Linear Quadratic Regulator

Instructor: Chris Mavrogiannis

TAs: Kay Ke, Gilwoo Lee, Matt Schmittle

*Slides based on or adapted from Sanjiban Choudhury, Drew Bagnell

Logistics

New Office Hours

Chris: Tuesdays at 1:00pm (CSE1 436)

Kay: Tuesdays at 4:00pm (CSE1 022)

Just for **this** week, Wednesday at 5:00pm

Gilwoo: Thursdays at 4:00pm (CSE1 022)

Schmittle: Fridays at 4:00pm (CSE1 022)

Different control laws

1. PID control
2. Pure-pursuit control
3. Lyapunov control
4. LQR
5. MPC

Recap of controllers

PID / Pure pursuit: Worked well, no provable guarantees

Lyapunov: Provable stability, convergence rate depends on gains

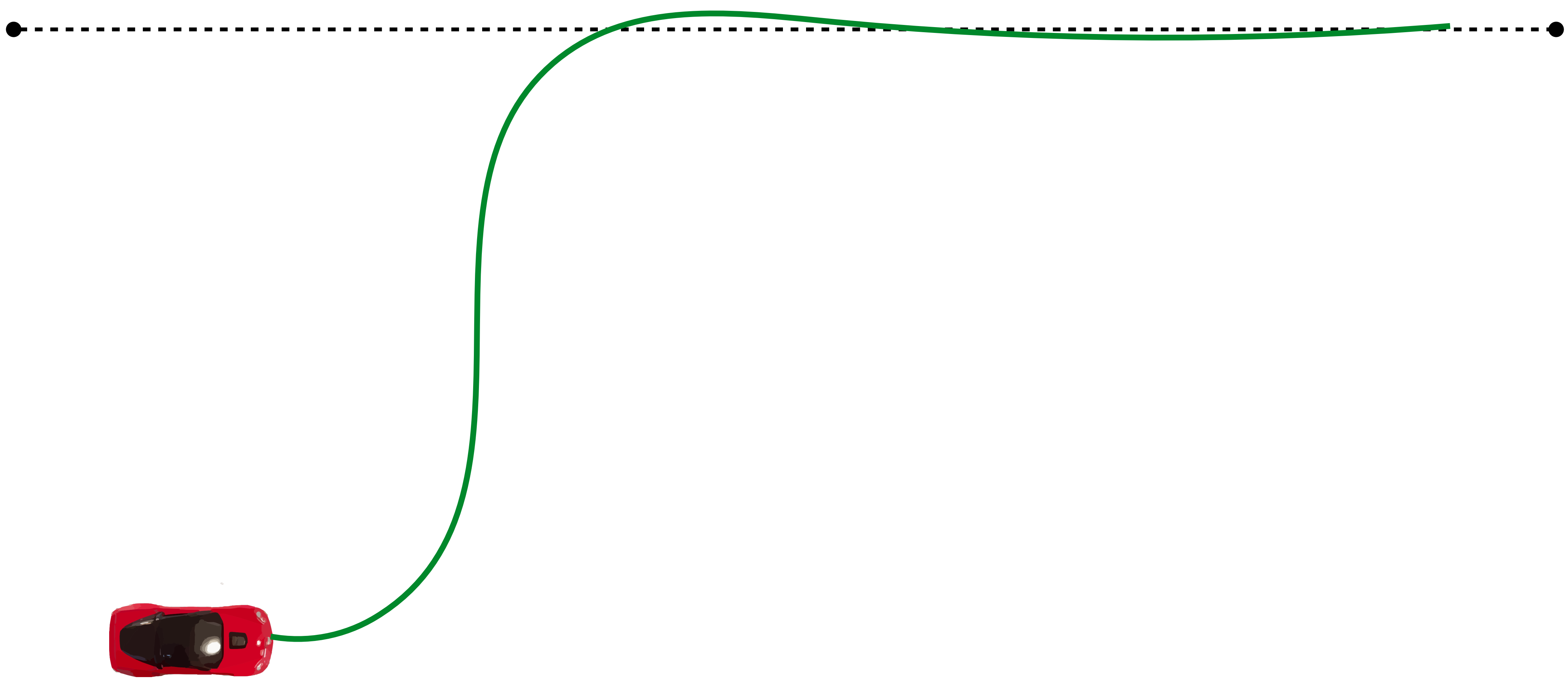
Table of controllers

	Control Law	Uses model	Stability Guarantee	Minimize Cost
PID	$u = K_p e + \ldots$	No	No	No
Pure Pursuit	$u = \tan^{-1}\left(\frac{2B\sin\alpha}{L}\right)$	Circular arcs	Yes - with assumptions	No
Lyapunov	$u = \tan^{-1}\left(-\frac{k_1 e_{ct} B}{\theta_e}\sin\theta_e - \frac{B}{V}k_2\theta_e\right)$	Non-linear	Yes	No

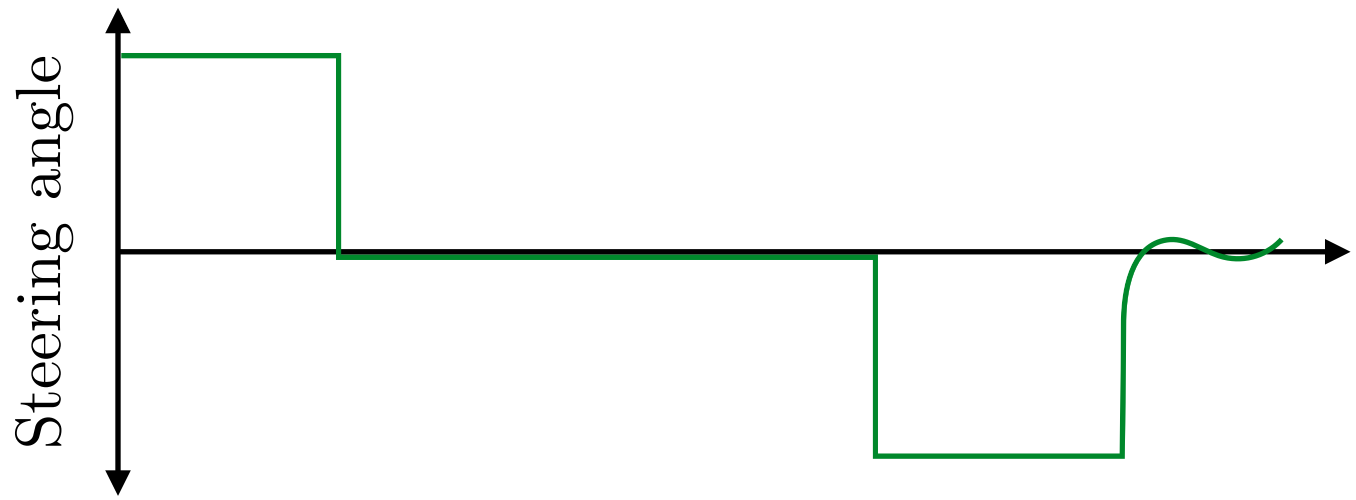
Is stability enough?

$$\lim_{t \to \infty} e(t) = 0$$

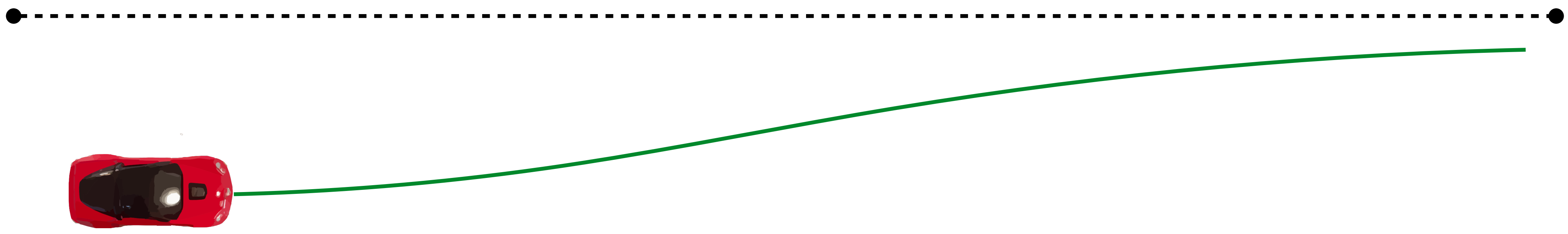
Is stability enough of a guarantee?



Control action changes abruptly - why is this bad?



Is stability enough of a guarantee?



What if we just choose really small gains?

Stability guarantees that the error will go to zero ...
but can take arbitrary long time

Question:

How do we trade-off both driving error to zero AND keeping control action small?

Key Idea:

Turn the problem into an optimization

$$\min_{u(t)} \int_0^\infty \left(w_1 e(t)^2 + w_2 u(t)^2 dt\right)$$

Optimal Control

Given:

$\bar{x}_0$

For $t = 0, 1, 2, \ldots, T$

Solve $$\min_{x,u} \sum_{k=0}^{T} c_k(x_k, u_k)$$

$$\text{s.t. } x_{k+1} = f(x_k, u_k), \forall k \in \{t, t+1, \ldots, T-1\}$$

$$x_t = \bar{x}_t$$

*Slide adapted from Ruslan Salakhutdinov

Special Case: Linear Quadratic Regulator (LQR)

Linear dynamics

$$f(x, u) = Ax + Bu$$

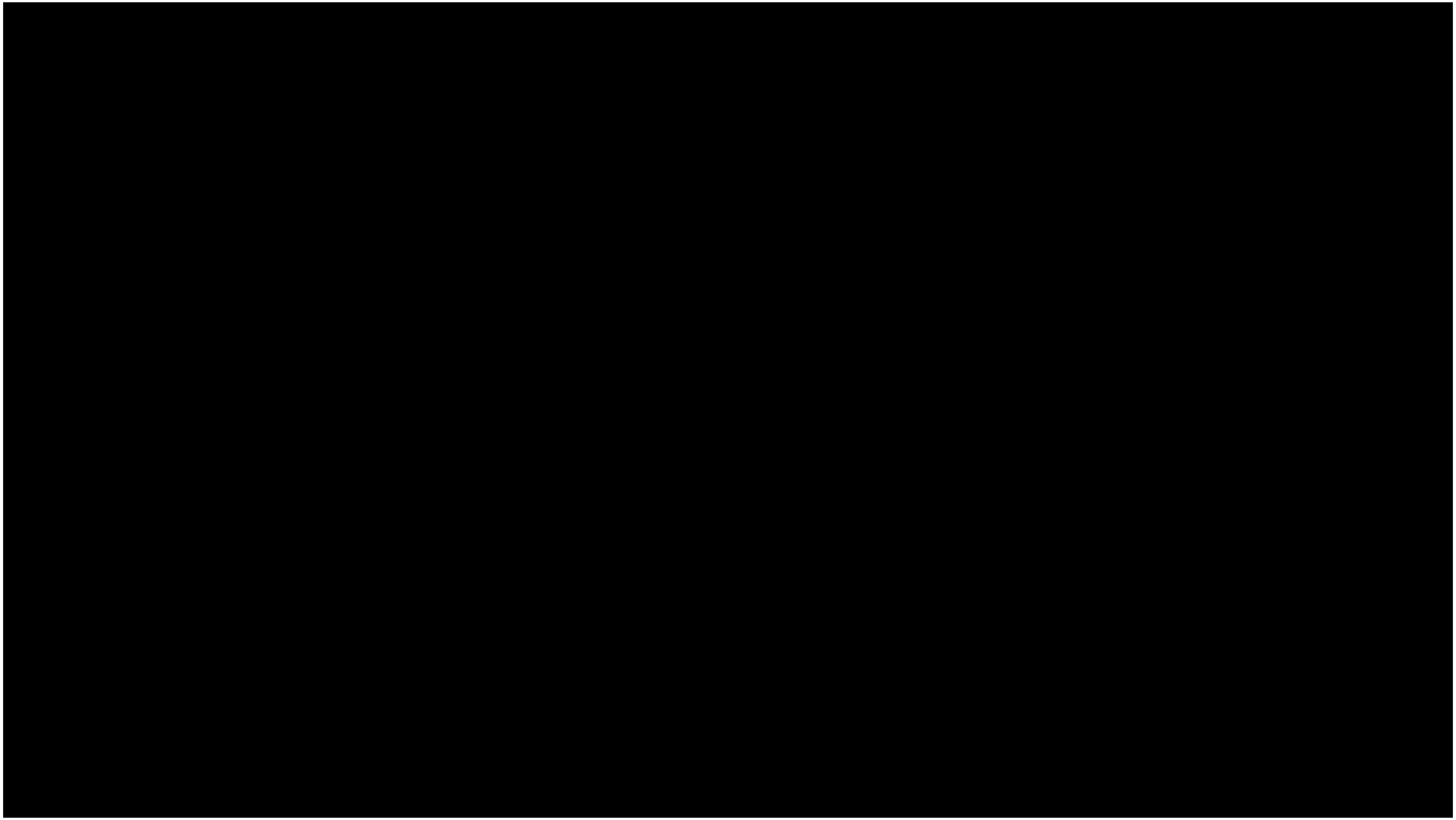
Quadratic cost

$$c(x, u) = x^T Q x + u^T R u$$

Trivia! :) (from http://www.uta.edu/utari/acs/history.htm)

In 1960 three major papers were published by R. Kalman and coworkers...
1. One of these [Kalman and Bertram 1960], presented the vital work of Lyapunov in the time-domain control of nonlinear systems.
2. The next [Kalman 1960a] discussed the optimal control of systems, providing the design equations for the linear quadratic regulator (LQR).
3. The third paper [Kalman 1960b] discussed optimal filtering and estimation theory, providing the design equations for the discrete Kalman filter.

LQR flying RC helicopters



(Excellent work by Pieter Abeel et al. https://people.eecs.berkeley.edu/~pabbeel/autonomous_helicopter.html)

The Linear Quadratic Regulator (LQR)

Given:

1. Linear dynamical system

$$x_{t+1} = Ax_t + Bu_t \quad \text{(assume controllable)}$$

2. A reference state which we are regulating around

$$x_{ref} = 0$$

3. A quadratic cost function to minimize

$$c(x_t, u_t) = (x_t - x_{ref})^T Q (x_t - x_{ref}) + u_t^T R u_t$$

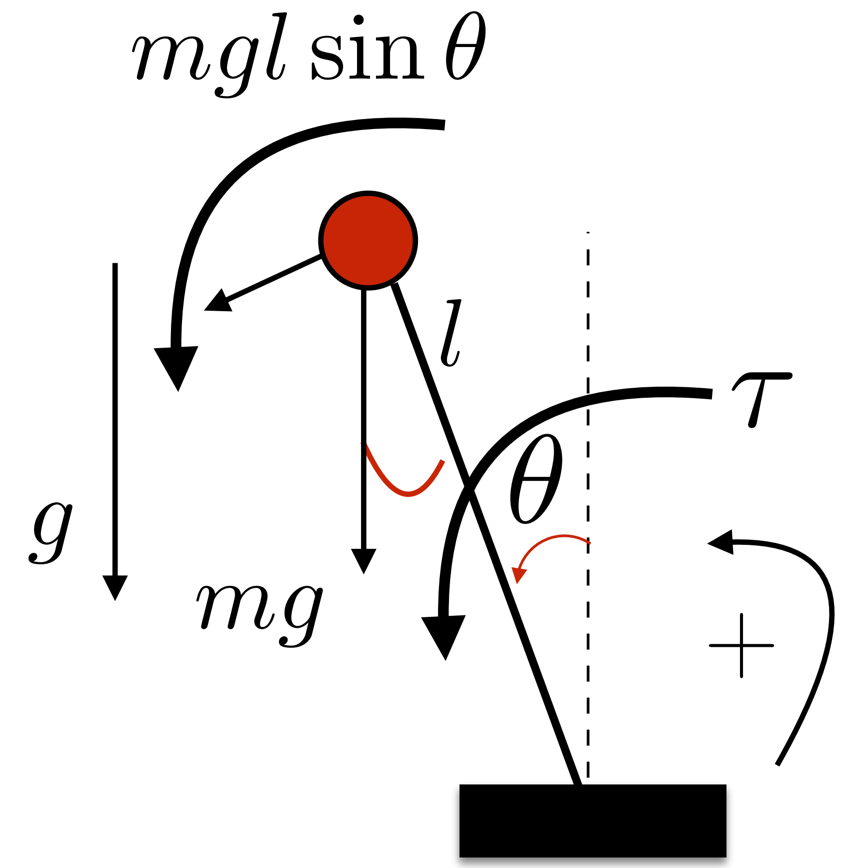
$$= x_t^T Q x_t + u_t^T R u_t \quad , Q, R \succ 0^*$$

Goal: Compute control actions to minimize cumulative cost

$$J = \sum_{t=0}^{T-1} c(x_t, u_t)$$

$^* X \succ 0 \leftrightarrow z^T X z > 0, \ \forall z \neq 0$

Example: Inverted Pendulum



Equations of motion

$$\sum M = J\ddot{\theta}$$

$$mgl\sin\theta + \tau = ml^2\ddot{\theta}$$

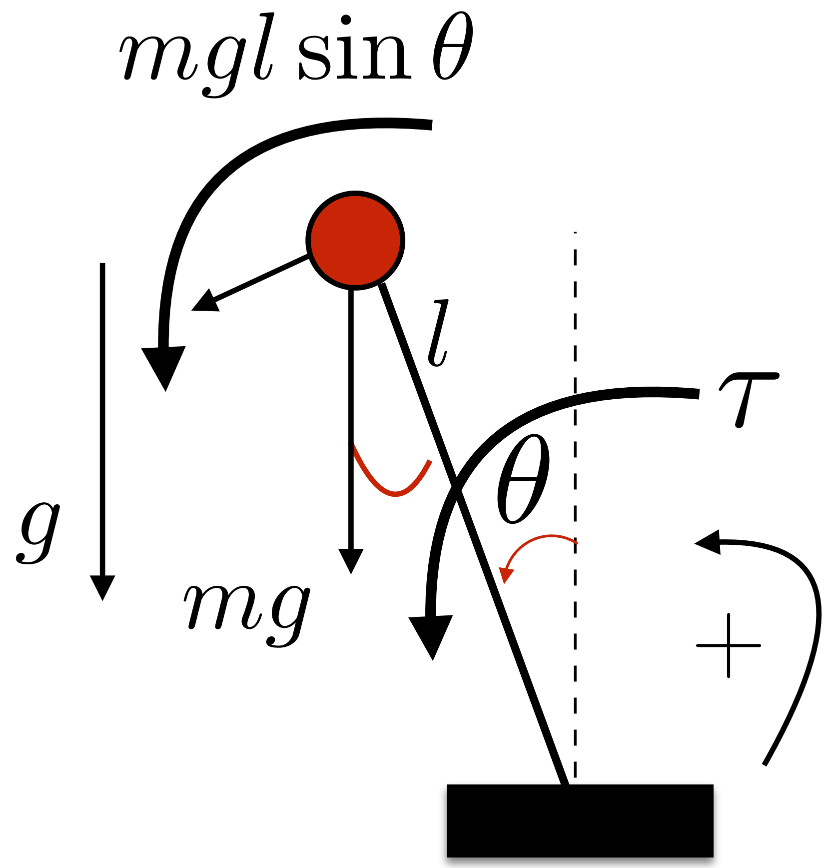
$$\ddot{\theta} = \frac{g}{l}\sin\theta + \frac{1}{ml^2}\tau$$

Linearization $\approx \frac{g}{l}\theta + \frac{1}{ml^2}\tau$

$$\begin{bmatrix}\dot{\theta} \\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & 1 \\ \frac{g}{l} & 0\end{bmatrix}\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0 \\ \frac{1}{ml^2}\end{bmatrix}\tau \quad \text{(Continuous time)}$$

State deriv $\quad A \quad$ State $\quad B$

Example: Inverted Pendulum



Equations of motion

$$\sum M = J\ddot{\theta}$$

$$mgl\sin\theta + \tau = ml^2\ddot{\theta}$$

$$\ddot{\theta} = \frac{g}{l}\sin\theta + \frac{1}{ml^2}\tau$$

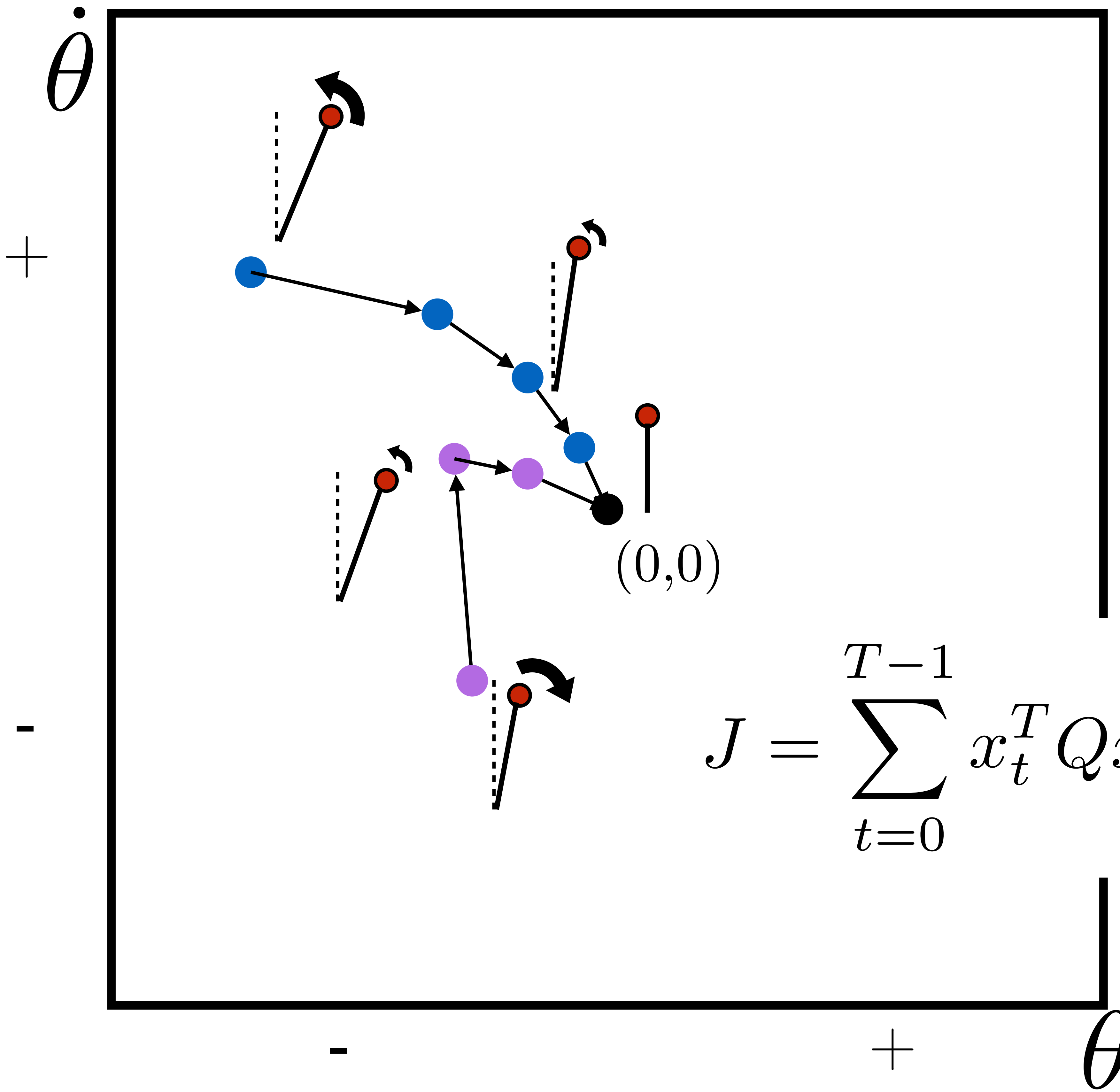
Linearization $\approx \frac{g}{l}\theta + \frac{1}{ml^2}\tau$

$$\begin{bmatrix}\theta_{t+1} \\ \dot{\theta}_{t+1}\end{bmatrix} = \begin{bmatrix}1 & \Delta t \\ \frac{g}{l}\Delta t & 1\end{bmatrix}\begin{bmatrix}\theta_t \\ \dot{\theta}_t\end{bmatrix} + \begin{bmatrix}0 \\ \frac{\Delta t}{ml^2}\end{bmatrix}\tau$$

(discrete time Euler approx)

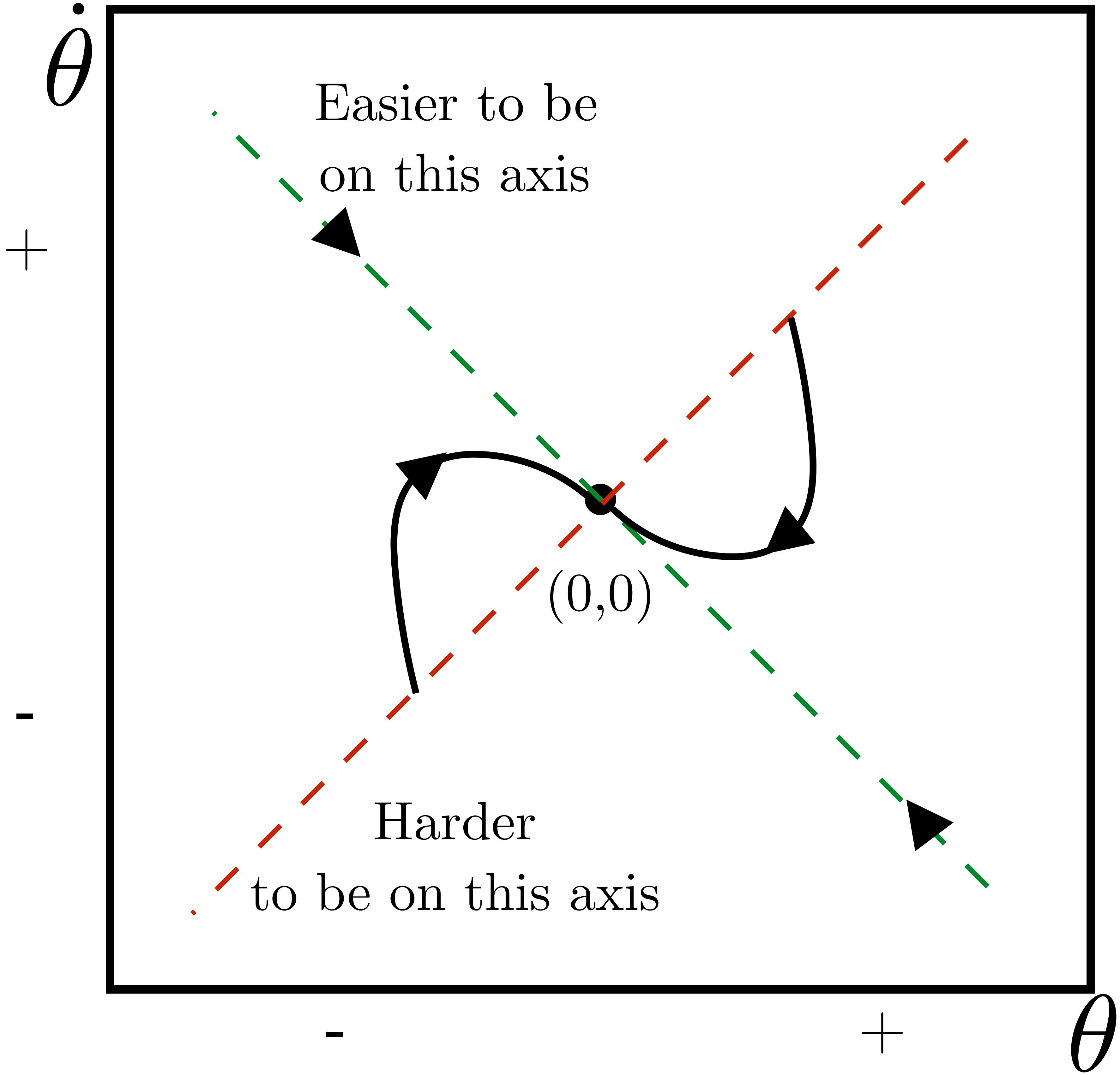
State deriv $\quad A \quad$ State $\quad B$

Get to (0,0) while minimizing cost



$$J = \sum_{t=0}^{T-1} x_t^T Q x_t + u_t^T R u_t$$

Observation: Cost-to-go is not uniform



How do we solve for controls?

Dynamic programming to the rescue!

- efficient, recursive method to solve LQR least-squares problem
- cost is O(Nn3)

Bellman (Value) function (minimum cost to go starting from x_t)

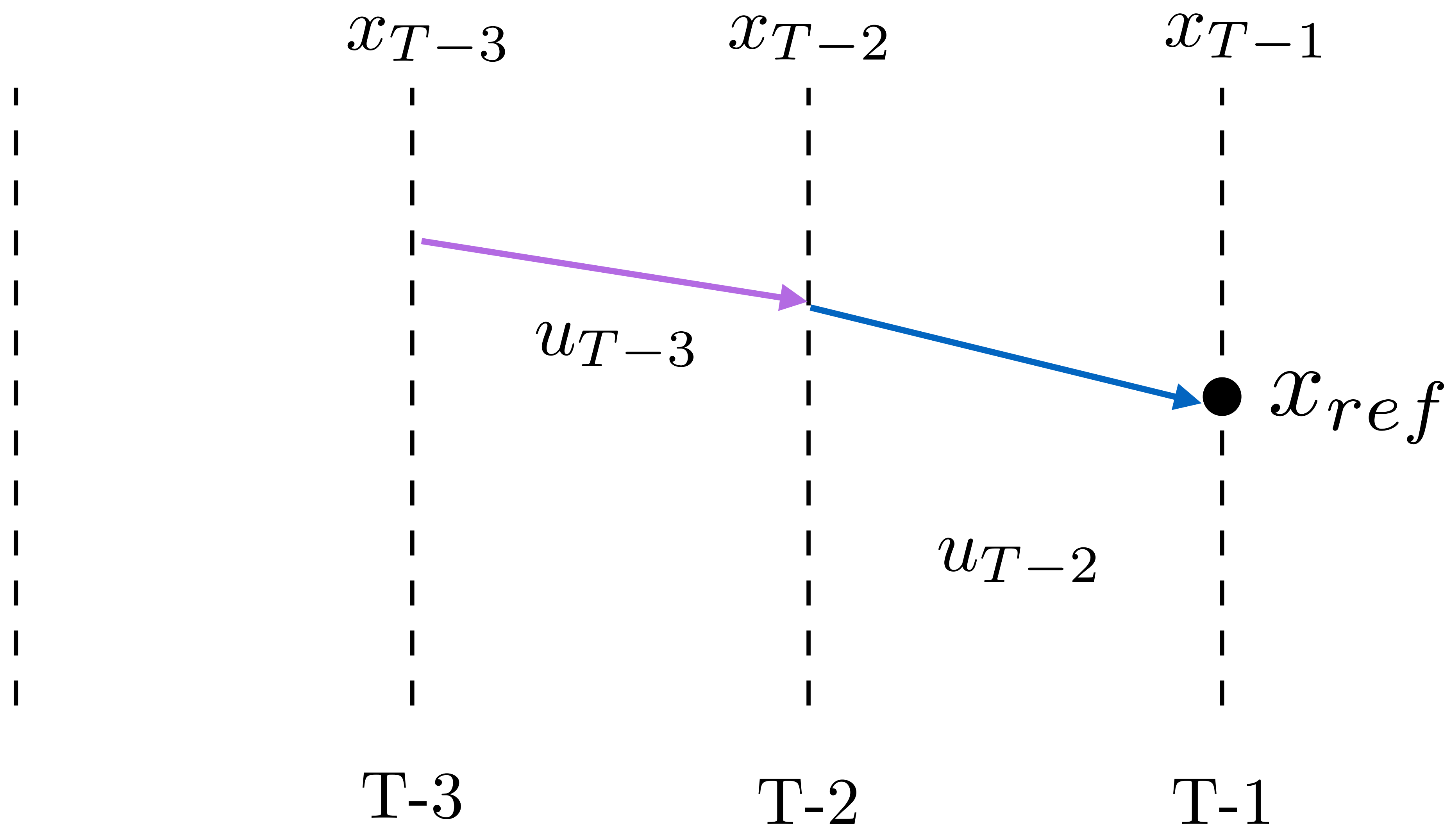
$$J^*(x_t) = \min_{u_t} c(x_t, u_t) + J^*(x_{t+1})$$

where

$$J = \sum_{t=0}^{T-1} c(x_t, u_t)$$

$$c(x_t, u_t) = x^T Q x + u^T R u$$

Solve backwards from final state



Last time step T-1

We have only 1 term in the cost function

$$J^*(x_{T-1}) = \min_{u_T} \; x_{T-1}^T Q x_{T-1} + u_{T-1}^T R u_{T-1}$$

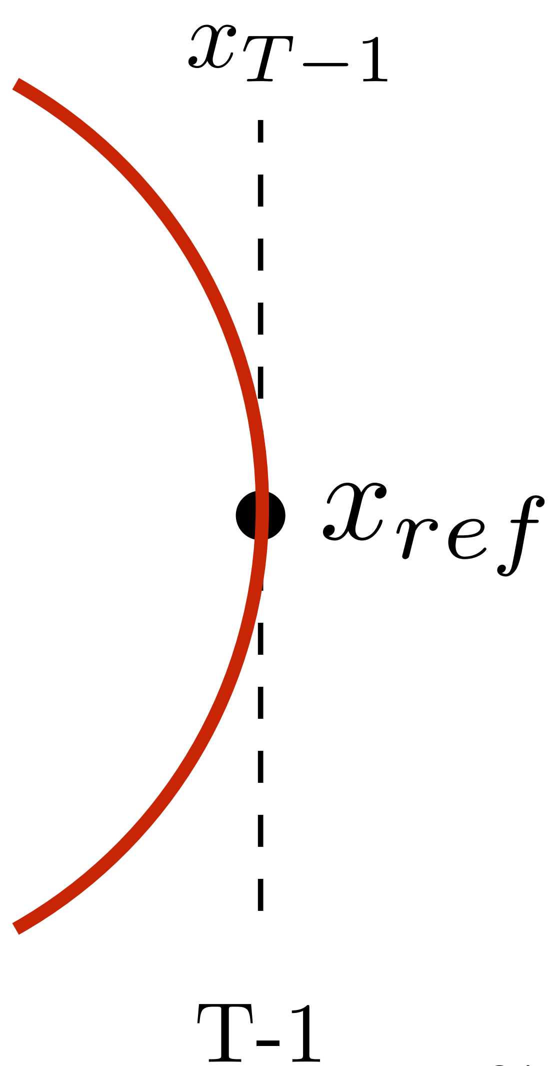
To minimize cost, set control to 0

$$u_{T-1} = 0$$

The cost function is a quadratic

$$\begin{aligned} J^*(x_{T-1}) &= x_{T-1}^T Q x_{T-1} \\ &= x_{T-1}^T \boxed{V_{T-1}} x_{T-1} \end{aligned}$$

(Value matrix)



Previous time step T-2

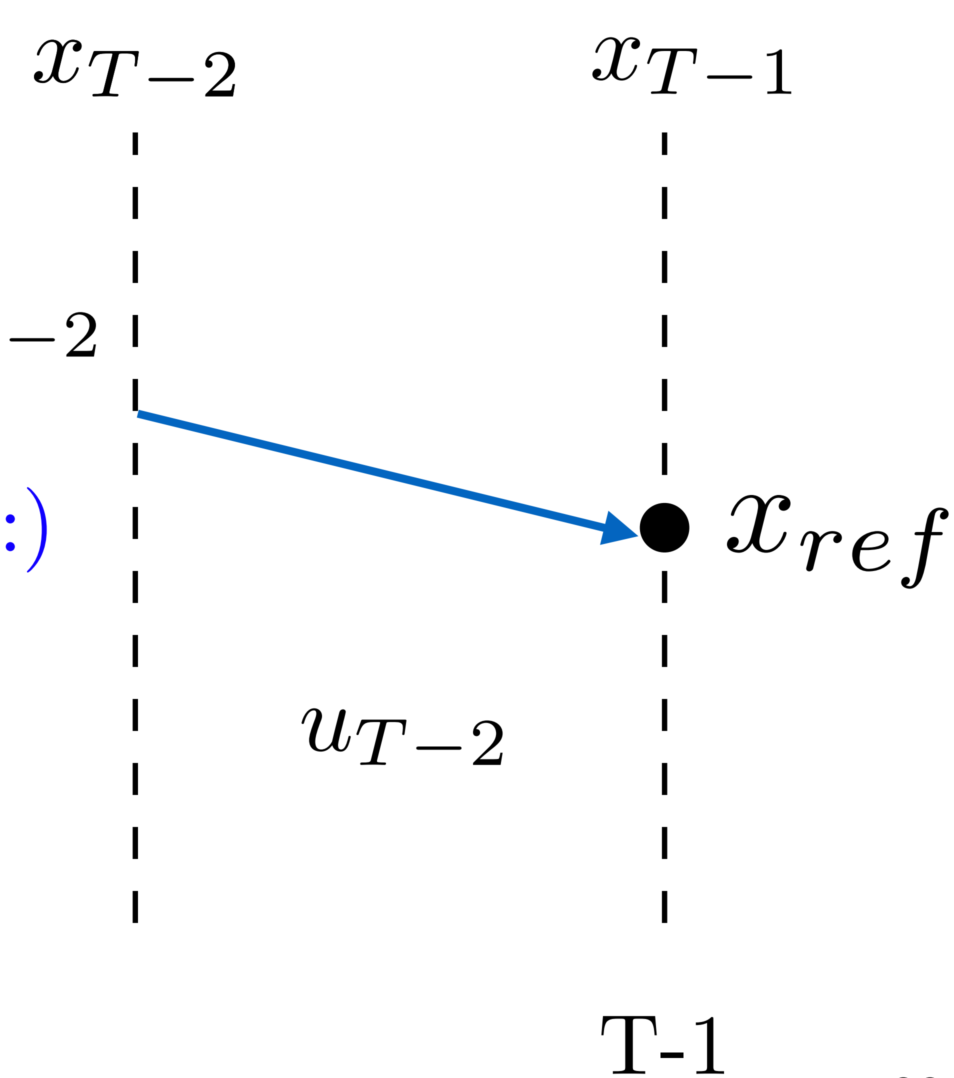
$$J^*(x_{T-2}) = \min_{u_{T-2}} c(x_{T-2}, u_{T-2}) + J^*(x_{T-1})$$

$$= \min_{u_{T-2}} x_{T-2}^T Q x_{T-2} + u_{T-2}^T R u_{T-2} + x_{T-1}^T V_{T-1} x_{T-1}$$

Solve for control at timestep T-2
(set derivative wrt u_{T-2} to 0)

$$u_{T-2} = -\underbrace{(R + B^T V_{T-1} B)^{-1} B^T V_{T-1} A}_{K_{T-2}} x_{T-2}$$

K_{T-2} Kalman Gain :)

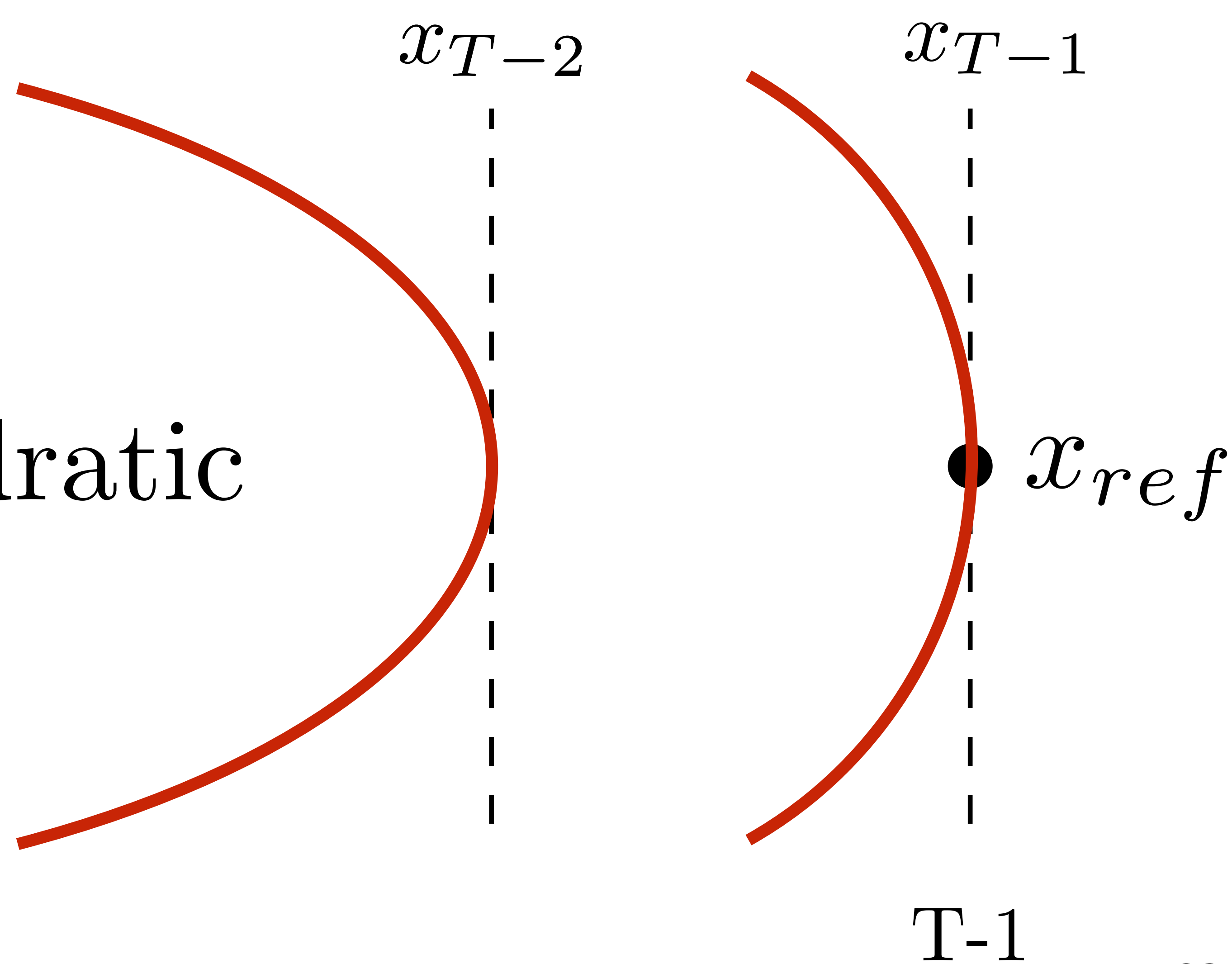


Observation: Control law is linear!

Plug control into Value Function

$$J^*(x_{T-2}) = x_{T-2}^T \underbrace{(Q + K_{T-2}^T R K_{T-2} + (A + BK_{T-2})^T V_{T-1}(A + BK_{T-2}))}_{V_{T-2}} x_{T-2}$$

Value function is quadratic



We can derive this relation at ALL time steps

$$K_t = -(R + B^T V_{t+1} B)^{-1} B^T V_{t+1} A$$

$$V_t = Q + K_t^T R K_t + (A + B K_t)^T V_{t+1} (A + B K_t)$$

Current cost	Action cost	Closed loop dynamics	Future value matrix	Closed loop dynamics

The LQR algorithm

Algorithm `OptimalValue`(A, B, Q, R, t, T)

  if $t = T - 1$ **then**

    return Q

  end

  else

    $V_{t+1} =$ `OptimalValue`$(A, B, Q, R, t+1, T)$

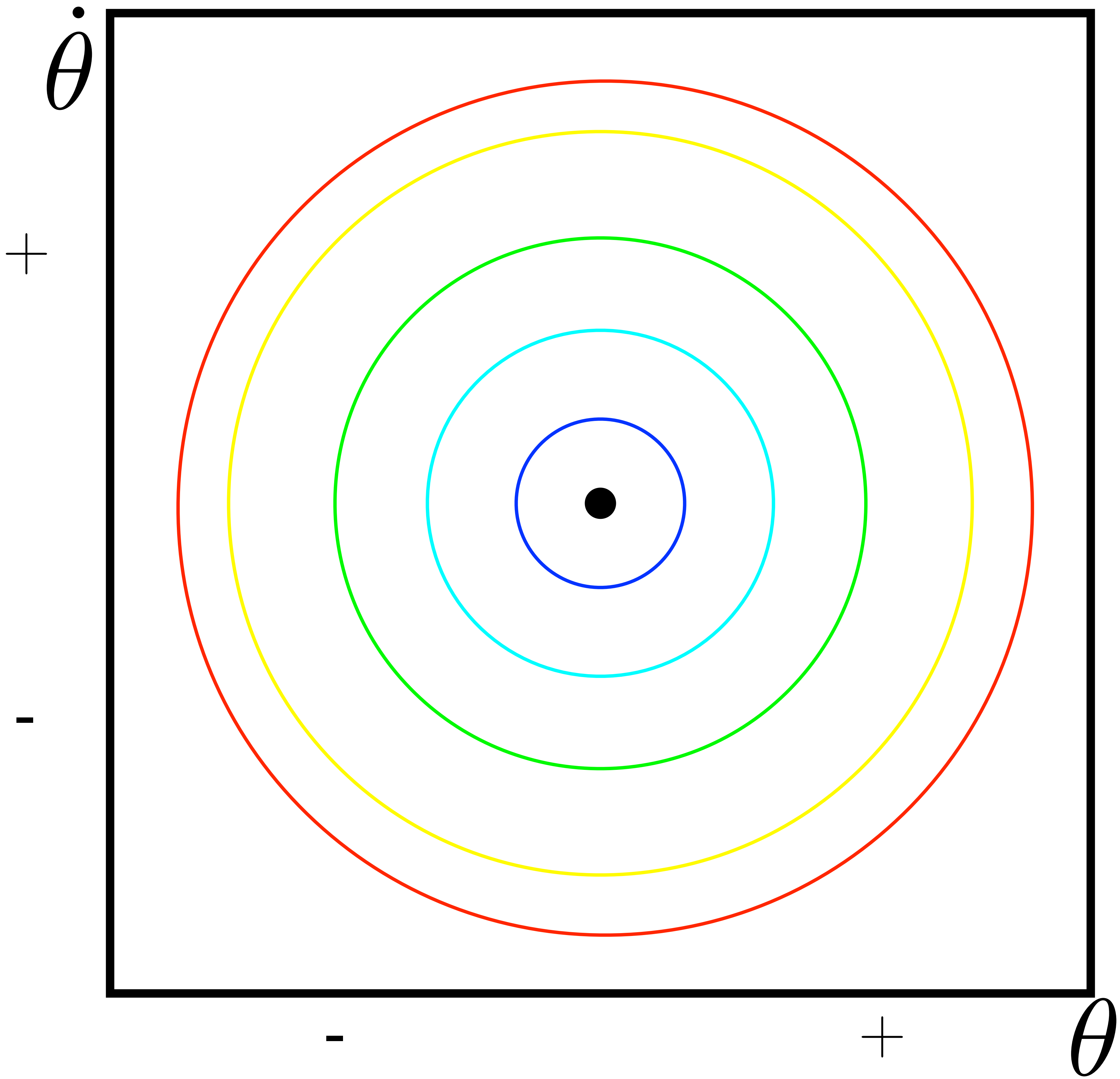
    $K_t = -(B^T V_{t+1} B + R)^{-1} B^T V_{t+1} A$

    return $V_t = Q + K_t^T R K_t + (A + BK_t)^T V_{t+1} (A + BK_t)$

  end

(Courtesy Drew Bagnell)

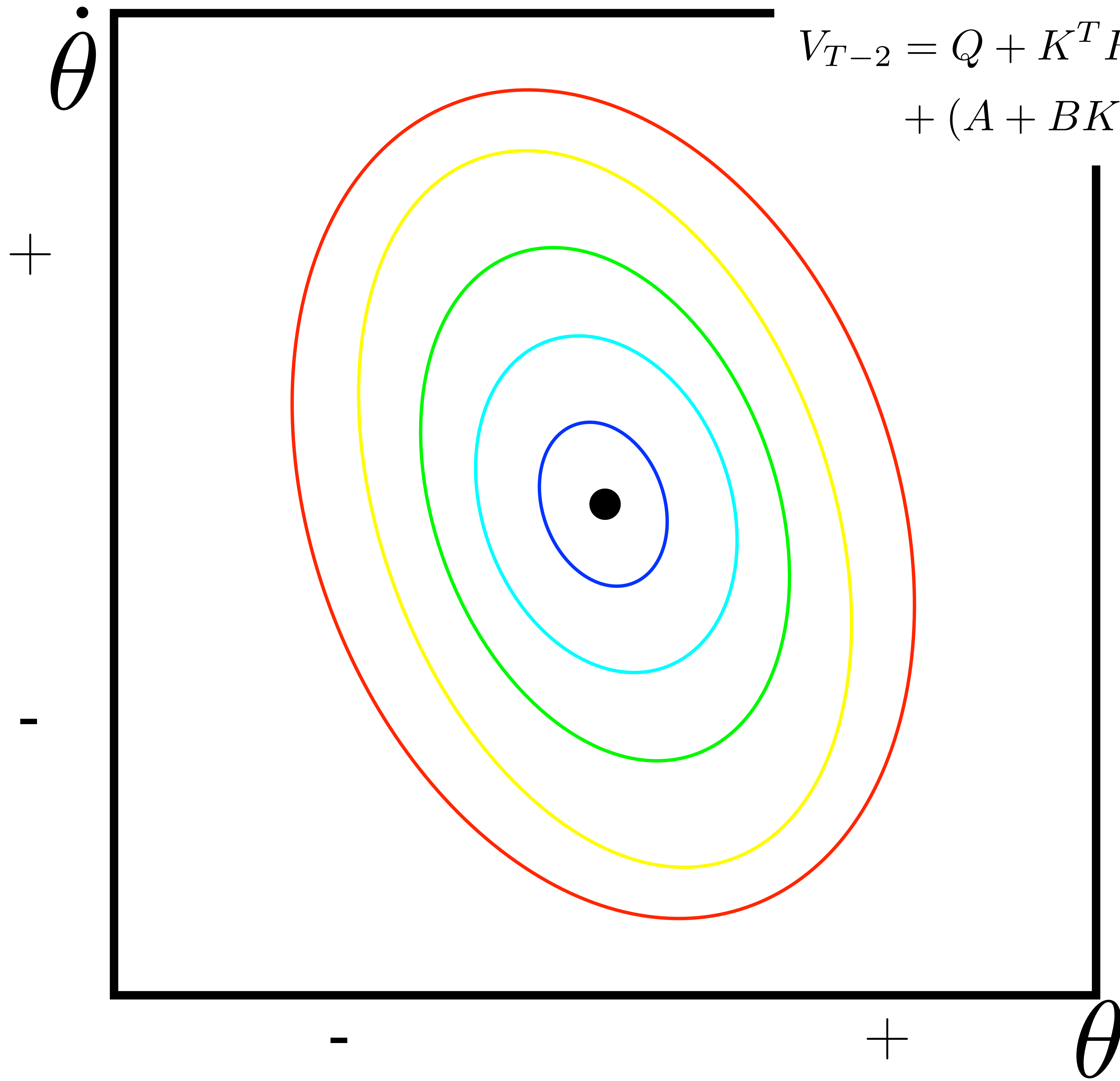
Contours of value function (T-1)



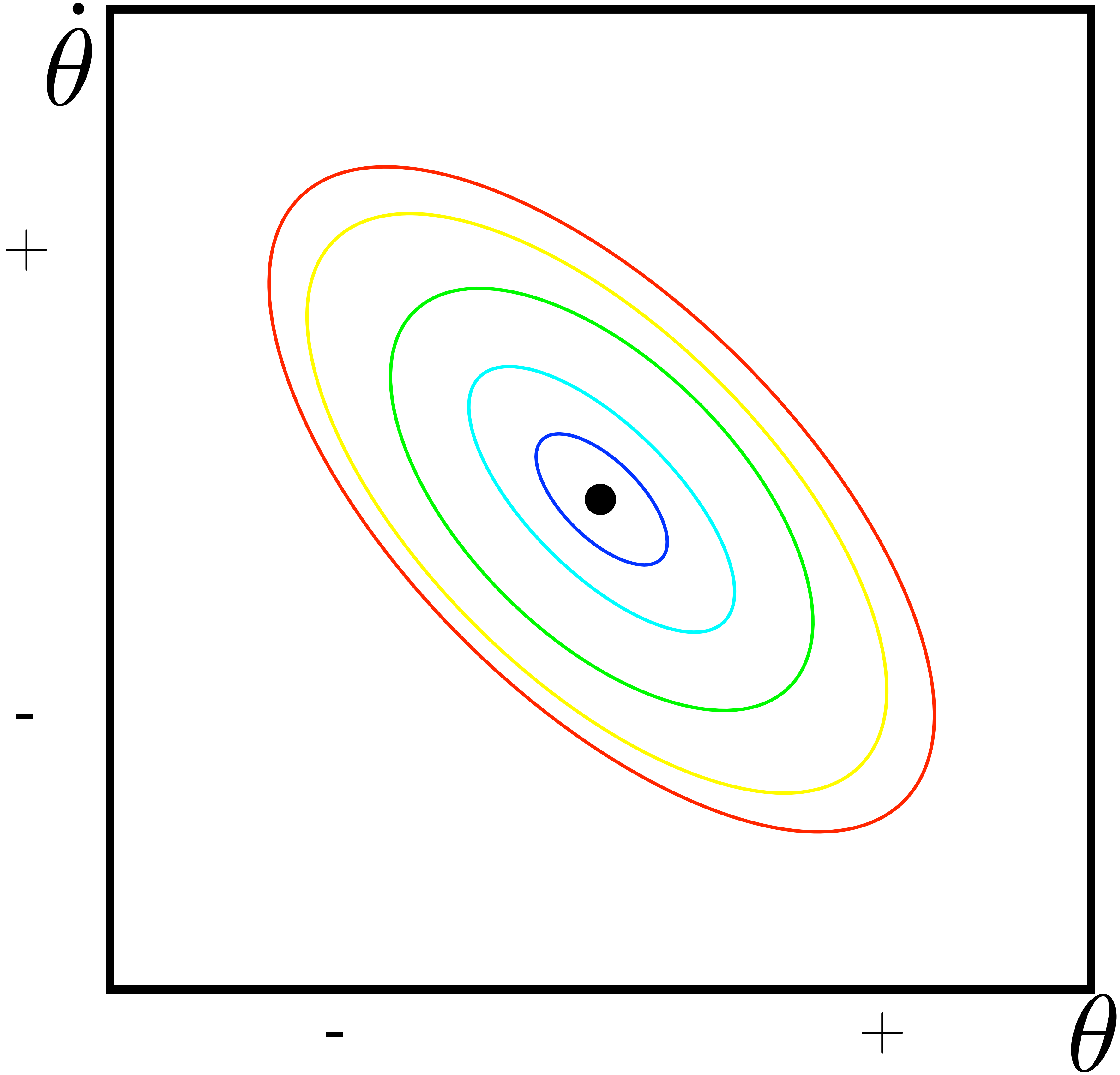
$V_{T-1} = Q$

Contours of value function (T-2)

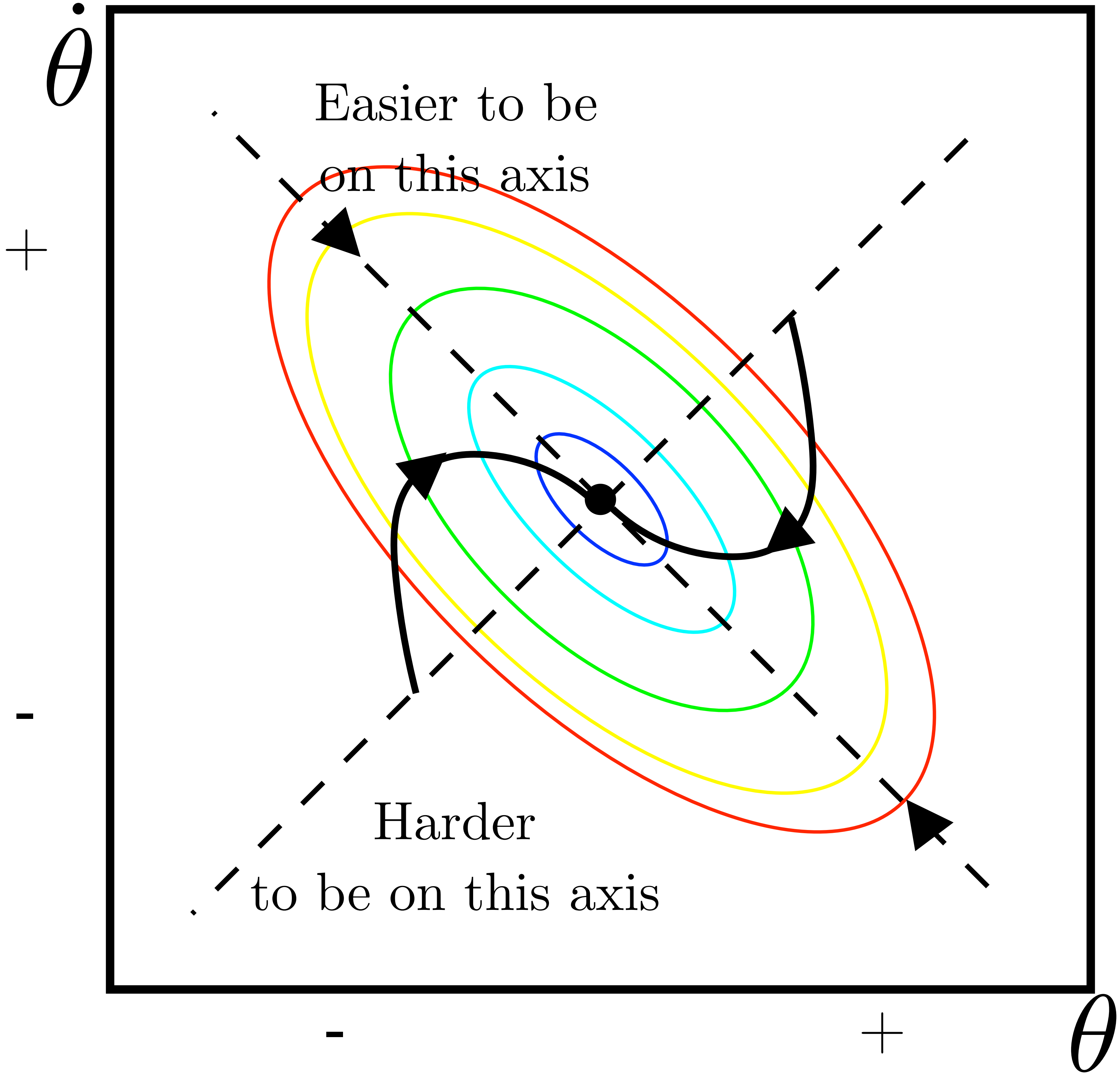
$$V_{T-2} = Q + K^T R K + (A + BK)^T V_{T-1} (A + BK)$$



Contours of value function (many steps)



How does the value function evolve?



What if my time horizon is very very very large?

Convergence of value iteration

Theorem: If the system is stabilizable, then the value V will converge

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

$$K = -(R + B^T V B)^{-1} B^T V A$$

Discrete Algebraic Ricatti Equation (DARE)

How do I solve? Can iterate over V / use eigen value decomposition [1]

Type into MATLAB: `dare(A,B,Q,R)`

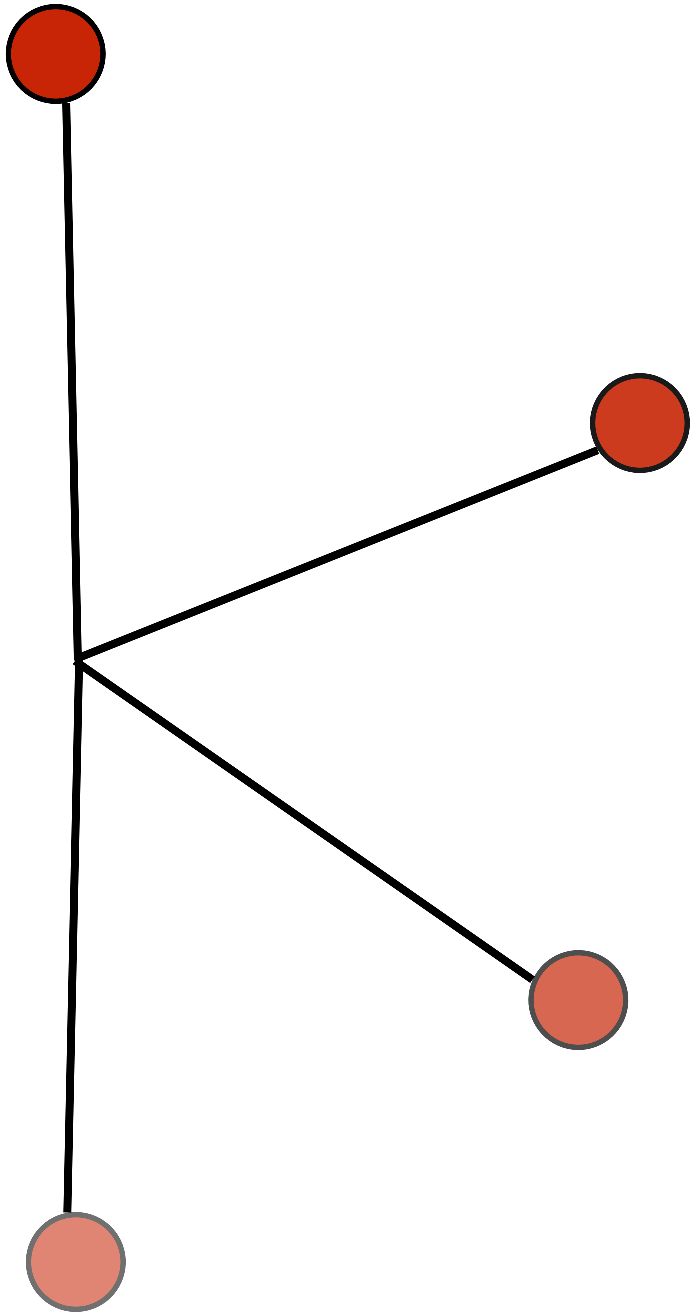
[1] Arnold, W.F., III and A.J. Laub, "Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations," *Proc. IEEE®*, 72 (1984), pp. 1746-1754.

So, can this controller stabilize inverted pendulum for all angles?

No!

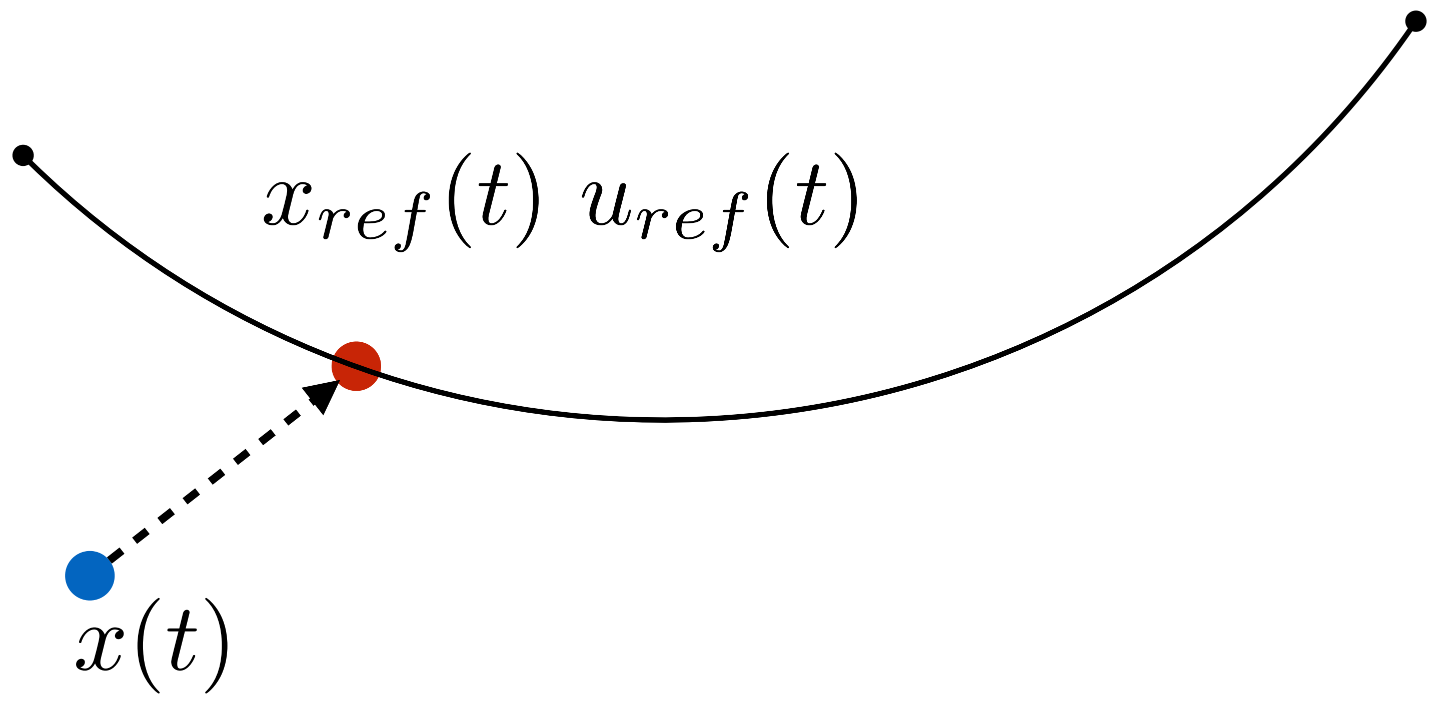
Linearization error is too large when angle is large

Instead, can we use LQR to track reference trajectory?



Yes

But but we need to linearize about nominal trajectory



LQR for Time-Varying Dynamical Systems

$$x_{t+1} = A_t x_t + B_t u_t$$

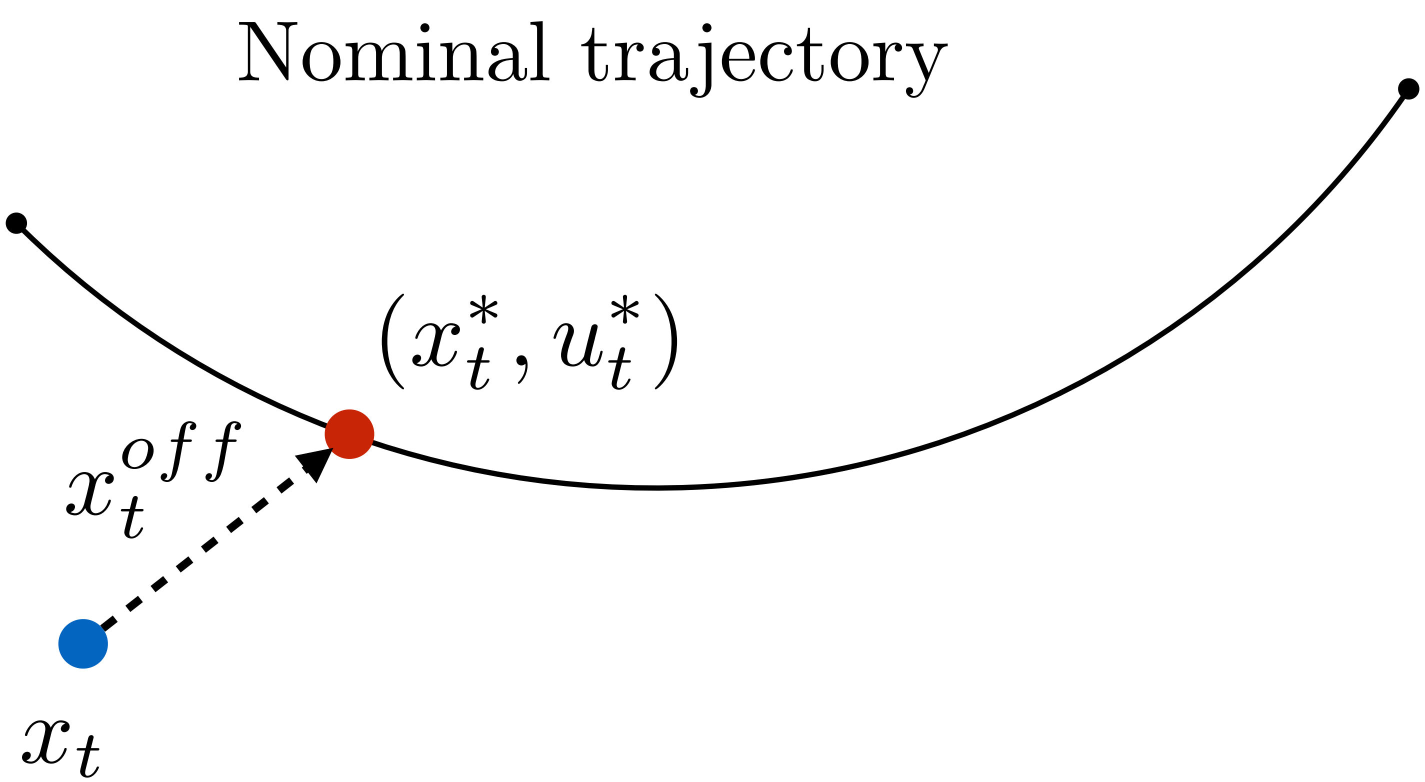
$$c(x_t, u_t) = x_t^T Q_t x_t + u_t^T R_t u_t$$

Straight forward to get LQR equations

$$K_t = -(R_t + B_t^T V_{t+1} B_t)^{-1} B_t^T V_{t+1} A_t$$

$$V_t = Q_t + K_t^T R_t K_t + (A_t + B_t K_t)^T V_{t+1} (A_t + B_t K_t)$$

Linearize about trajectory



$$\dot{x} = f(x, u)$$

$$A_t = \left.\frac{\partial f}{\partial x}\right|_{x_t^*}$$

$$B_t = \left.\frac{\partial f}{\partial u}\right|_{u_t^*}$$

$$x_{t+1} = A_t x_t + B_t u_t + x_t^{off}$$

Trick to write in Linear System Form

$$x_{t+1} = A_t x_t + B_t u_t + x_t^{off}$$

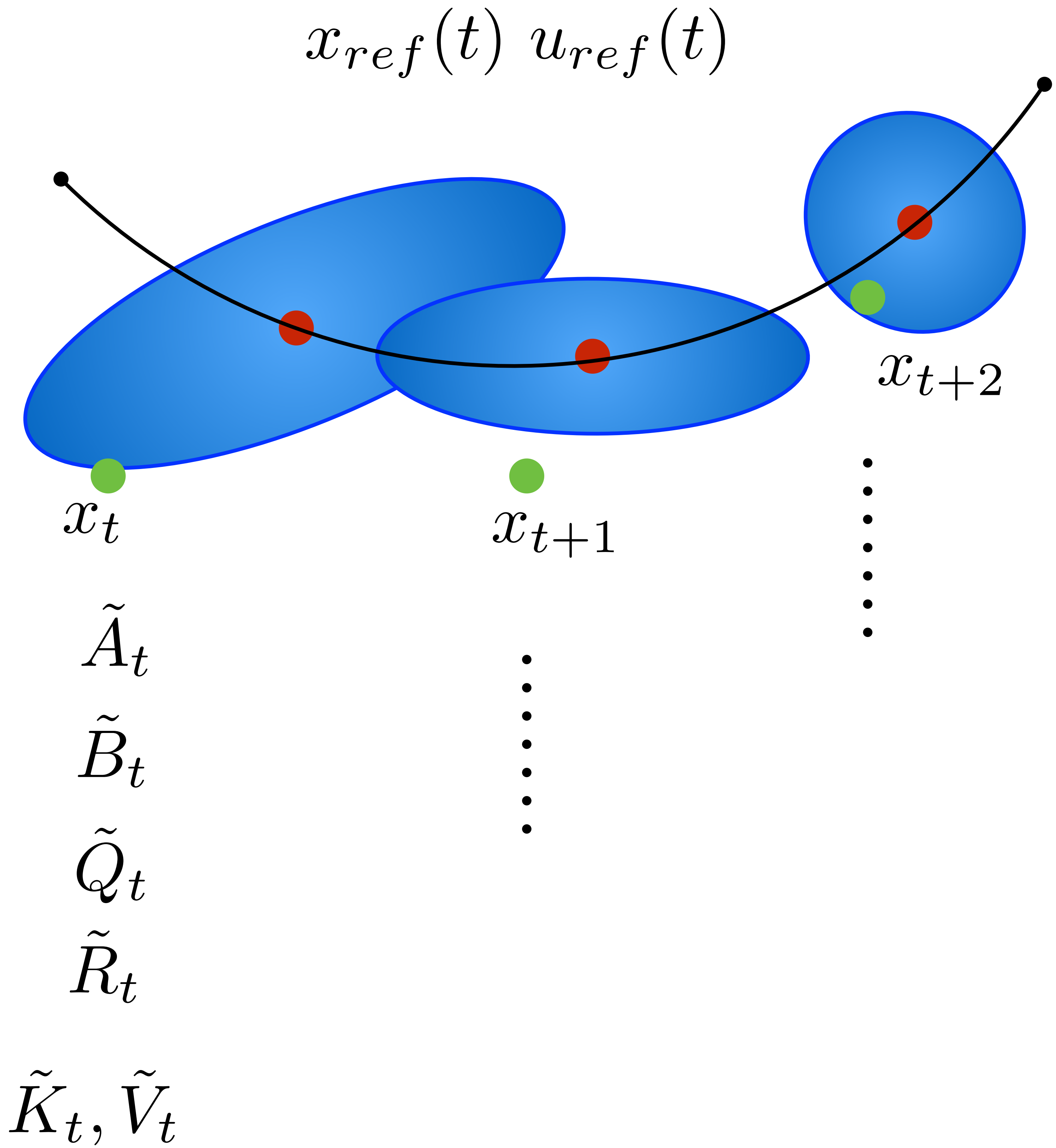
Homogeneous coordinates $\tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}$

$$\tilde{x}_{t+1} = \begin{pmatrix} A_t & x_t^{off} \\ 0 & 1 \end{pmatrix} x_t + \begin{pmatrix} B_t \\ 0 \end{pmatrix} u_t$$

Similarly you can transform cost function

$$c(\tilde{x}_t, u_t) = \tilde{x}_t^T \tilde{Q}_t \tilde{x}_t + u_t^T R_t u_t$$

Shape of the value function changes along trajectory



Questions

1. Can we solve LQR for continuous time dynamics?

   Yes! Refer to Continuous Algebraic Ricatti Equations (CARE)

2. Can LQR handle arbitrary costs (not just tracking)?

   Yes! We will talk about iterative LQR next class

3. What if I want to penalize control derivatives?

   No problem! Add control as part of state space

4. Can we handle noisy dynamics?

   Yes! Gaussian noise does not change the answer

Trivia: Duality between control and estimation

R. Kalman "A new approach to linear filtering and prediction problems." (1960)

linear-quadratic regulator	**Kalman-Bucy filter**	
V	Σ	(state variance)
A	A^T	(dynamics)
B	H^T	(measurement)
R	DD^T	(dynamics noise)
Q	CC^T	(motion noise)
t	$t_f - t$	

(Table from E.Todorov "General duality between optimal control and estimation", CDC, 2008)