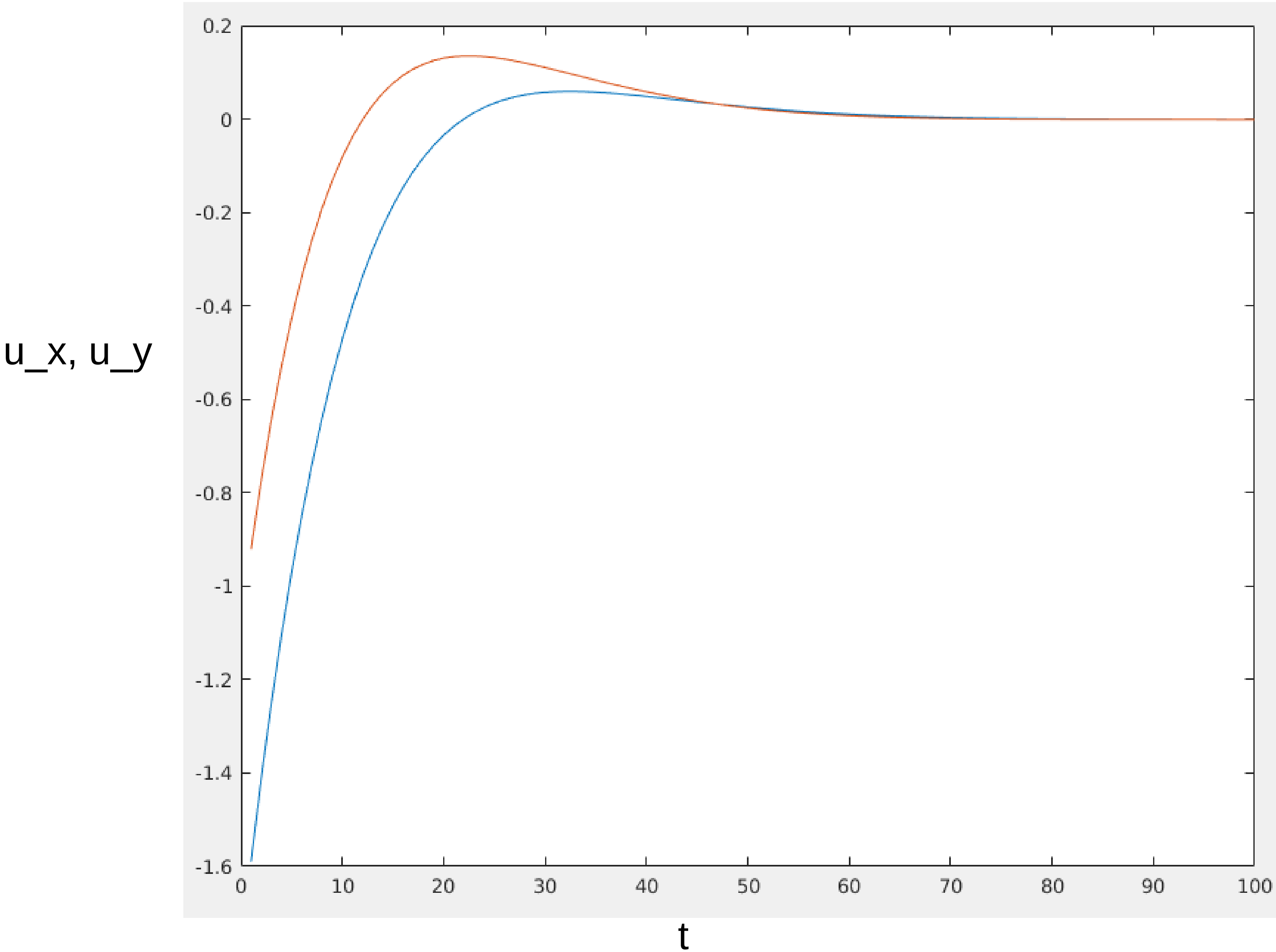
$$u_2 = -(R + B^T P_2 B)^{-1} B^T P_2 A x$$

$$\vdots$$

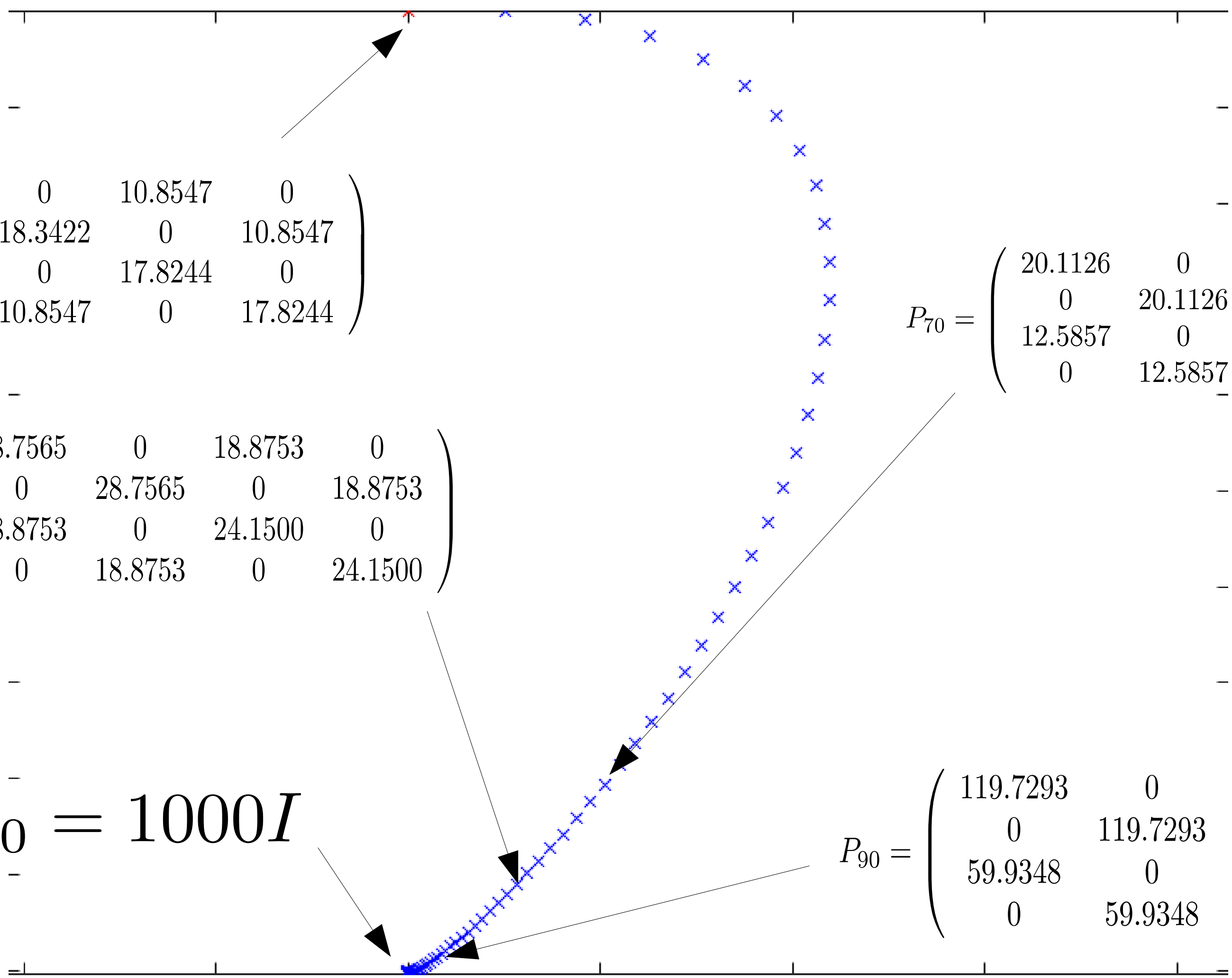
Example: planar double integrator



Example: planar double integrator



Example: planar double integrator



$$P_1 = \begin{pmatrix} 18.3422 & 0 & 10.8547 & 0 \\ 0 & 18.3422 & 0 & 10.8547 \\ 10.8547 & 0 & 17.8244 & 0 \\ 0 & 10.8547 & 0 & 17.8244 \end{pmatrix}$$

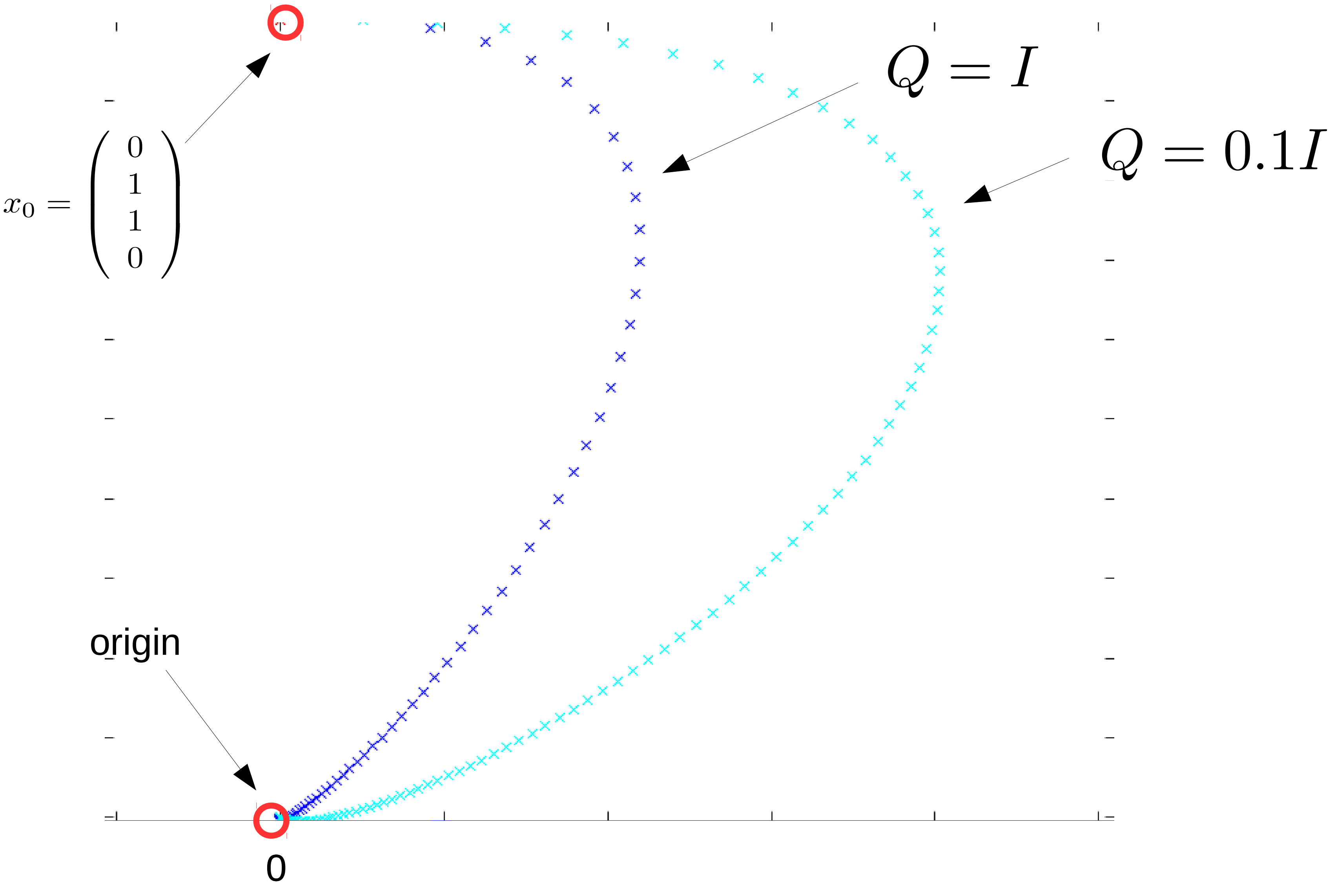
$$P_{70} = \begin{pmatrix} 20.1126 & 0 & 12.5857 & 0 \\ 0 & 20.1126 & 0 & 12.5857 \\ 12.5857 & 0 & 19.5454 & 0 \\ 0 & 12.5857 & 0 & 19.5454 \end{pmatrix}$$

$$P_{80} = \begin{pmatrix} 28.7565 & 0 & 18.8753 & 0 \\ 0 & 28.7565 & 0 & 18.8753 \\ 18.8753 & 0 & 24.1500 & 0 \\ 0 & 18.8753 & 0 & 24.1500 \end{pmatrix}$$

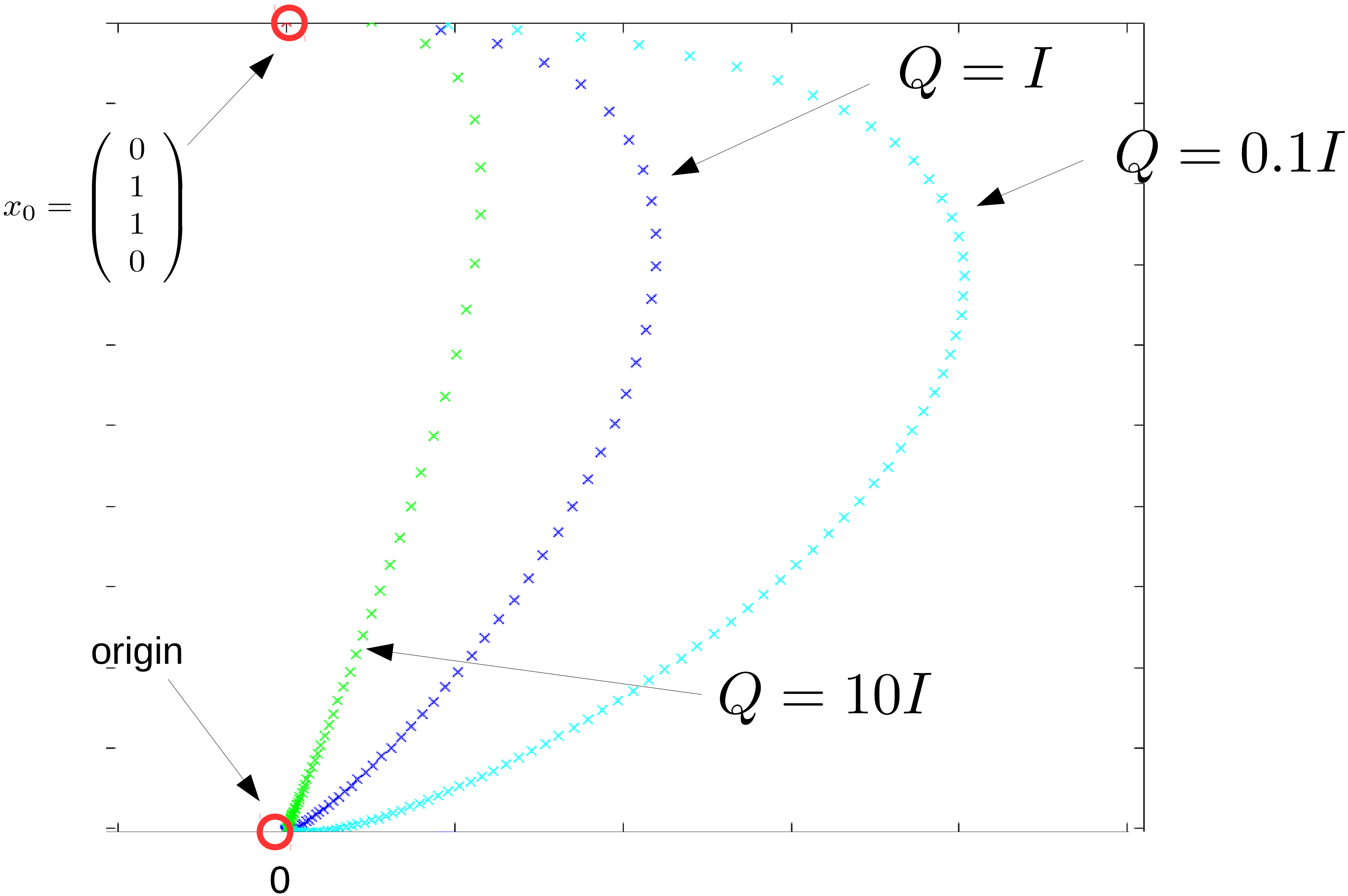
$$P_{100} = 1000I$$

$$P_{90} = \begin{pmatrix} 119.7293 & 0 & 59.9348 & 0 \\ 0 & 119.7293 & 0 & 59.9348 \\ 59.9348 & 0 & 43.1627 & 0 \\ 0 & 59.9348 & 0 & 43.1627 \end{pmatrix}$$

Example: planar double integrator



Example: planar double integrator



The infinite horizon case

So far: we have optimized cost over a fixed horizon, *T*.

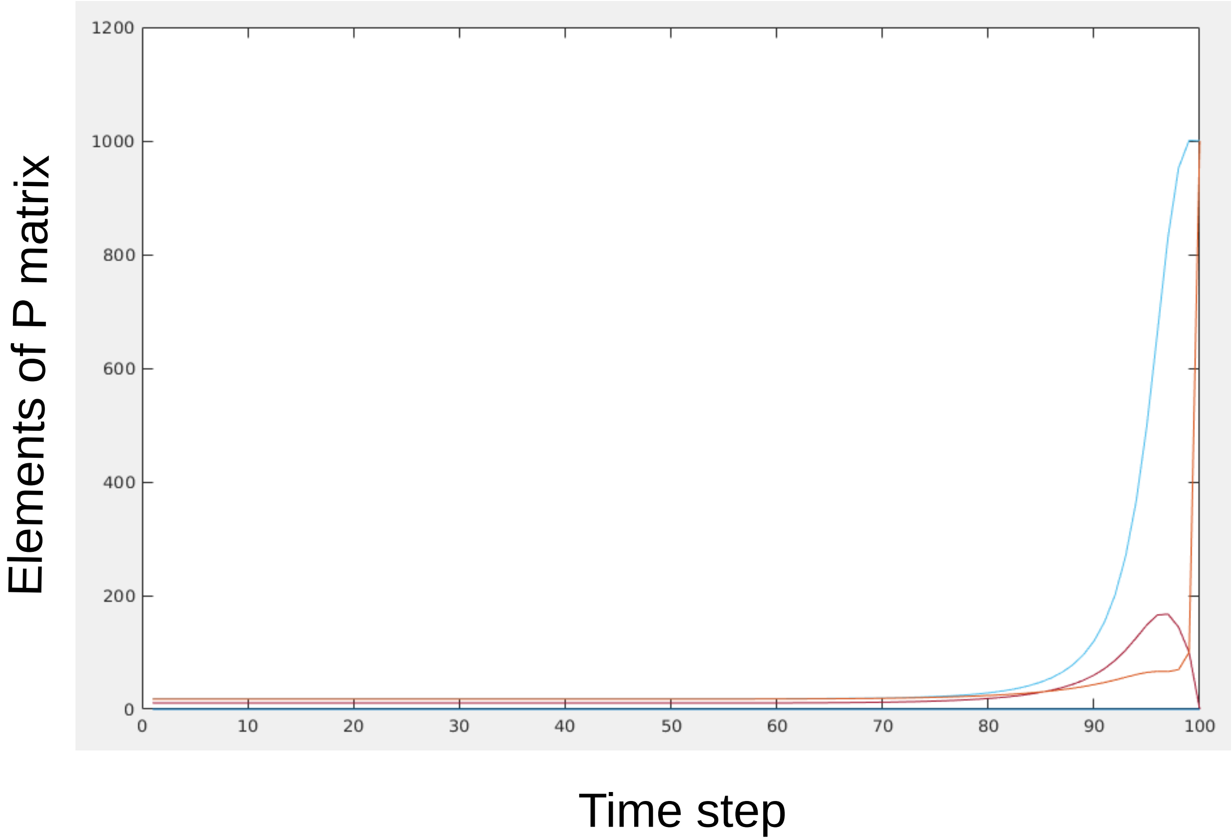
- optimal if you only have *T* time steps to do the job

But, what if time doesn't end in *T* steps?

One idea:

- at each time step, assume that you *always* have *T* more time steps to go
- this is called a *receding horizon* controller

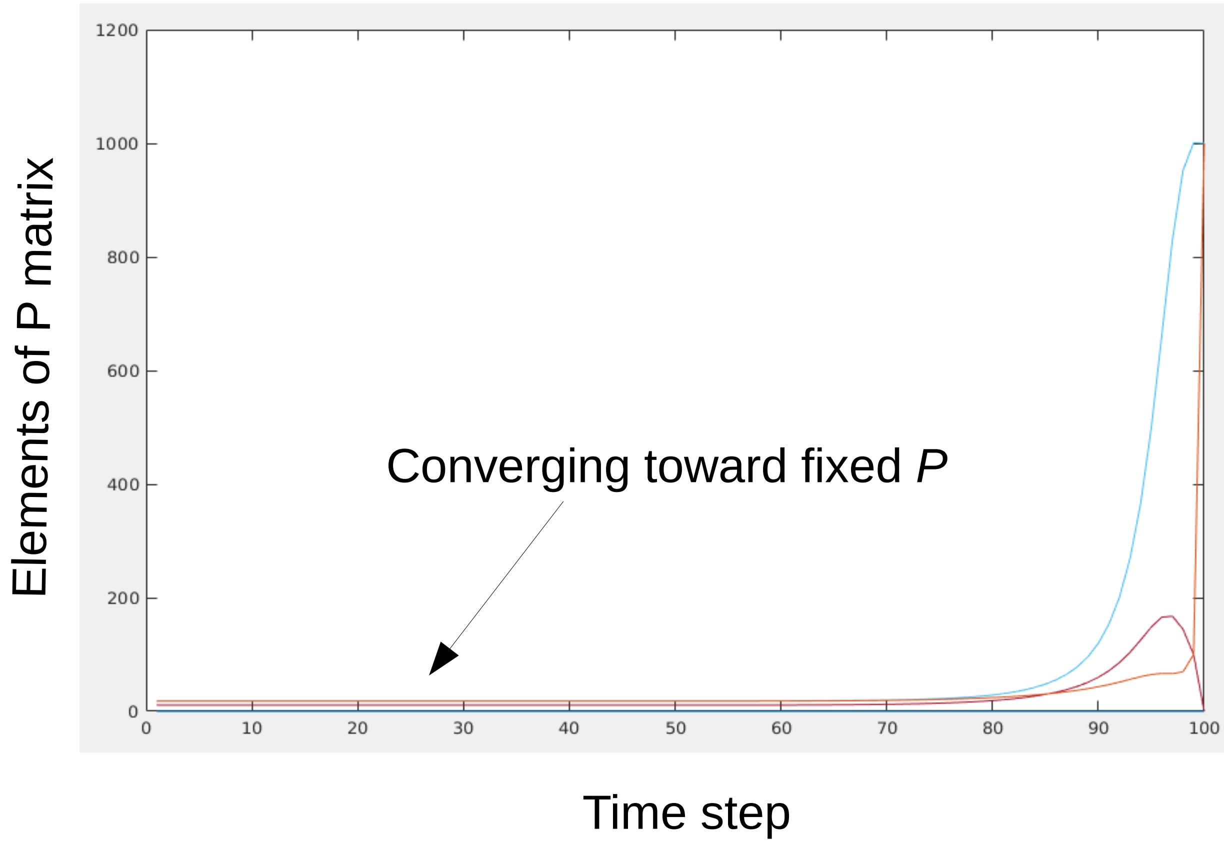
The infinite horizon case



Notice that elt's of *P* stop changing (much) more than 20 or 30 time steps prior to horizon.

– what does this imply about the infinite horizon case?

The infinite horizon case



Notice that elt's of *P* stop changing (much) more than 20 or 30 time steps prior to horizon.

– what does this imply about the infinite horizon case?

The infinite horizon case

We can solve for the infinite horizon *P* exactly:

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

$$P = Q + A^T P A - A^T P B(R + B^T P B)^{-1} B^T A$$

Discrete Time Algebraic Riccati Equation

So, what are we optimizing for now?

<u>Given:</u>

System: $$x_{t+1} = Ax_t + Bu_t$$

Cost function: $$J(X,U) = \sum_{t=1}^{\infty} x_t^T Q x_t + u_t^T R u_t$$

where: $$X = (x_1, \ldots, x_\infty)$$ $$U = (u_1, \ldots, u_\infty)$$

Initial state: x_1

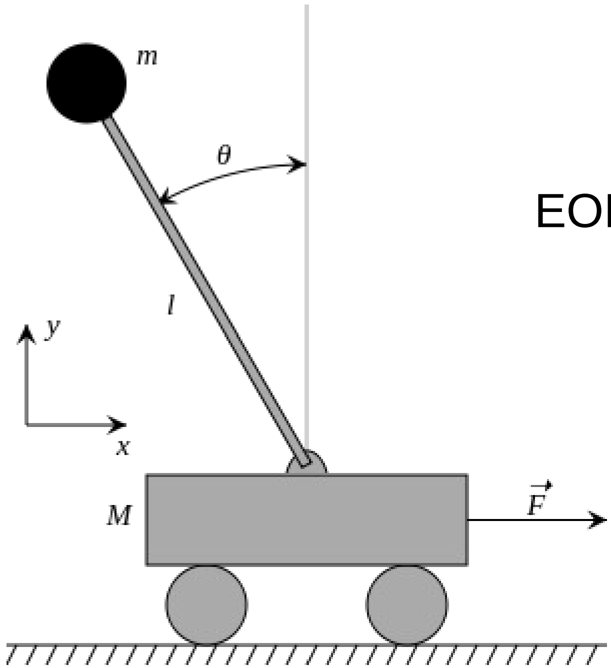
<u>Calculate:</u> *U* that minimizes J(*X*,*U*)

So, how do we control this thing?

Overview

1. expressing a linear system in state space form
2. discrete time linear optimal control (LQR)
3. linearizing around an operating point
4. linear model predictive control
5. LQR variants
6. model predictive control for non-linear systems

Inverted pendulum

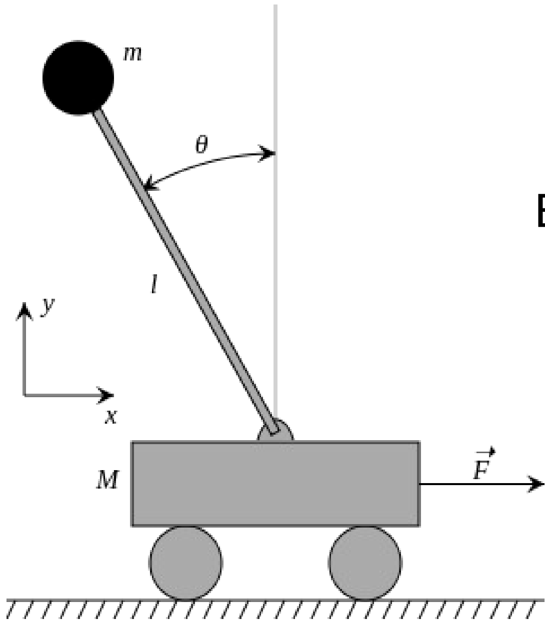


EOM for pendulum: $\ddot{\theta} + \frac{g}{l}\sin\theta = 0$

How do we get this system in the standard form: $x_{t+1} = Ax_t + Bu_t$

?

Inverted pendulum

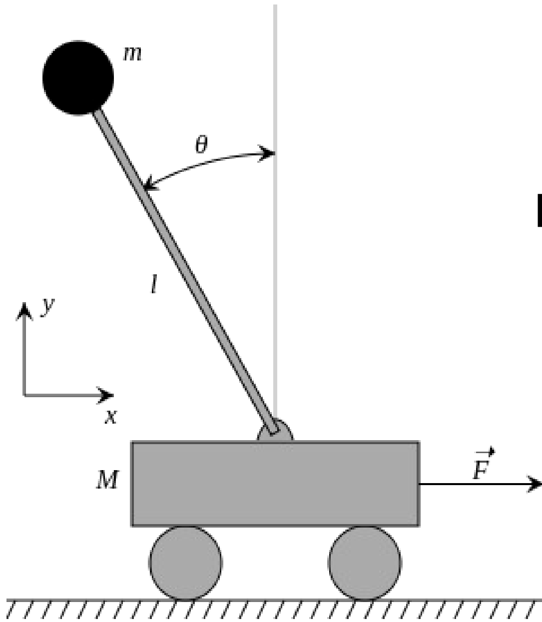


EOM for pendulum: $$\ddot{\theta} + \frac{g}{l}\sin\theta = 0$$

How do we get this system in the standard form: $$x_{t+1} = Ax_t + Bu_t$$

$$\theta_{t+1} = \theta_t + \dot{\theta}_t dt$$
$$\dot{\theta}_{t+1} = \dot{\theta}_t - \frac{g}{l}\sin\theta_t dt$$

Inverted pendulum



EOM for pendulum: $\ddot{\theta} + \frac{g}{l} \sin \theta = 0$

How do we get this system in the standard form: $x_{t+1} = Ax_t + Bu_t$

$$\theta_{t+1} = \theta_t + \dot{\theta}_t dt$$

$$\dot{\theta}_{t+1} = \dot{\theta}_t - \frac{g}{l} \boxed{\sin \theta_t dt}$$!!!!!!!

Linearizing a non-linear system

<u>Idea:</u> use first-order Taylor series expansion

$$x_{t+1} = f(x_t) + Bu_t$$ ← original non-linear system

Linearizing a non-linear system

<u>Idea:</u> use first-order Taylor series expansion

$$x_{t+1} = f(x_t) + Bu_t$$

← original non-linear system

Linearize about x^* →

$$\approx f(x^*) + \underbrace{\frac{\partial f(x^*)}{\partial x}(x_t - x^*)}_{\text{first order term}} + Bu_t$$

first order term

Linearizing a non-linear system

<u>Idea:</u> use first

We just linearized the system about x^*

$$x_{t+1} = f(x_t) + Bu_t$$ ← original non-linear system

Linearize about x^* →

$$\approx f(x^*) + \underbrace{\frac{\partial f(x^*)}{\partial x}(x_t - x^*)}_{\text{first order term}} + Bu_t$$

first order term

Linearizing a non-linear system

$$x_{t+1} \approx f(x^*) + \frac{\partial f(x^*)}{\partial x}(x_t - x^*) + Bu_t$$

Suppose that x^* is a fixed point (or a steady state) of the system...

Then: $f(x^*) = x^*$

Linearizing a non-linear system

$$x_{t+1} \approx f(x^*) + \frac{\partial f(x^*)}{\partial x}(x_t - x^*) + Bu_t$$

Suppose that x^* is a fixed point (or a steady state) of the system...

Then: $f(x^*) = x^*$

$$x_{t+1} - f(x^*) \approx \frac{\partial f(x^*)}{\partial x}(x_t - x^*) + Bu_t$$

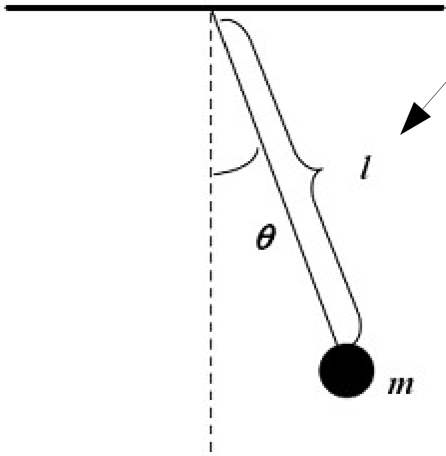
$$x_{t+1} - x^* \approx \frac{\partial f(x^*)}{\partial x}(x_t - x^*) + Bu_t$$

$$\bar{x}_{t+1} \approx \frac{\partial f(x^*)}{\partial x}\bar{x}_t + Bu_t$$ where $\bar{x}_t = x_t - x^*$

Change of coordinates

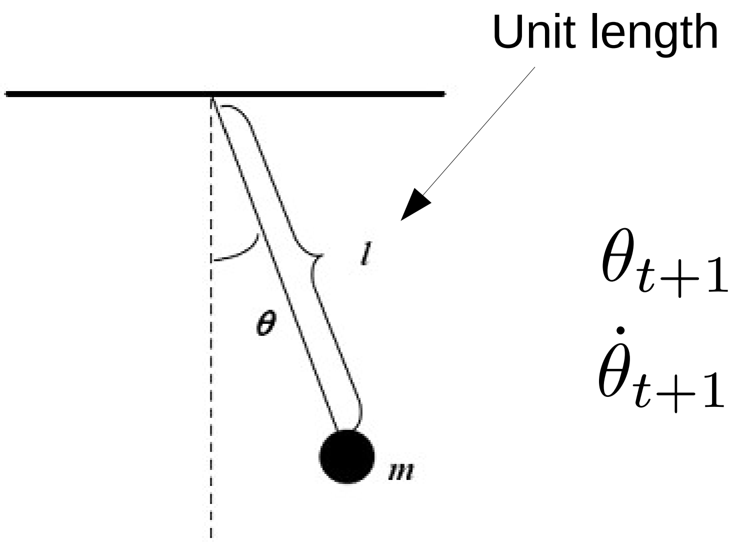
Example: pendulum

Unit length



$$\theta_{t+1} = \theta_t + \dot{\theta}_t dt$$
$$\dot{\theta}_{t+1} = \dot{\theta}_t - g \sin \theta_t dt$$

Example: pendulum

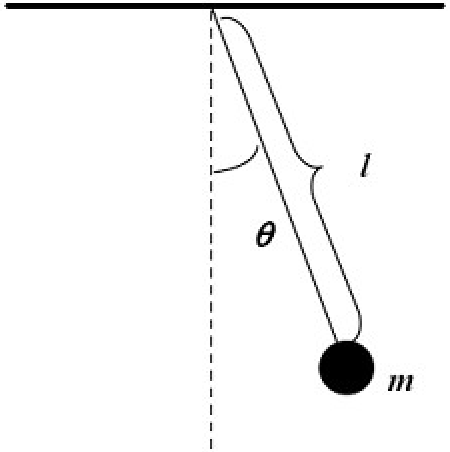


$$\theta_{t+1} = \theta_t + \dot{\theta}_t dt$$
$$\dot{\theta}_{t+1} = \dot{\theta}_t - g \sin \theta_t dt$$

$$\frac{\partial f(x^*)}{\partial x} = \begin{pmatrix} 1 & dt \\ -g \cos \theta_t dt & 1 \end{pmatrix}$$

Linearize about: $x^* = \begin{pmatrix} \pi \\ 0 \end{pmatrix} \longrightarrow \frac{\partial f(x^*)}{\partial x} = \begin{pmatrix} 1 & dt \\ g dt & 1 \end{pmatrix}$

Example: pendulum



$$\bar{x}_{t+1} \approx A\bar{x}_t + Bu_t \quad \text{where} \quad A = \begin{pmatrix} 1 & dt \\ gdt & 1 \end{pmatrix}$$

$$\bar{x}_t = x_t - \begin{pmatrix} \pi \\ 0 \end{pmatrix}$$

Overview

1. expressing a linear system in state space form
2. discrete time linear optimal control (LQR)
3. linearizing around an operating point
4. linear model predictive control
5. LQR variants
6. model predictive control for non-linear systems

Linear Model Predictive Control

Drawbacks to LQR: hard to encode constraints
– suppose you have a hard goal constraint?
– suppose you have piecewise linear state and action constraints?

Answer:
– solve control as a new optimization problem on every time step

Linear Model Predictive Control

Given:

System: $$x_{t+1} = Ax_t + Bu_t$$

Cost function: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

where: $$X = (x_1, \ldots, x_T)$$
$$U = (u_1, \ldots, u_{T-1})$$

Initial state: x_1

Calculate: *U* that minimizes J(*X*,*U*)

Linear Model Predictive Control

Given:

System: x

Cost function: $u_t^T R u_t$

We're going to solve this problem
by expressing it explicitly as
a quadratic program

Initial state:

Calculate:

U that minimizes J(*X*,*U*)

Quadratic program

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

Subject to: $$g_1^T x \leq h_1, \ldots, g_n^T x \leq h_n$$

$$a_1^T x = b_1, \ldots, a_n^T x = b_n$$

Quadratic program

Constants are part of problem statement:

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

Subject to: $$g_1^T x \leq h_1, \ldots, g_n^T x \leq h_n$$

$$a_1^T x = b_1, \ldots, a_n^T x = b_n$$

x is the variable

Problem: find the value of x that minimizes the objective subject to the constraints

Quadratic program

Quadratic objective function

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

Linear inequality constraints

Subject to: $$g_1^T x \leq h_1, \ldots, g_n^T x \leq h_n$$

$$a_1^T x = b_1, \ldots, a_n^T x = b_n$$

Linear equality constraints

Quadratic program

$$P = P^T \geq 0$$

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

Subject to: $$g_1^T x \leq h_1, \ldots, g_n^T x \leq h_n$$

$$a_1^T x = b_1, \ldots, a_n^T x = b_n$$

Quadratic program

$$P = P^T \geq 0$$

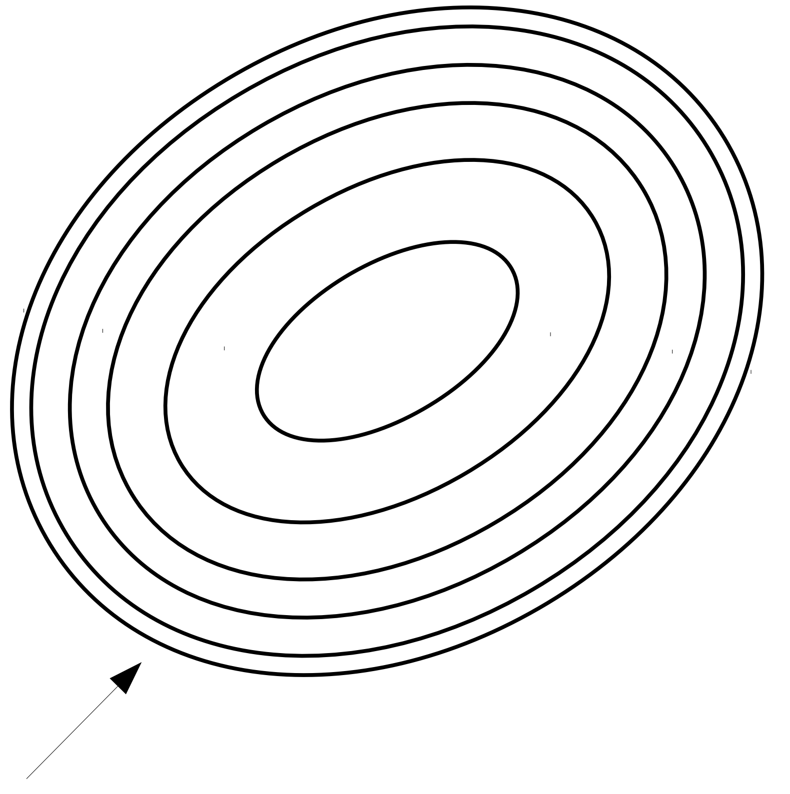
Why?

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

Subject to: $$g_1^T x \leq h_1, \ldots, g_n^T x \leq h_n$$

$$a_1^T x = b_1, \ldots, a_n^T x = b_n$$

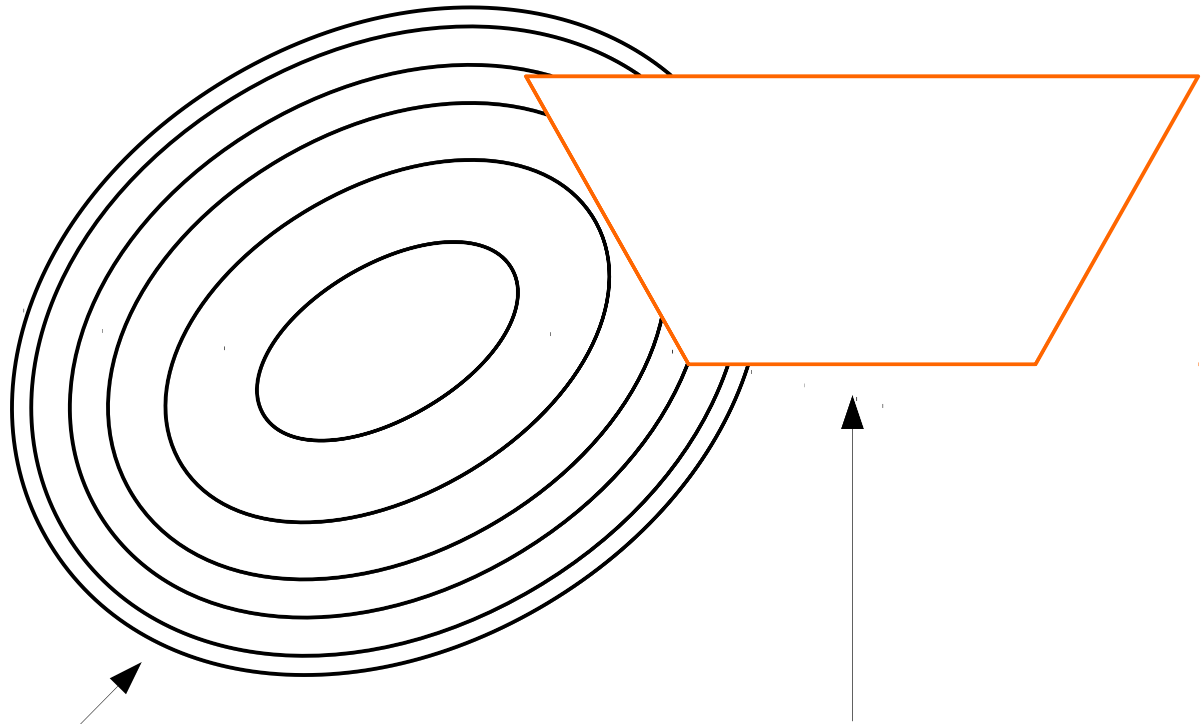
Quadratic program



$$\frac{1}{2}x^T P x + q^T x + r$$

Quadratic objective function

Quadratic program



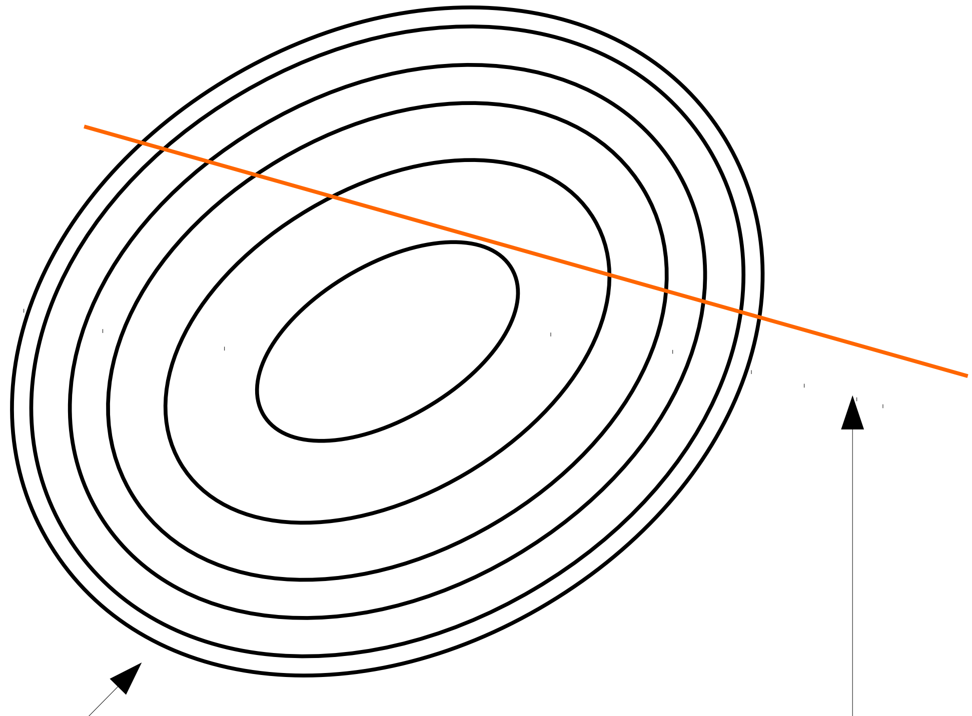
$$\frac{1}{2}x^TPx + q^Tx + r$$

Quadratic objective function

$$g_1^Tx \leq h_1, \ldots, g_n^Tx \leq h_n$$

Inequality constraints

Quadratic program



$$\frac{1}{2}x^T P x + q^T x + r$$

Quadratic objective function

$$a_1^T x = b_1, \ldots, a_n^T x = b_n$$

equality constraints

QP versus Unconstrained Optimization

Original QP

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

Subject to: $$g_1^T x \leq h_1, \ldots, g_n^T x \leq h_n$$

$$a_1^T x = b_1, \ldots, a_n^T x = b_n$$

QP versus Unconstrained Optimization

Unconstrained version of original QP

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

~~Subject to:~~ $$g_1^T x \leq h_1, \ldots, g_n^T x \leq h_n$$

$$a_1^T x = b_1, \ldots, a_n^T x = b_n$$

QP versus Unconstrained Optimization

Unconstrained version of original QP

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

How do we minimize this expression?

QP versus Unconstrained Optimization

Unconstrained version of original QP

Minimize: $$\frac{1}{2}x^T P x + q^T x + r$$

How do we minimize this expression?

$$\frac{\partial\left[\frac{1}{2}x^T P x + q^T x + r\right]}{\partial x} = 0$$

$$x = -P^{-1}q$$

Linear Model Predictive Control

Minimize: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

Subject to: $$x_{t+1} = Ax_t + Bu_t$$

$$x_1 = \text{start state}$$

$$x_T = \text{goal state}$$

Linear Model Predictive Control

Minimize: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

Subject to: $$x_{t+1} = Ax_t + Bu_t$$

$$x_1 = \text{start state}$$

$$x_T = \text{goal state}$$

What are the variables?

Linear Model Predictive Control

Minimize: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

Subject to: $$x_{t+1} = Ax_t + Bu_t$$

$$x_1 = \text{start state}$$

$$x_T = \text{goal state}$$

What other constraints might we want add?

Linear Model Predictive Control

Minimize: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

Subject to: $$x_{t+1} = Ax_t + Bu_t$$

$$x_1 = \text{start state}$$

$$x_T = \text{goal state}$$

$$|\dot{y}_t| \leq c$$

$$\dot{y}_{20} = 0$$

Linear Model Predictive Control

Minimize: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

Subject to: $$x_{t+1} = Ax_t + Bu_t$$

$$x_1 = \text{start state}$$

$$x_T = \text{goal state}$$

$$|\dot{y}_t| \leq c$$

$$\dot{y}_{20} = 0$$

Can't express these constraints in standard LQR

Linear MPC Receding Horizon Control

Re-solve the quadratic program on each time step:
– always plan another T time steps into the future

Minimize: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

Subject to: $$x_{t+1} = Ax_t + Bu_t$$

$$x_1 = \text{start state}$$

Controllability

A system is **controllable** if it is possible to reach any goal state from any other start state in a finite period of time.

When is a linear system controllable?

$$x_{t+1} = Ax_t + Bu_t$$

← It's property of the system dynamics...

Controllability

A system is **controllable** if it is possible to reach any goal state from any other start state in a finite period of time.

When is a linear system controllable?

Remember this?

$$x_{t+1} = Ax_t + Bu_t$$

$$\begin{pmatrix} x_1 \\ \vdots \\ x_T \end{pmatrix} = \begin{pmatrix} 0 & \dots & & & \\ B & 0 & \dots & & \\ AB & B & 0 & \dots & \\ A^2B & AB & B & 0 & \dots \\ \dots & & & & \\ A^{T-1}B & A^{T-2}B & \dots & \dots & B \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_{T-1} \end{pmatrix} + \begin{pmatrix} I \\ A \\ A^2 \\ \vdots \\ A^T \end{pmatrix} x_1$$

Controllability

$$\begin{pmatrix} x_1 \\ \vdots \\ x_T \end{pmatrix} = \begin{pmatrix} 0 & \dots & & & \\ B & 0 & \dots & & \\ AB & B & 0 & \dots & \\ A^2B & AB & B & 0 & \dots \\ \dots & & & & \\ A^{T-1}B & A^{T-2}B & \dots & \dots & B \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_{T-1} \end{pmatrix} + \begin{pmatrix} I \\ A \\ A^2 \\ \vdots \\ A^T \end{pmatrix} x_1$$

What property must this matrix have?

Controllability

$$\begin{pmatrix} x_1 \\ \vdots \\ x_T \end{pmatrix} = \begin{pmatrix} 0 & \dots & & & \\ B & 0 & \dots & & \\ AB & B & 0 & \dots & \\ A^2B & AB & B & 0 & \dots \\ \dots & & & & \\ A^{T-1}B & A^{T-2}B & \dots & \dots & B \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_{T-1} \end{pmatrix} + \begin{pmatrix} I \\ A \\ A^2 \\ \vdots \\ A^T \end{pmatrix} x_1$$

This submatrix must be full rank.

– *i.e.* the rank must equal the dimension of the state space

NonLinear Model Predictive Control

Given:

System: $x_{t+1} = Ax_t + Bu_t$

Cost function: $J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$

where: $X = (x_1, \ldots, x_T)$
$U = (u_1, \ldots, u_{T-1})$

Initial state: x_1

Calculate: *U* that minimizes J(*X*,*U*)

NonLinear Model Predictive Control

Given:

System: $x_{t+1} = Ax_t + Bu_t$

Cost function: $J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$

where: $X = (x_1, \ldots, x_T)$

$U = (u_1, \ldots, u_{T-1})$

Initial state: x_1

Calculate: *U* that minimizes J(*X*,*U*)

NonLinear Model Predictive Control

Given:

System: $$x_{t+1} = f(x_t, u_t)$$

Cost function: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

where: $$X = (x_1, \ldots, x_T)$$
$$U = (u_1, \ldots, u_{T-1})$$

Initial state: x_1

Calculate: *U* that minimizes J(*X*,*U*)

NonLinear Model Predictive Control

Minimize: $$J(X,U) = x_T^T Q_T x_T + \sum_{t=1}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

Subject to: $$x_{t+1} = f(x_t, u_t)$$
$$x_1 = \text{start state}$$
$$x_T = \text{goal state}$$

But, this is a nonlinear constraint
– so how do we solve it now?

<u>Sequential quadratic programming</u>
– iterative numerical optimization for problems with non-convex objectives or constraints
– similar to Newton's method, but it incorporates constraints
– on each step, linearize the constraints about the current iterate
– implemented by FMINCON in matlab...