Linear Optimal Control (LQR)

Robert Platt
Northeastern University

The linear control problem

Given:

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

The linear control problem

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

The linear control problem

Given:

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

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

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

Given:

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

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

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

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

The linear control problem

<u>Given:</u>

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

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

System:

Cost functi

Important problem!

How do we solve it?

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

$$x_T)$$

$$u_{T-1})$$

Initial state: $\quad x_1$

<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

$$x_1 = x_1$$

$$x_2 = Ax_1 + Bu_1$$

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

$$x_4 = \ldots$$

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

One solution: least squares

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

$$X = GU + Hx_1$$

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:

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

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

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

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

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

Initial state: x_1

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

One solution: least squares

<u>Given:</u>

System: $$X = GU + Hx_1$$

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

Initial state: x_1

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

One solution: least squares

Substitute *X* into *J*:

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

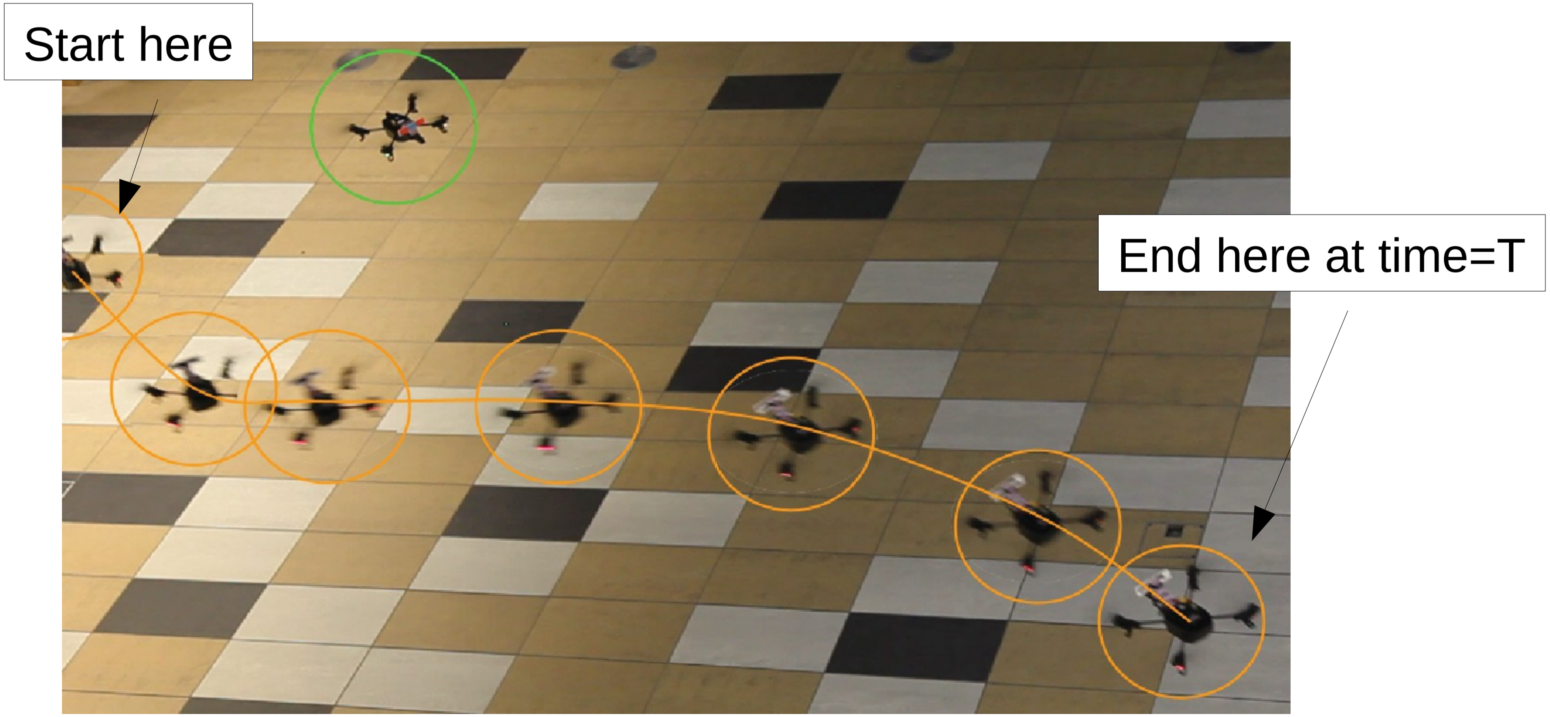
$$J(X,U) = U^T (G^T \mathbb{Q} G) U + U^T \mathbb{R} U + 2H^T x_1^T \mathbb{Q} G U$$

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

Bellman solution

<u>Cost-to-go function:</u> V(x)

– the cost that we have yet to experience if we travel along the minimum cost path.

– given the cost-to-go function, you can calculate the optimal path/policy

<u>Example:</u>

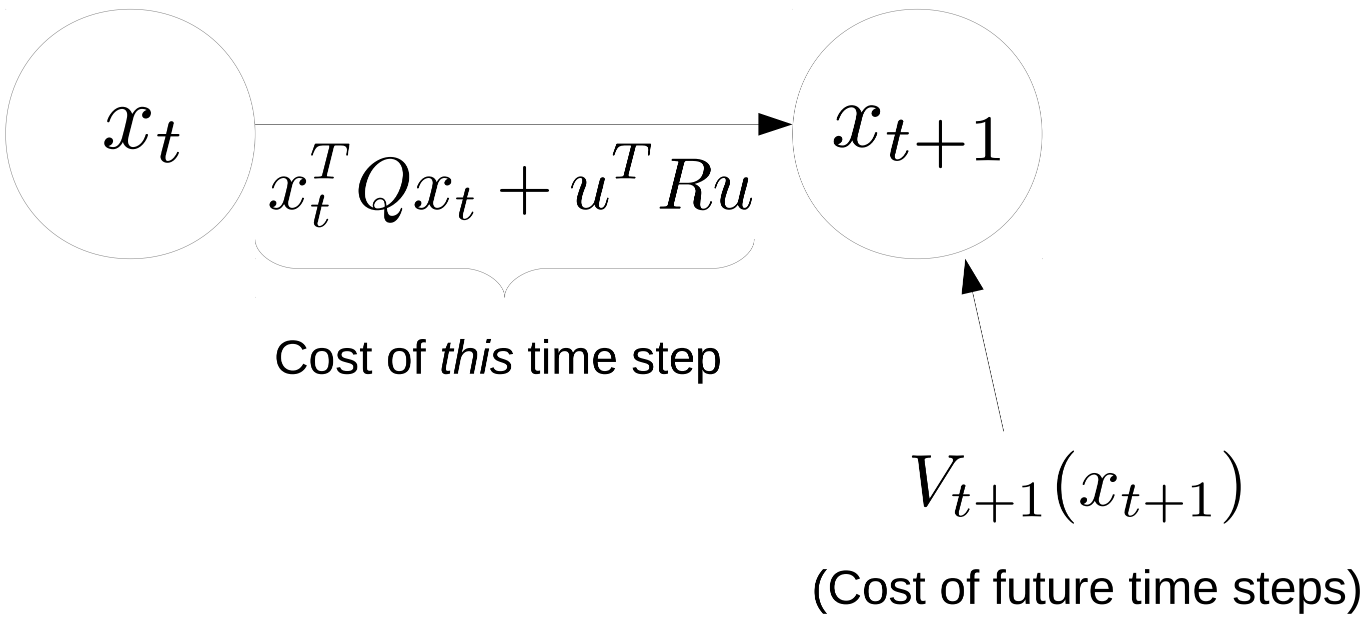
The number in each cell describes the number of steps “to-go” before reaching the goal state

7	**18**	17	16	15	14	13	12	11	10	9	9	9	9	9	9	9
6	17	17	16	15	14	13	12	11	10	9	8	8	8	8	8	8
5	17	16	16	15	14	13	12	11	10	9	8	7	7	7	7	7
4	17	16	15	15	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	6	6	6	6
3	17	16	15	14	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	5	5	5	5
2	17	16	15	14	13	12	11	10	9	8	7	6	5	4	4	4
1	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	3
0	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	**2**
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Bellman solution

Bellman optimality principle:

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



Bellman solution

Bellman optimality principle:

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

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

Bellman solution

<u>Bellman optimality principle:</u>

Cost-to-go from state *x* at time *t*

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

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

Cost incurred **on** this time step

Cost incurred **after** this time step

Bellman solution

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: $P_t = P_t^T \geq 0$

Bellman solution

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:

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

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

Bellman solution

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.

Bellman solution

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

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

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}...

Bellman solution

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

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

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}...

Bellman solution

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

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

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

Bellman solution

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

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

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

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

Bellman solution

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

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

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

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

Bellman solution

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

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

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

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

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

Bellman solution

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

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

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

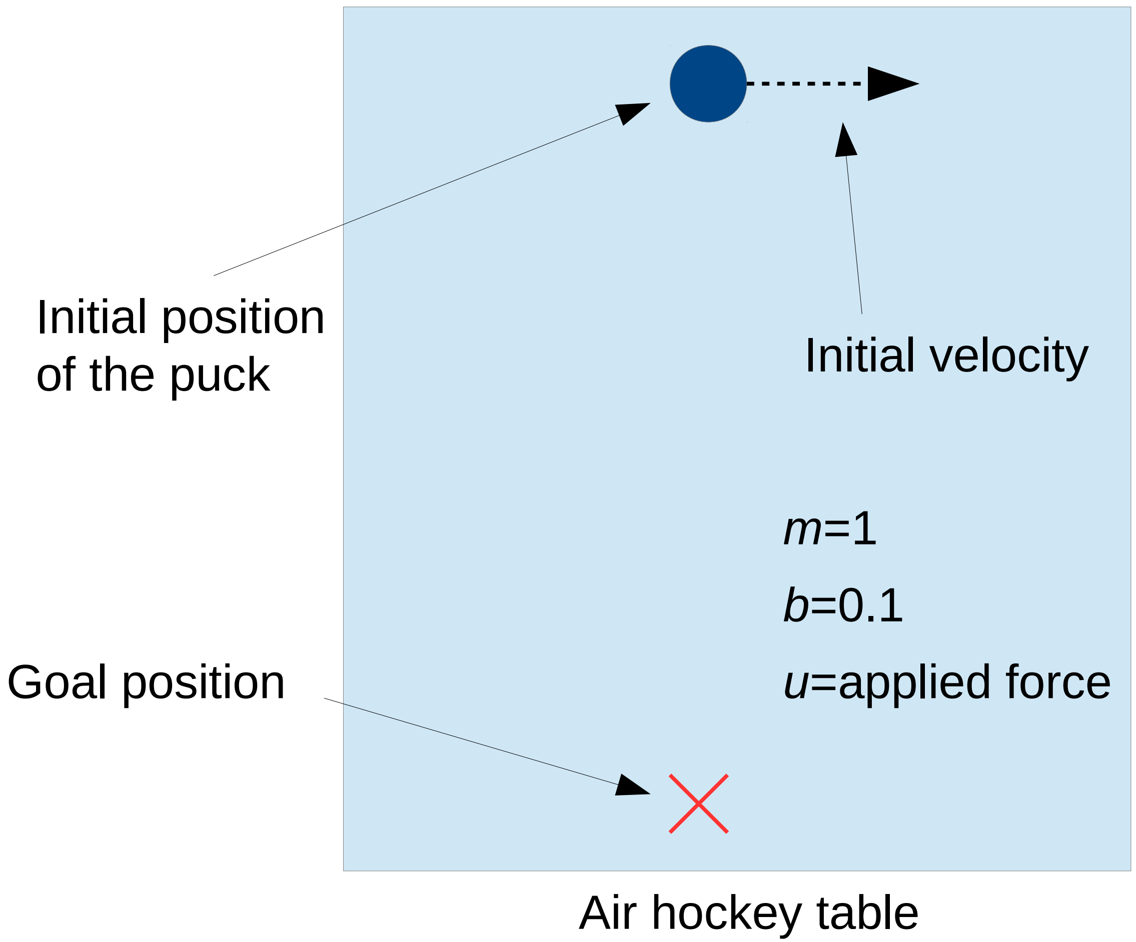
Dynamic Riccati Equation

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

$$P_t$$

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

Example: planar double integrator



Build the LQR controller for:

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

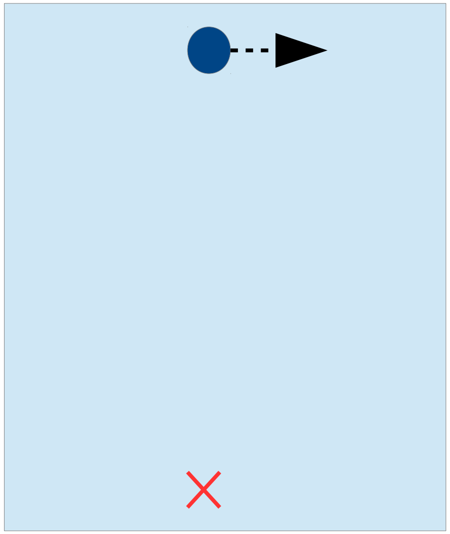
Time horizon: $T = 100$

Cost fn: $Q_T = 1000I$

$Q = I$

$R = I$

Example: planar double integrator

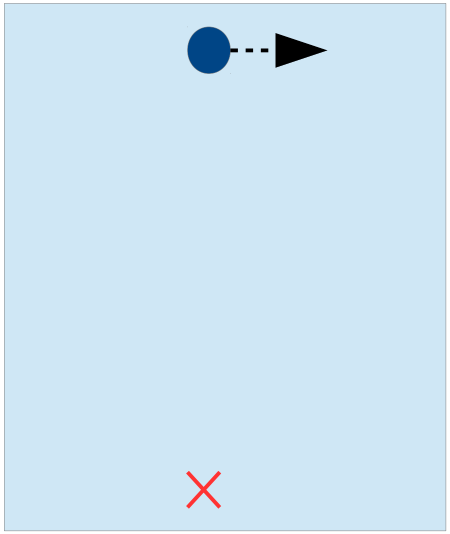


Air hockey table

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

HOW?

Example: planar double integrator



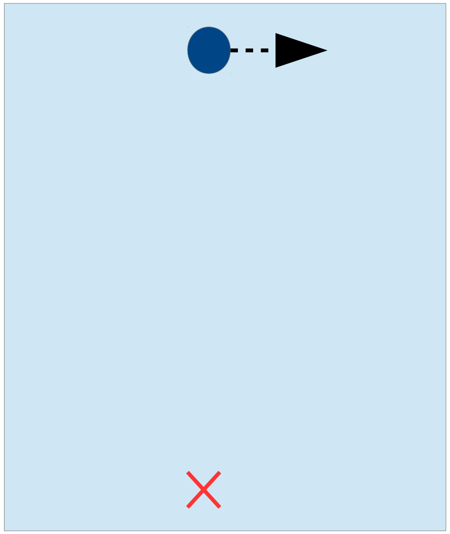
Air hockey table

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

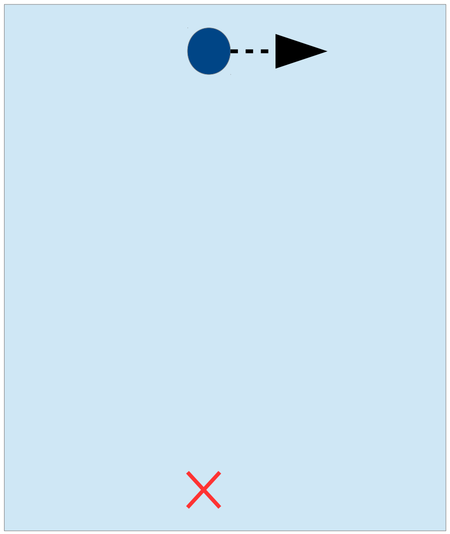
Step 1:

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

Step 1:

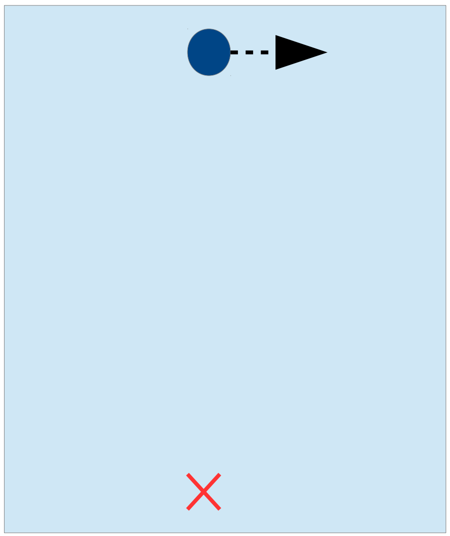
Calculate P backward from T: P_100, P_99, P_98, … , P_1

$$P_{100} = 1000I$$

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

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

Example: planar double integrator



Air hockey table

<u>Step 1:</u>

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

$$P_{100} = 1000I$$

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

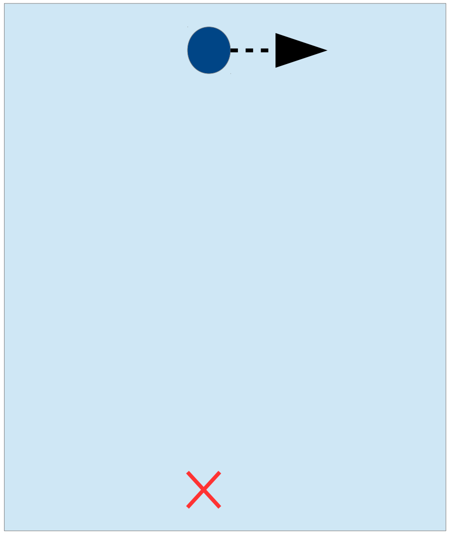
$$P_{99} = \begin{pmatrix} 1001 & 0 & 1000 & 0 \\ 0 & 1001 & 0 & 1000 \\ 1000 & 0 & 1001 & 0 \\ 0 & 1000 & 0 & 1001 \end{pmatrix}$$

$$\vdots$$

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

$$\vdots$$

Example: planar double integrator



Air hockey table

Step 2:

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

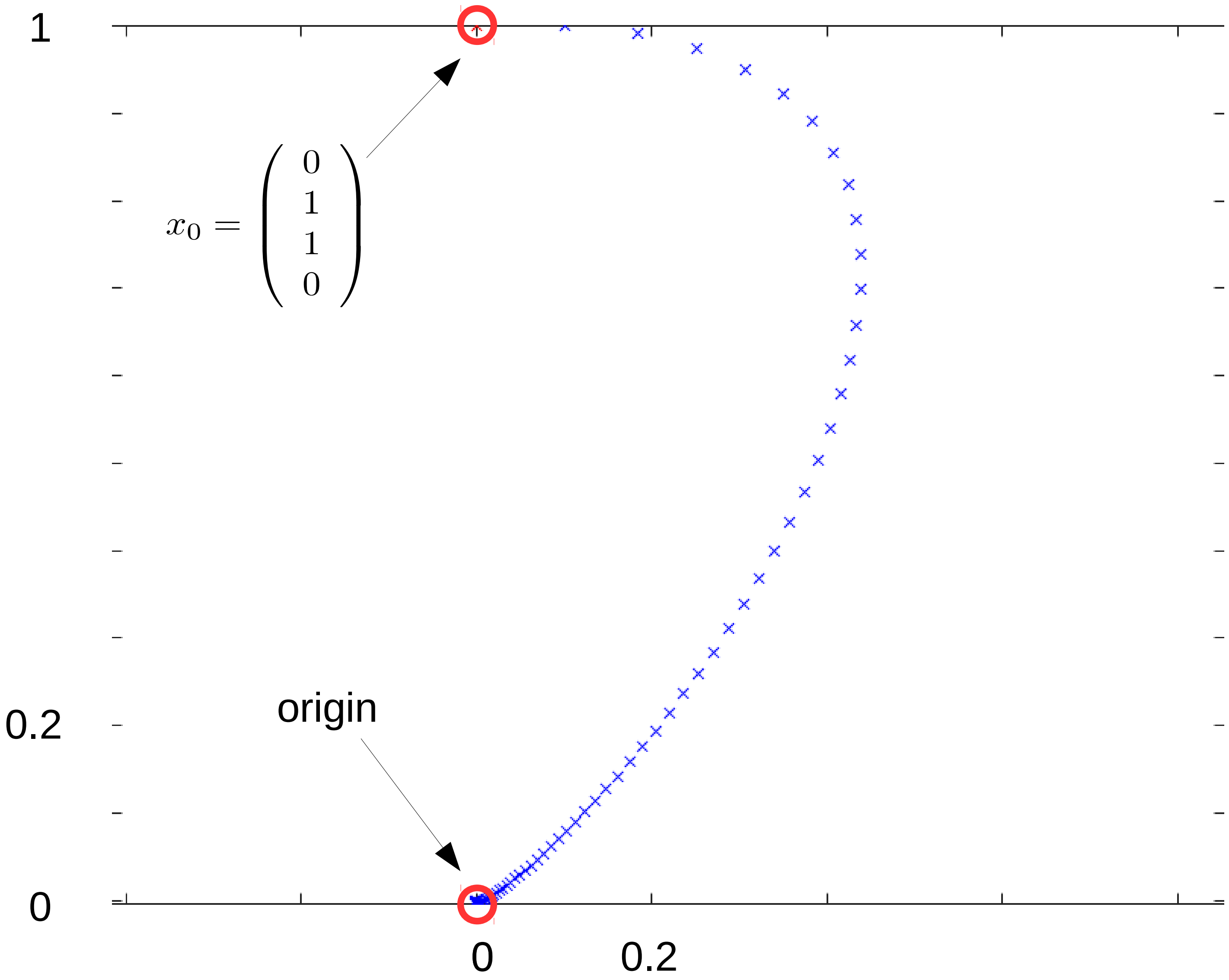
$$u_1 = -(R + B^T P_1 B)^{-1} B^T P_1 A x$$

$$\vdots$$

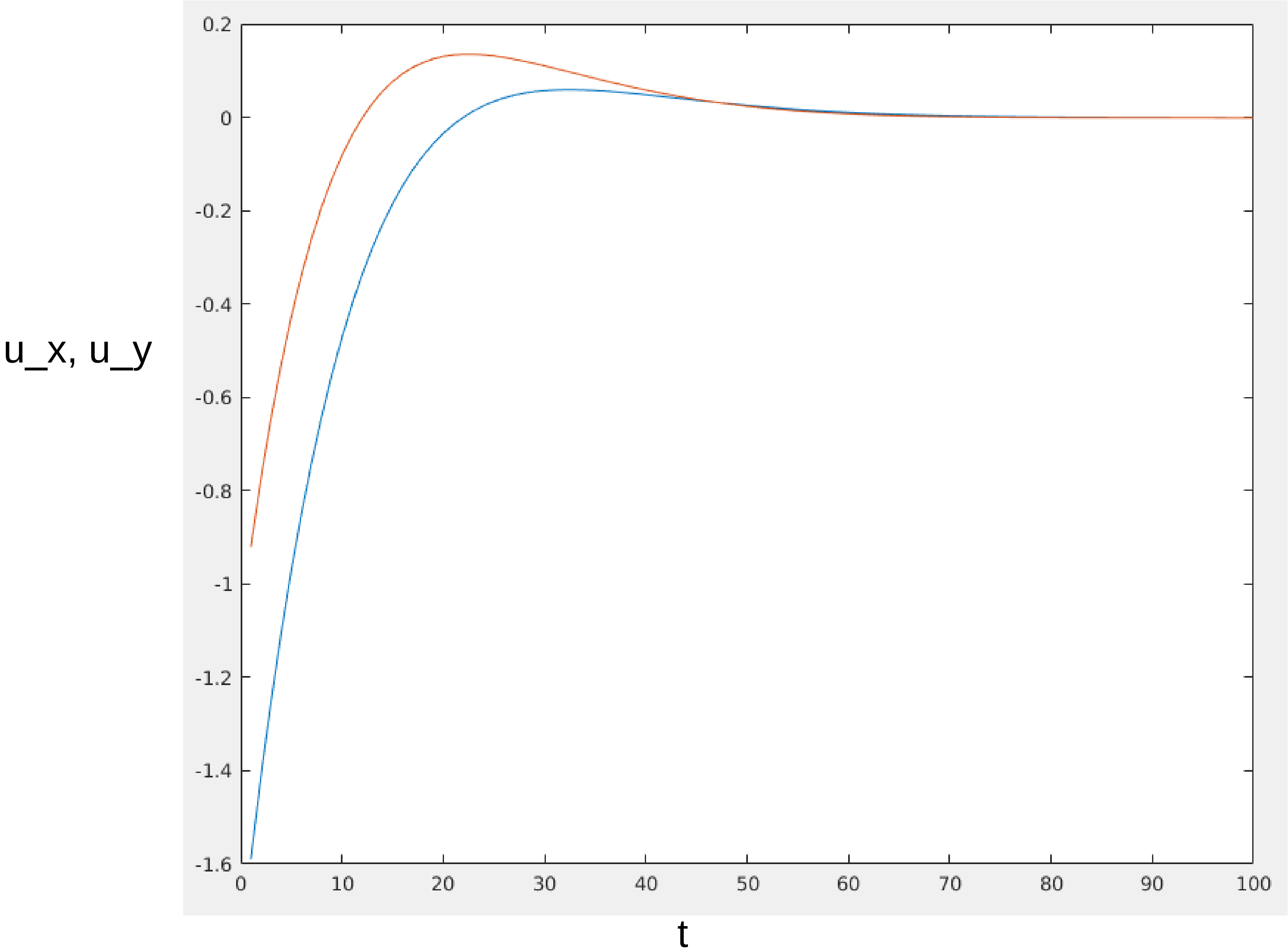
$$u_2 = -(R + B^T P_2 B)^{-1} B^T P_2 A x$$

$$\vdots$$

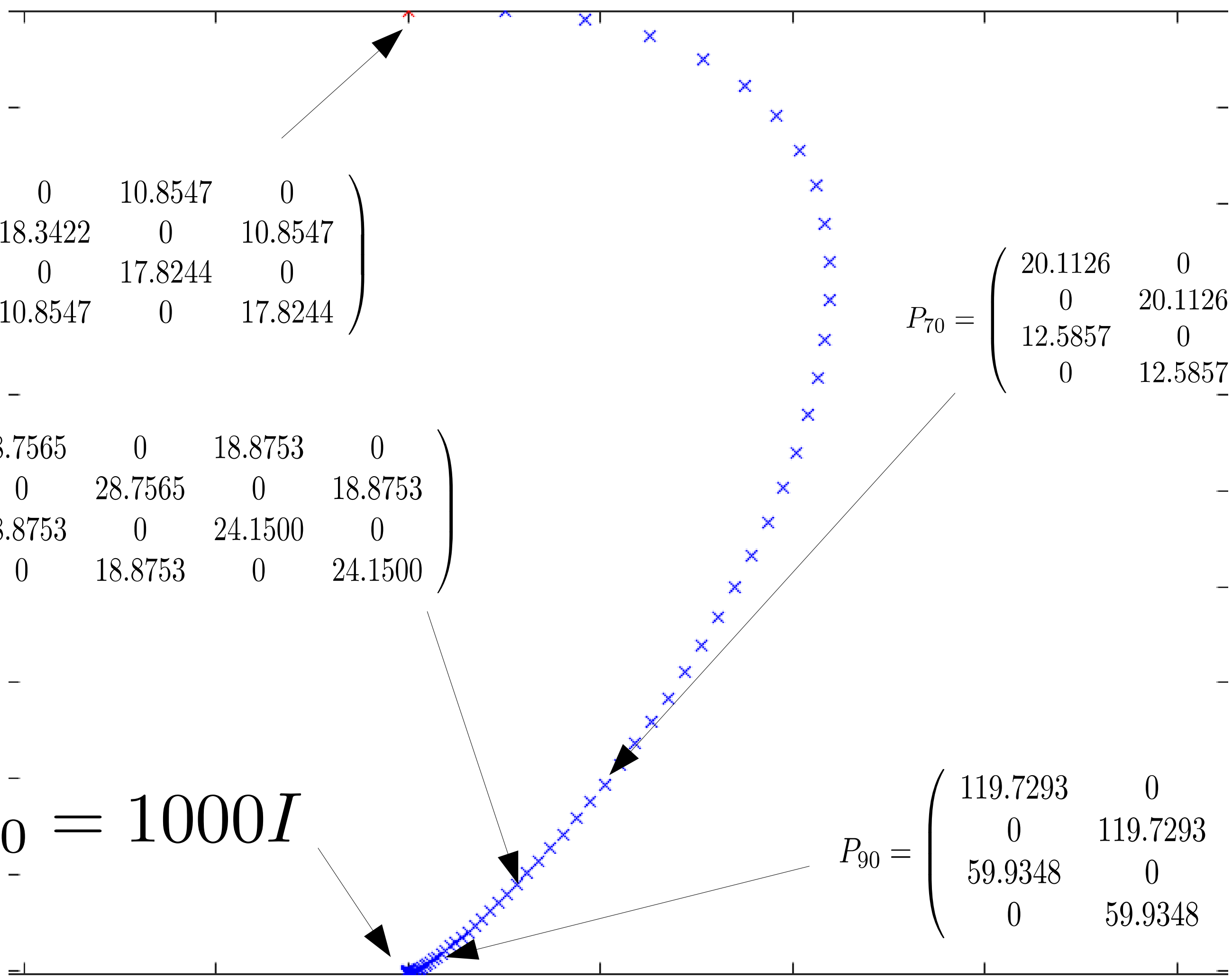
Example: planar double integrator



Example: planar double integrator



Example: planar double integrator



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

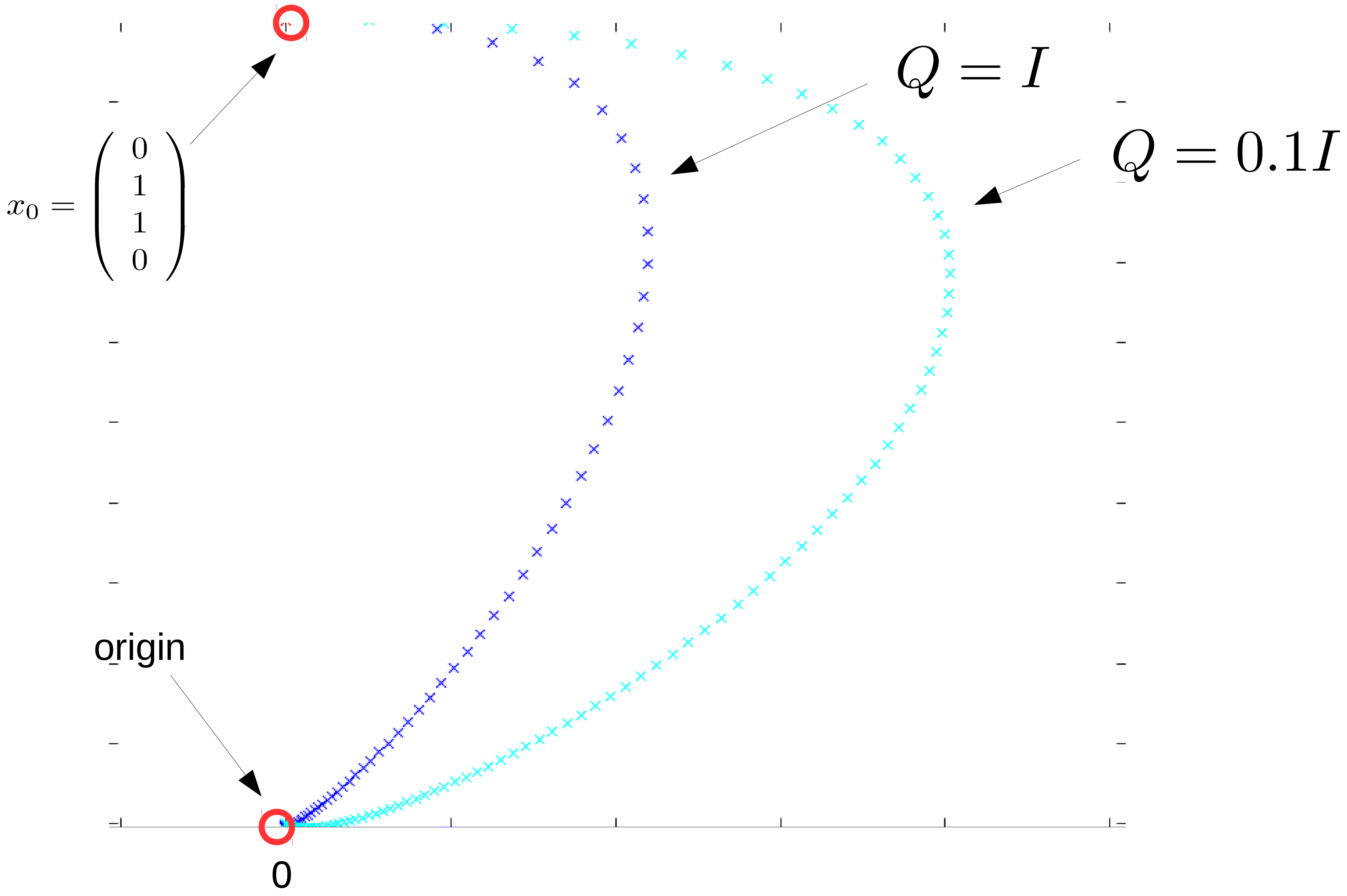
$$P_{70} = \begin{pmatrix} 20.1126 & 0 & 12.5857 & 0 \\ 0 & 20.1126 & 0 & 12.5857 \\ 12.5857 & 0 & 19.5454 & 0 \\ 0 & 12.5857 & 0 & 19.5454 \end{pmatrix}$$

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

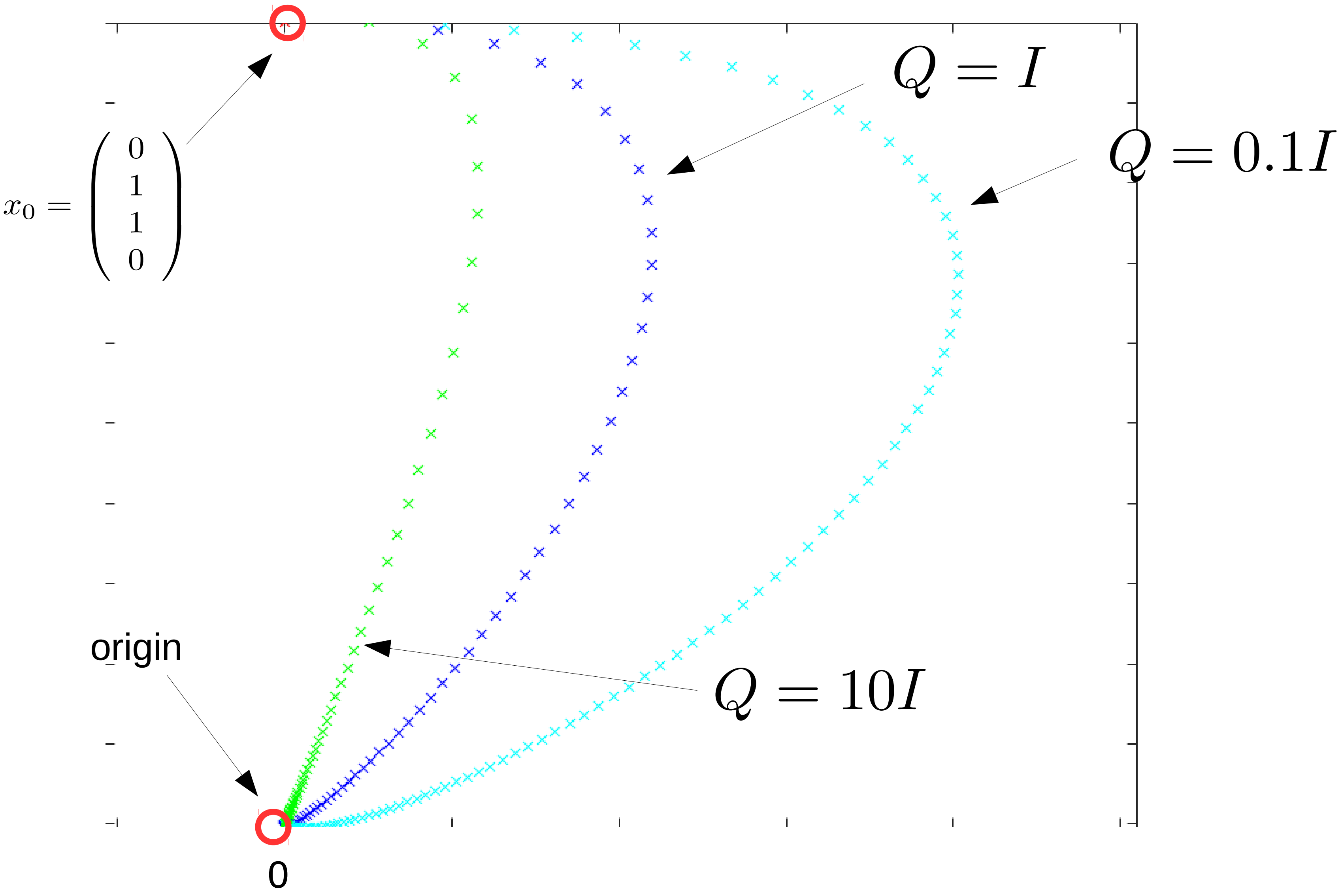
$$P_{100} = 1000I$$

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

Example: planar double integrator



Example: planar double integrator



The infinite horizon case

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

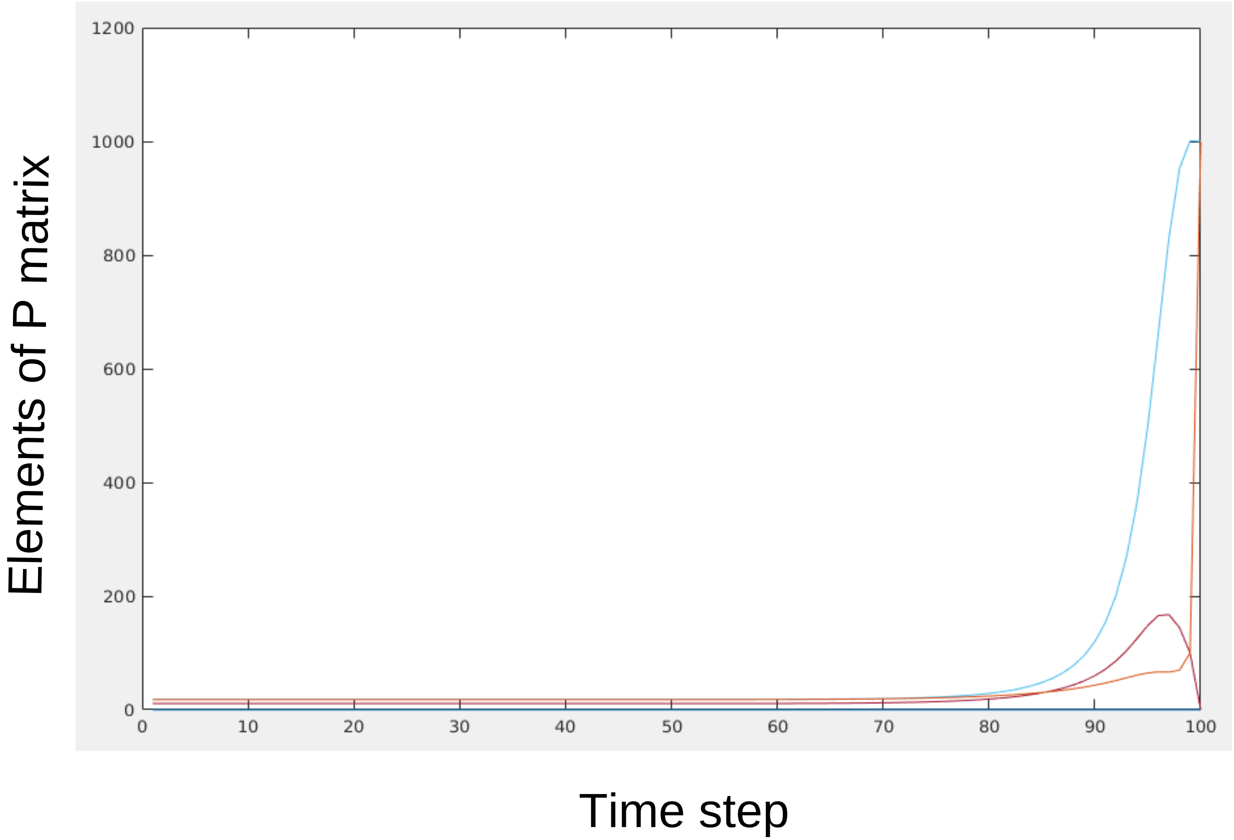
- optimal if you only have *T* time steps to do the job

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

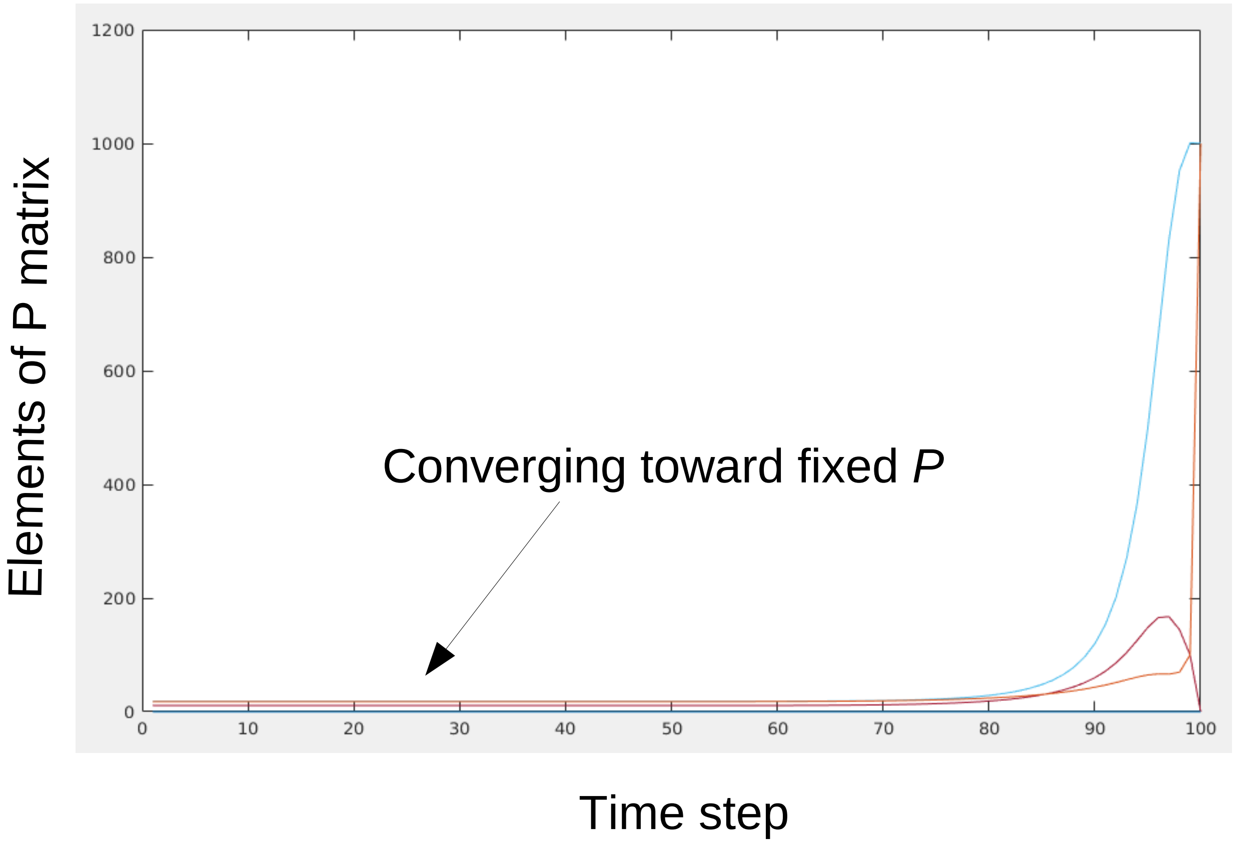
The infinite horizon case



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:

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

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

Discrete Time Algebraic Riccati Equation

So, what are we optimizing for now?

Given:

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

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

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 **<u>controllable</u>** 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?

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

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

Controllability

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

What property must this matrix have?

Controllability

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

This submatrix must be full rank.

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