GPOPS-II

And Trajectory Optimization

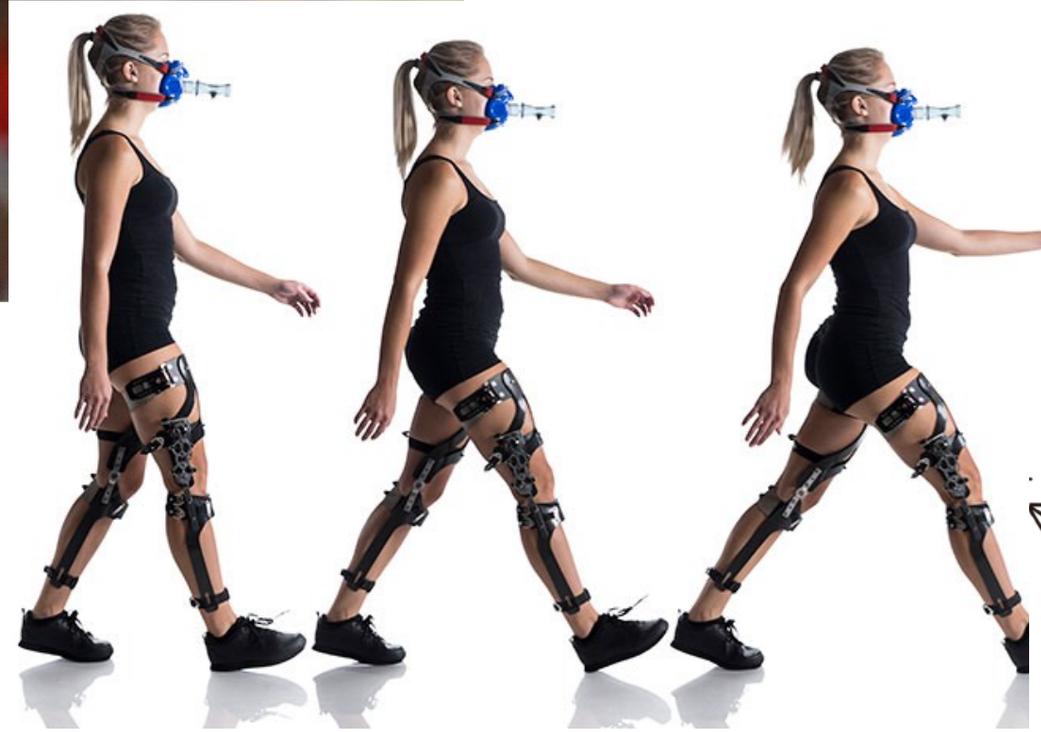
by Delyle Polet

Presented at Neuromechanics, Nov. 21 2018

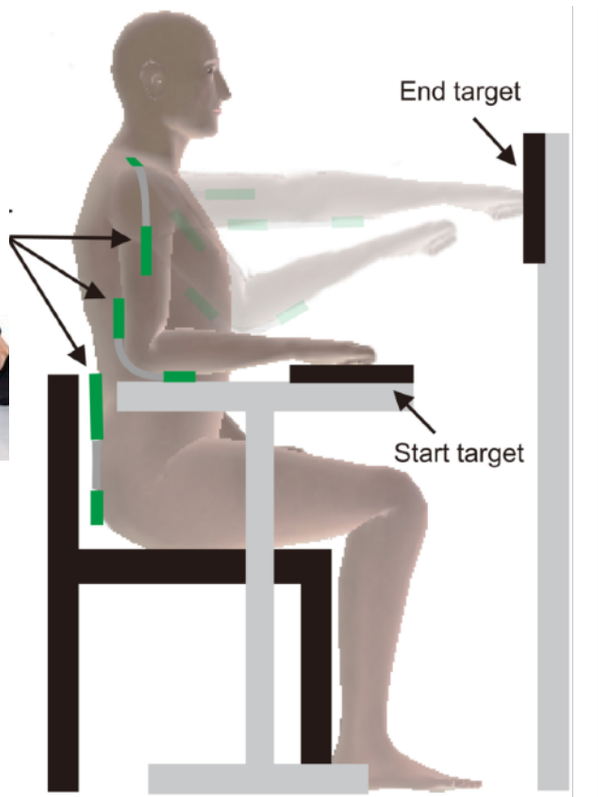
Why optimization?

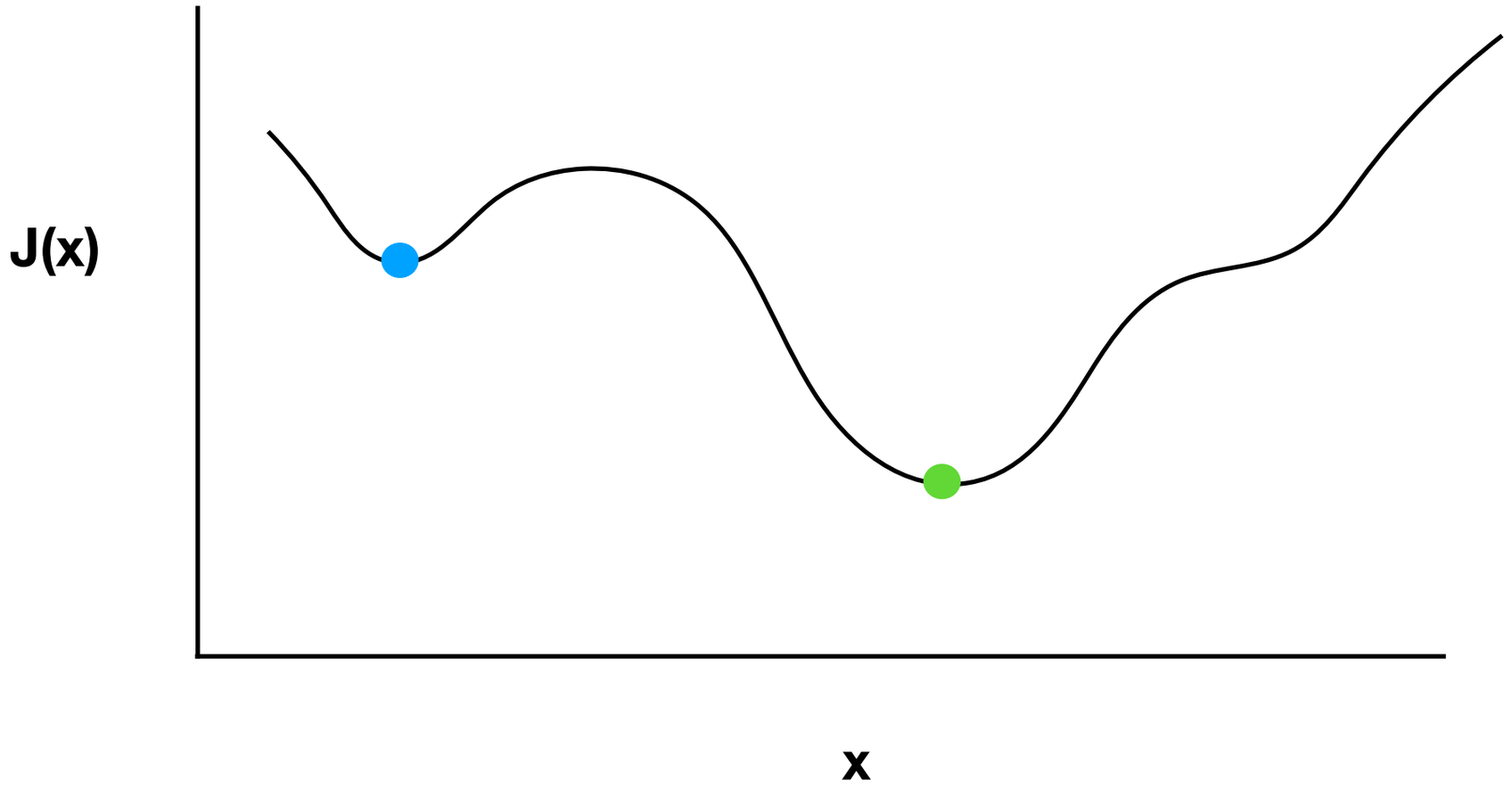
Maximize Athletic performance

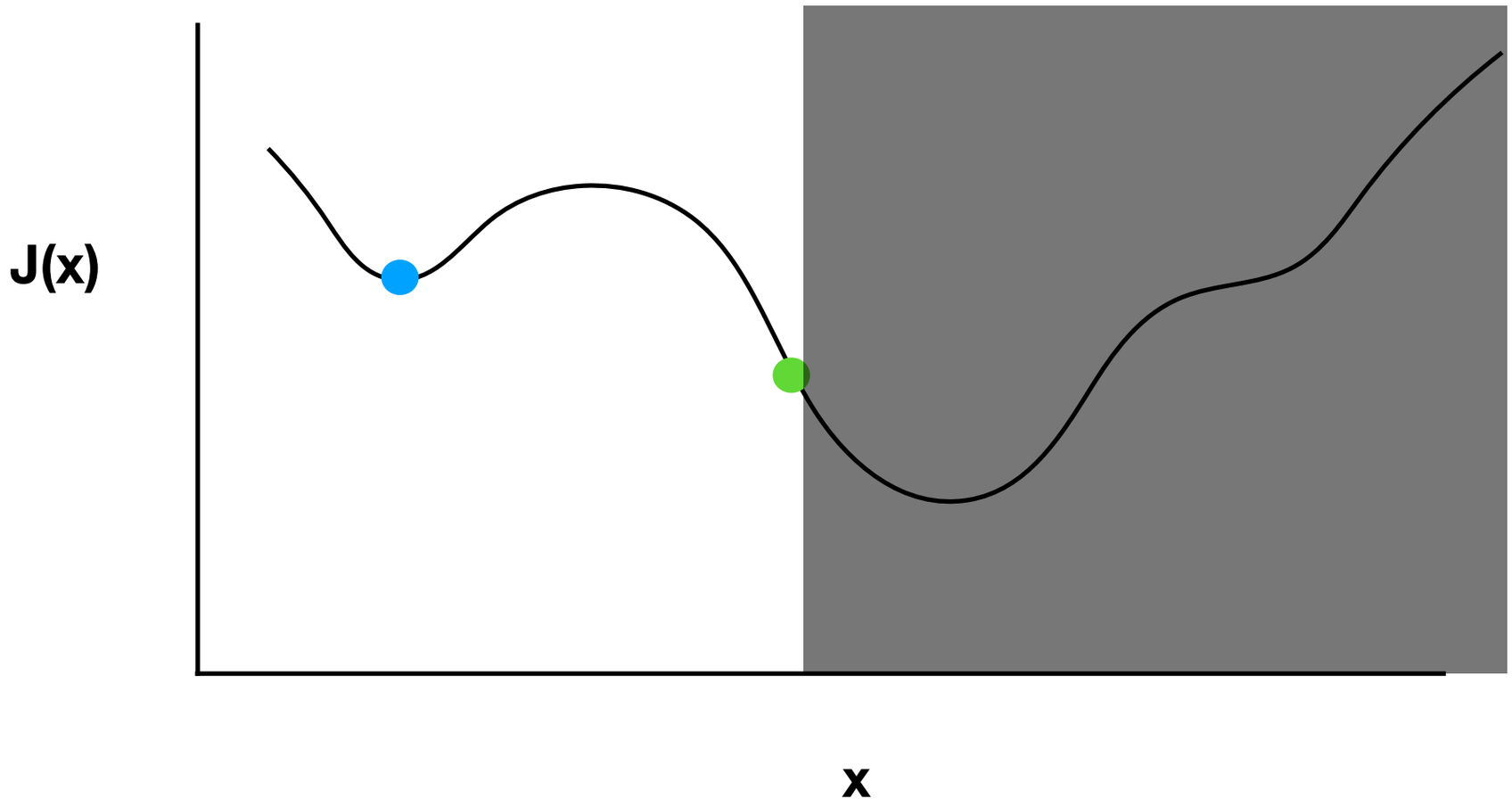
Minimize Cost

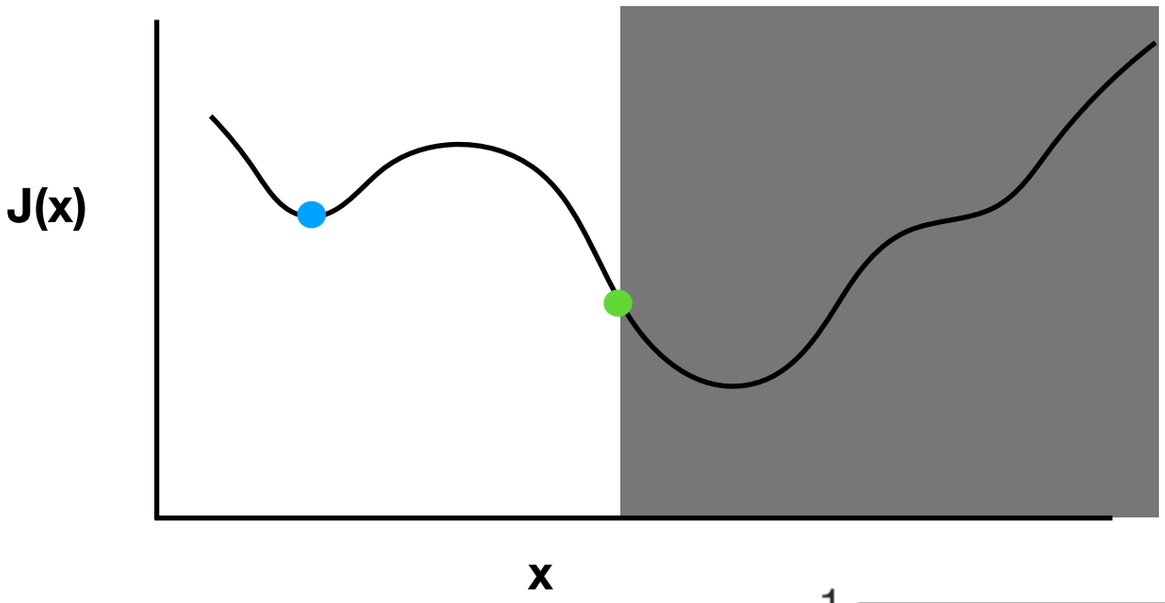


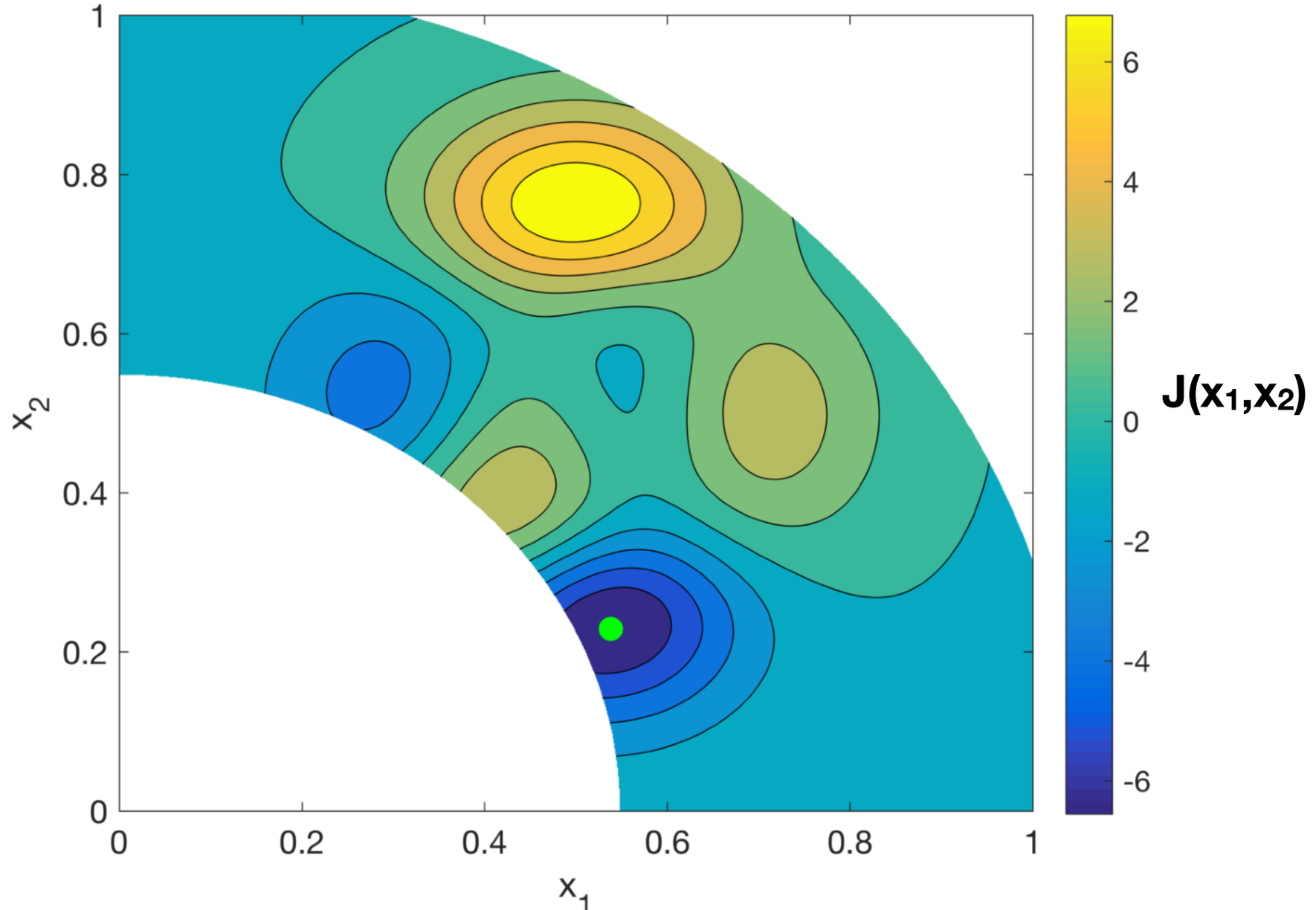
Fit a model

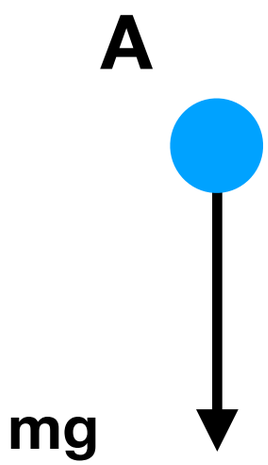



J(x)
x

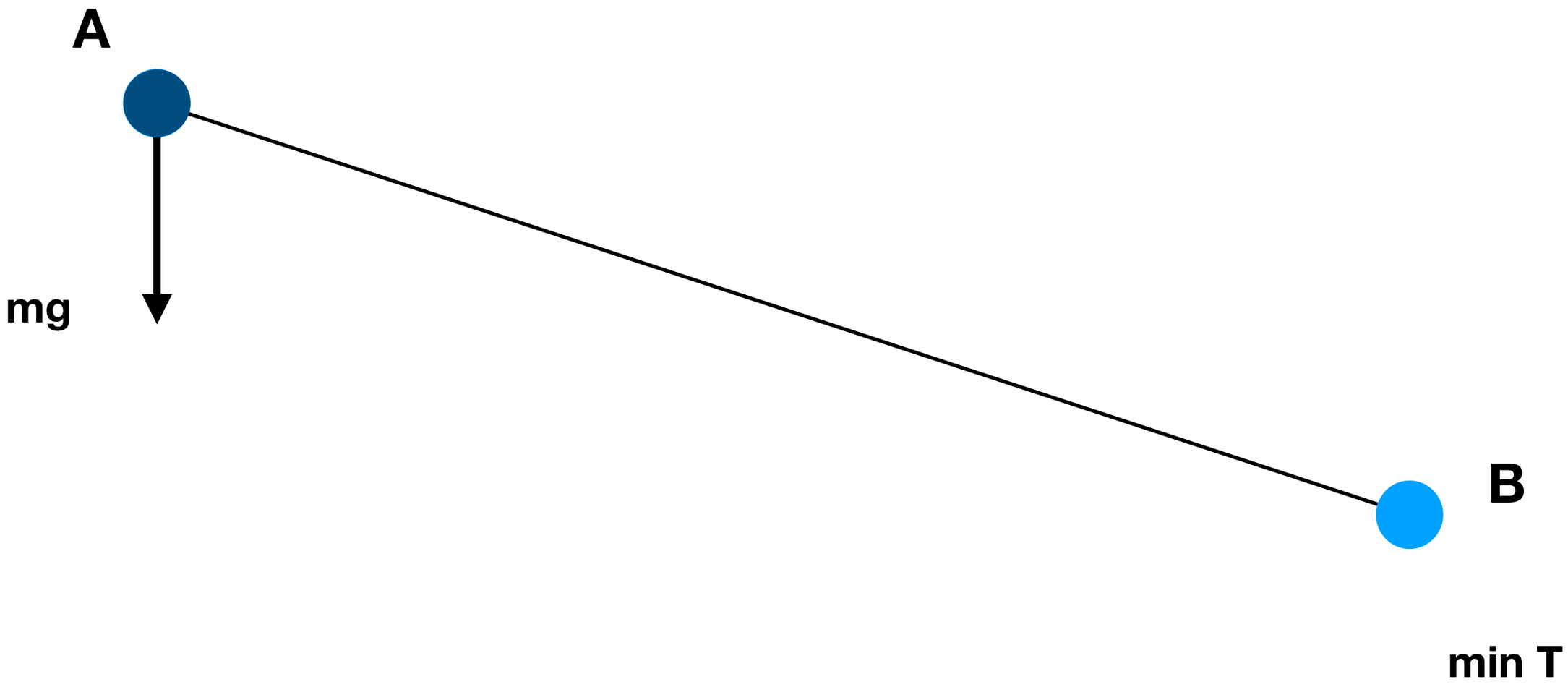

J(x)
x

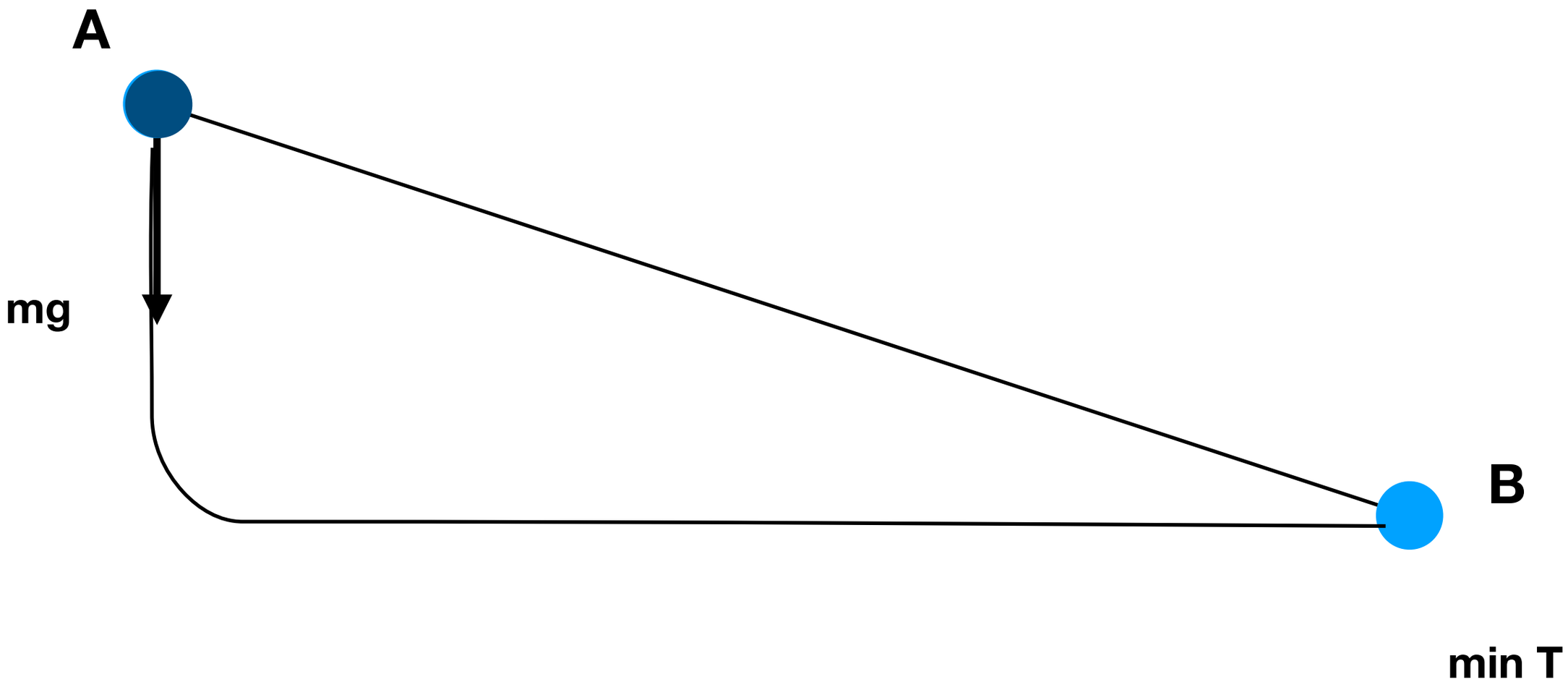

J(x)
x

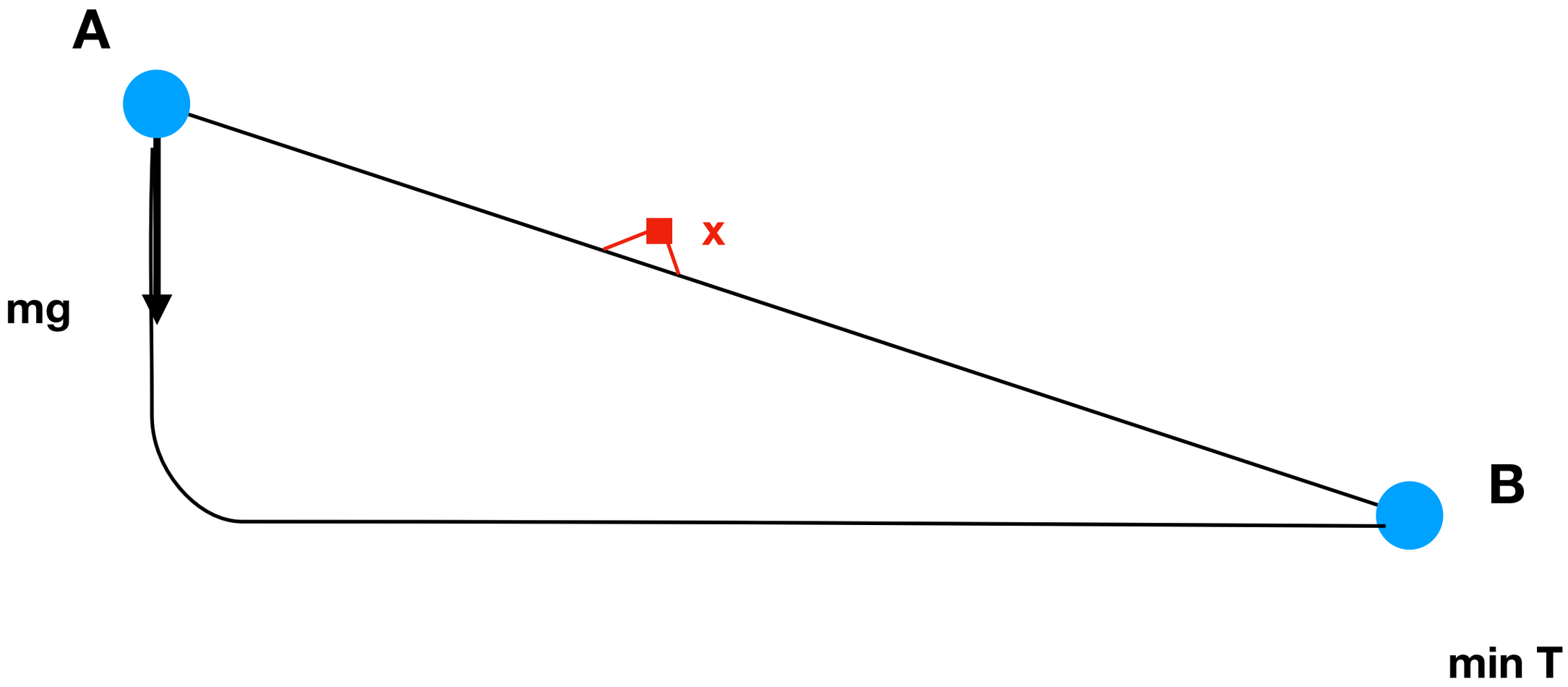

1
0.8
0.6
0.4
0.2
0
x_2
0
0.2
0.4
0.6
0.8
1
x_1
6
4
2
0
-2
-4
-6
$J(x_1,x_2)$

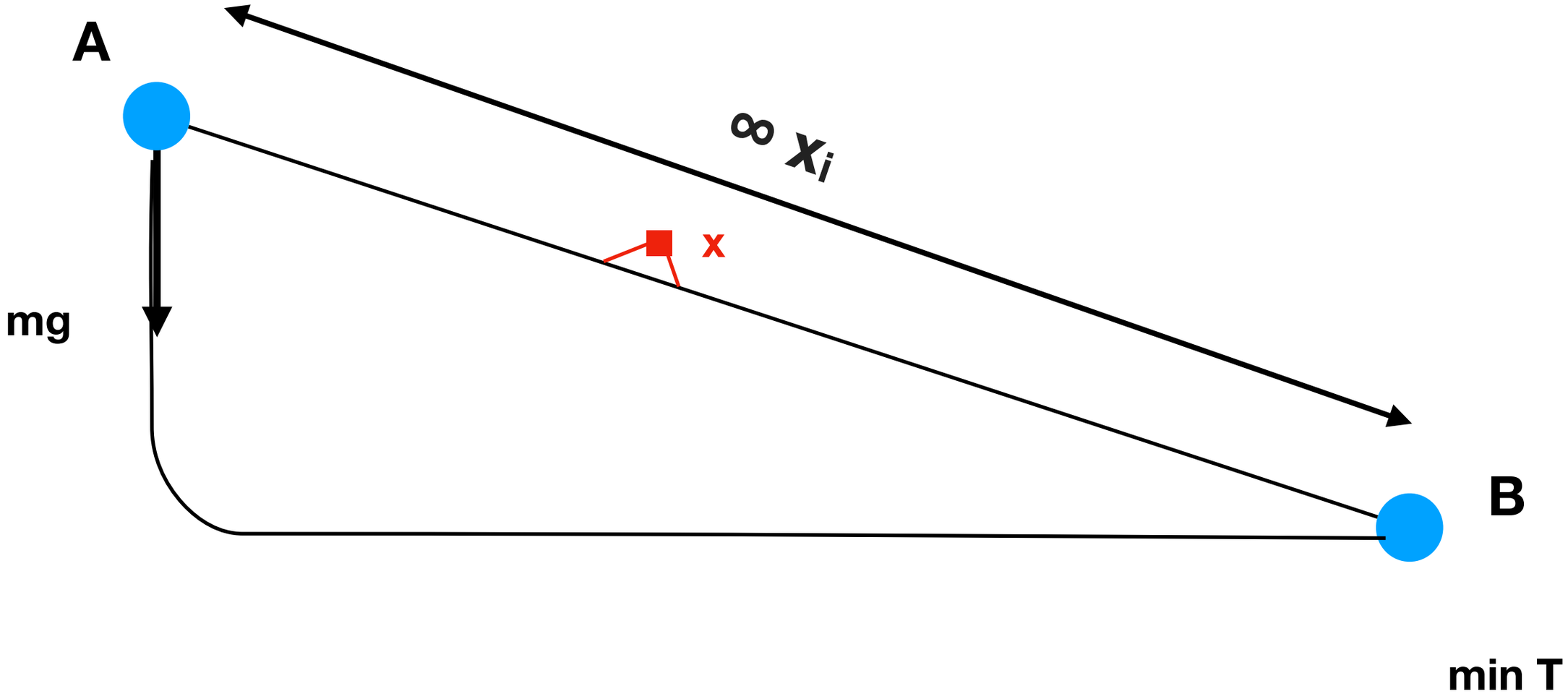


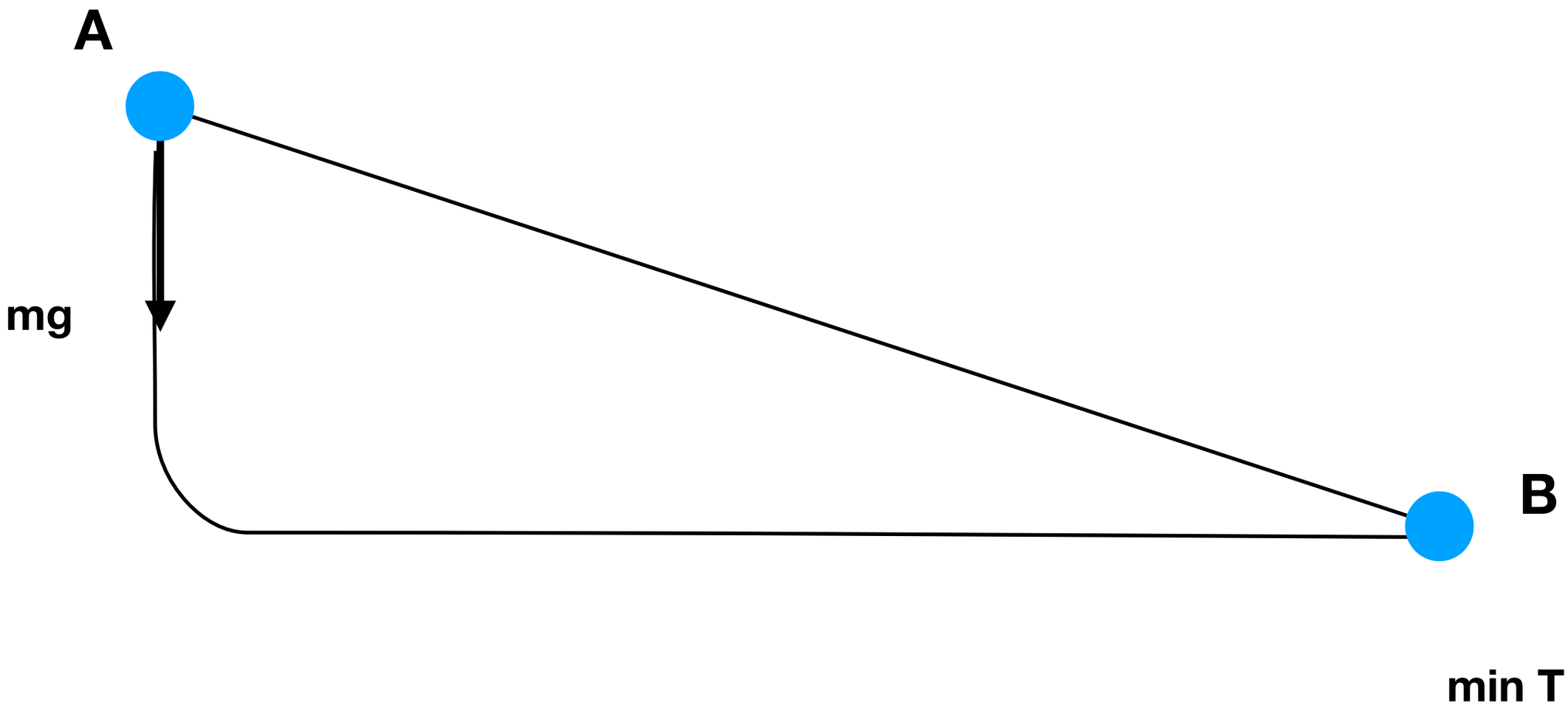
Classic Brachistochrone Problem

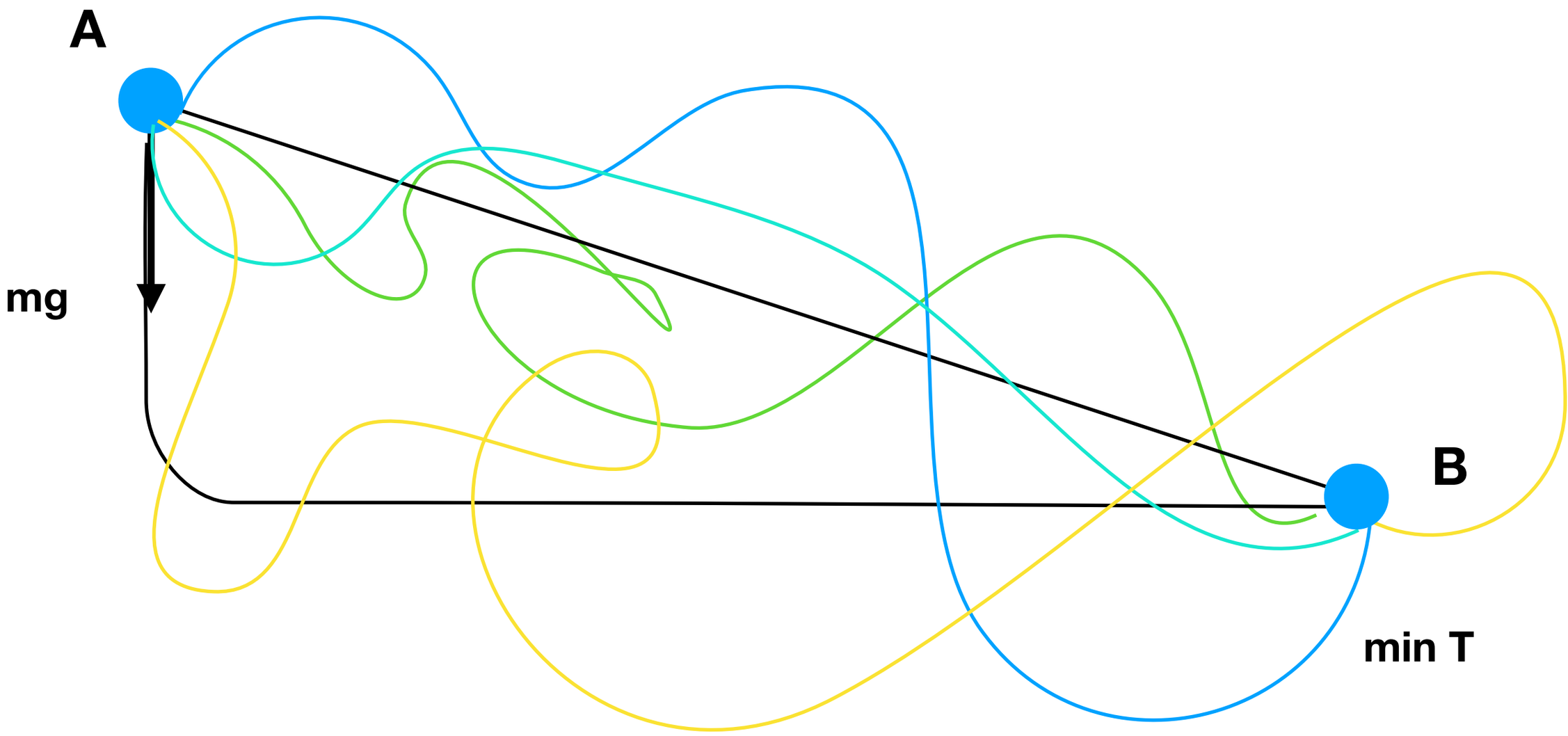

A
mg
B
min T

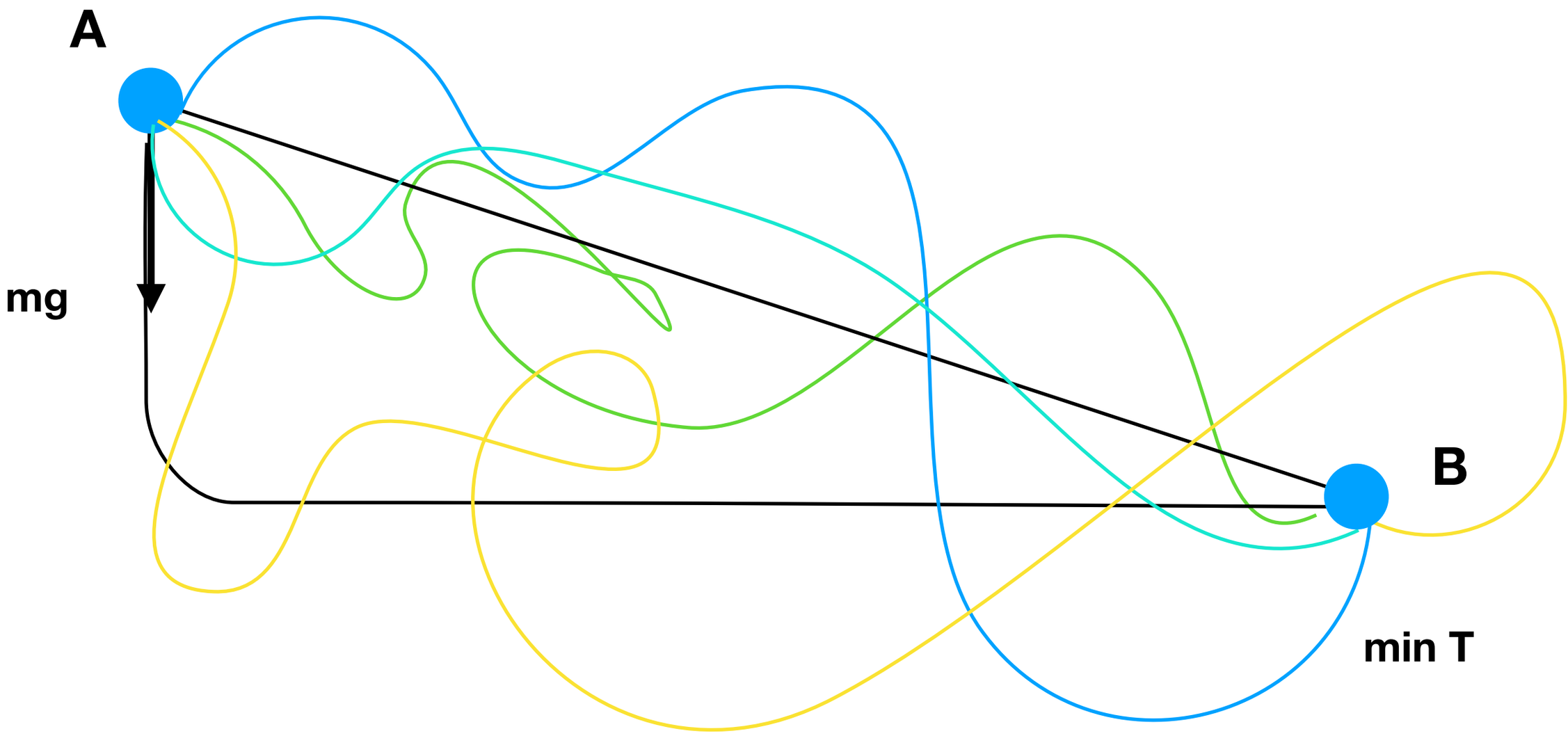

A
mg
B
min T


A
mg
x
B
min T


A
∞ x_i
x
mg
B
min T

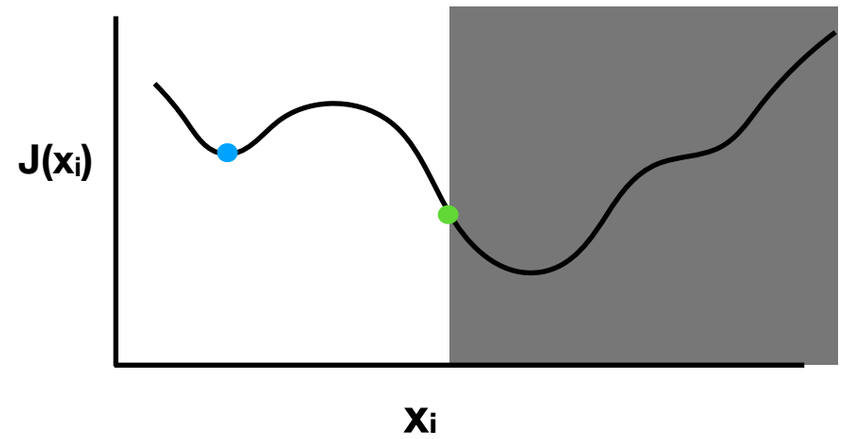

A
mg
B
min T


A
mg
B
min T



Infinitely many trajectories;
An infinite dimensional problem!

Trajectory Optimization



A

1. Discretize (transcription)

2. Gradient evaluation and descent

xi

mg

B

min T

3. Mesh error evaluation (are dynamics ok?)
 -> Return to 1 (new discretization)

Find the **states,** $\vec{X}(t) \in \mathbb{R}^n$

controls $\vec{U}(t) \in \mathbb{R}^m$

and **Parameters** $\vec{P} \in \mathbb{R}^l$

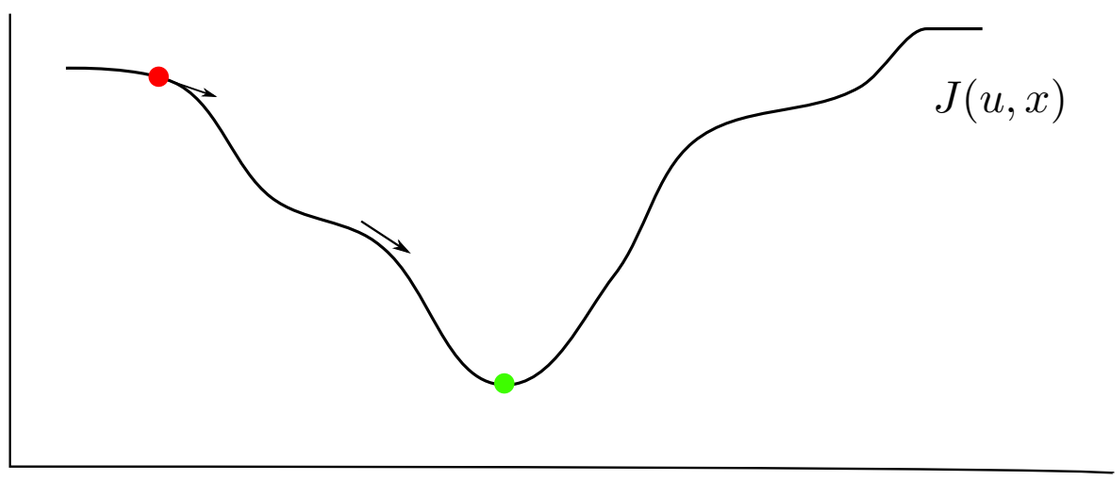
That minimize the **cost** $J(u, x) \in \mathbb{R}$

Subject to **dynamics** $\dot{X}(t) = f(t, x, u) \in \mathbb{R}^n$

and **constraints** $\vec{C}(x(t), u(t)) \leq 0 \in \mathbb{R}^k$

over a **time interval** $t \in [t_i, t_f]$

Gradient of Cost function -> minimum



Transcription Methods

Indirect Methods ⟷ **Direct Methods**

- "optimize then discretize"
- more accurate
- harder to pose and solve

- "discretize the optimize"
- less accurate
- easier to pose and solve

Shooting Methods ⟷ **Collocation Methods**

- based on simulation
- better for problems with simple control and no path constraints

- based on function approximation
- better for problems with complicated control and/or path constraints

---

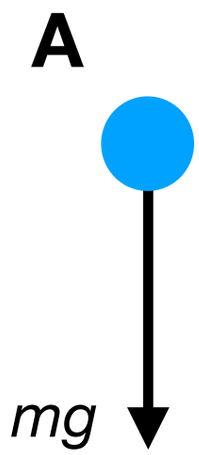
"h-methods" ⟷ **"p-methods"**

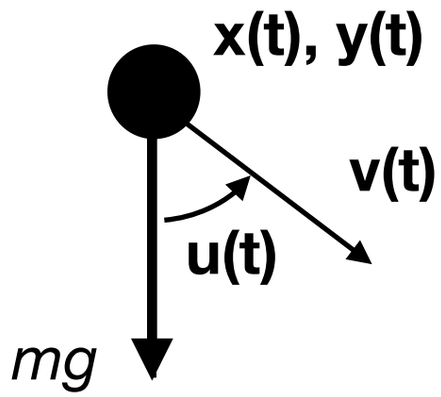
- low-order method
- converge by increasing number of segments

- high-order methods
- converge by increasing method order

^ From Matt Kelly's website

Cost: T
Constraints: Start at A, end at B
States: Time-varying properties of the system that have a temporal derivative
Controls: Time-varying properties of the system that do not have a temporal derivative





min T

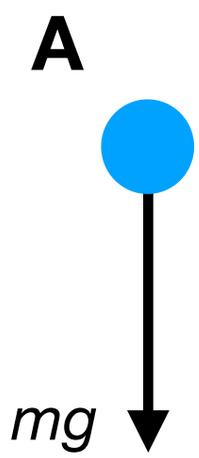
Dynamics:

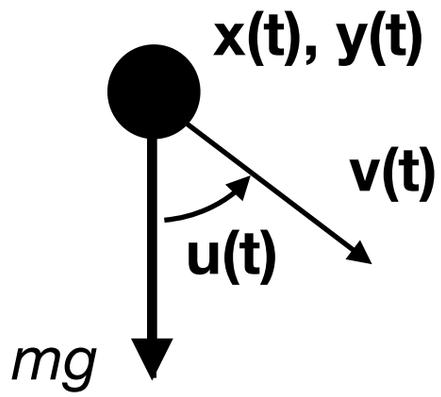
$$\dot{x} = v\sin(u)$$
$$\dot{y} = -v\cos u$$
$$\dot{v} = -g\cos(u)$$

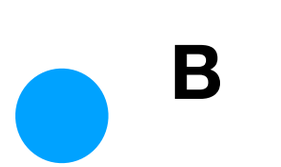
$\Rightarrow$

States are [*x*,*y*,*v*]
Control is u

Cost: T
Constraints: Start at A, end at B
States: [$x(t)$,$y(t)$,$v(t)$]
Controls: $u(t)$
Bounds: Constraints on min/max values of variables / time



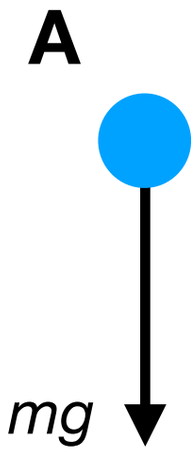


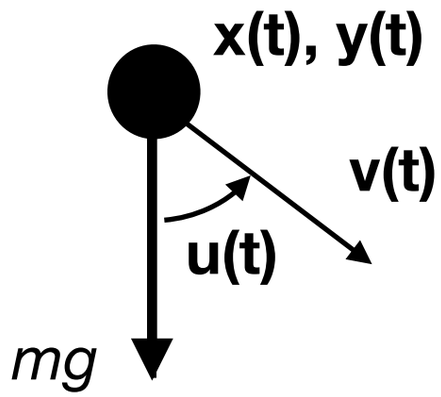


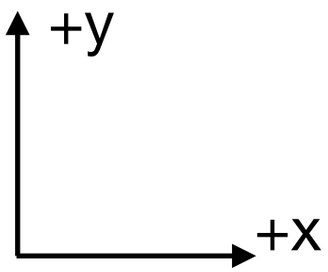
min T

$$
\begin{aligned}
t &\geq 0 \\
[x(0),\, y(0)] &= \text{A} \\
[x(\text{T}),\, y(\text{T})] &= \text{B} \\
x(\text{t}),y(\text{t}),v(\text{t}) &= \text{Unconstrained} \\
? \leq \text{u(t)} &\leq ?
\end{aligned}
$$

Cost: T
Constraints: Start at A, end at B
States: [$x(t)$,$y(t)$,$v(t)$]
Controls: $u(t)$
Bounds: Constraints on min/max values of variables / time







min T

$$t \geq 0$$
$$v(0) = 0$$
$$[x(0), y(0)] = A = [0,0]$$
$$[x(T), y(T)] = B$$
$$x(t),y(t),v(t) = \text{Unconstrained}$$
$$-\pi \leq u(t) \leq \pi$$

Find the **states,** $\vec{X}(t) = [x(t), y(t), v(t)]$

controls $\vec{U}(t) = u(t)$

That minimize the **cost** $J(u, x) = T$

Subject to **dynamics** $\dot{X}(t) = [v \sin u, -v \cos u, \quad g \cos u]$

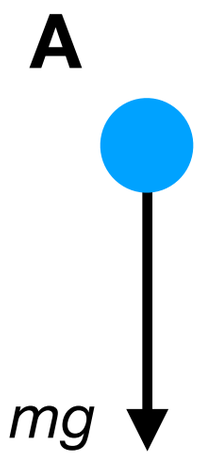
and **constraints** $[x(0), y(0), v(0)] = \vec{0}$

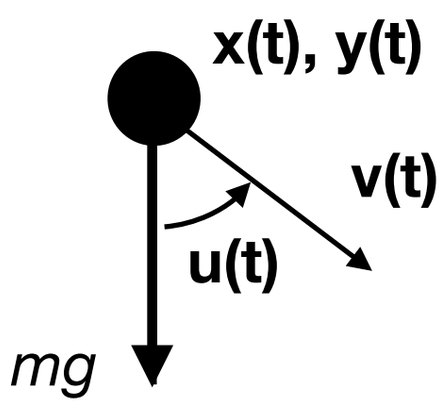
$[x(T), y(T)] = \vec{B}$

$-\pi \leq u(t) \leq \pi$

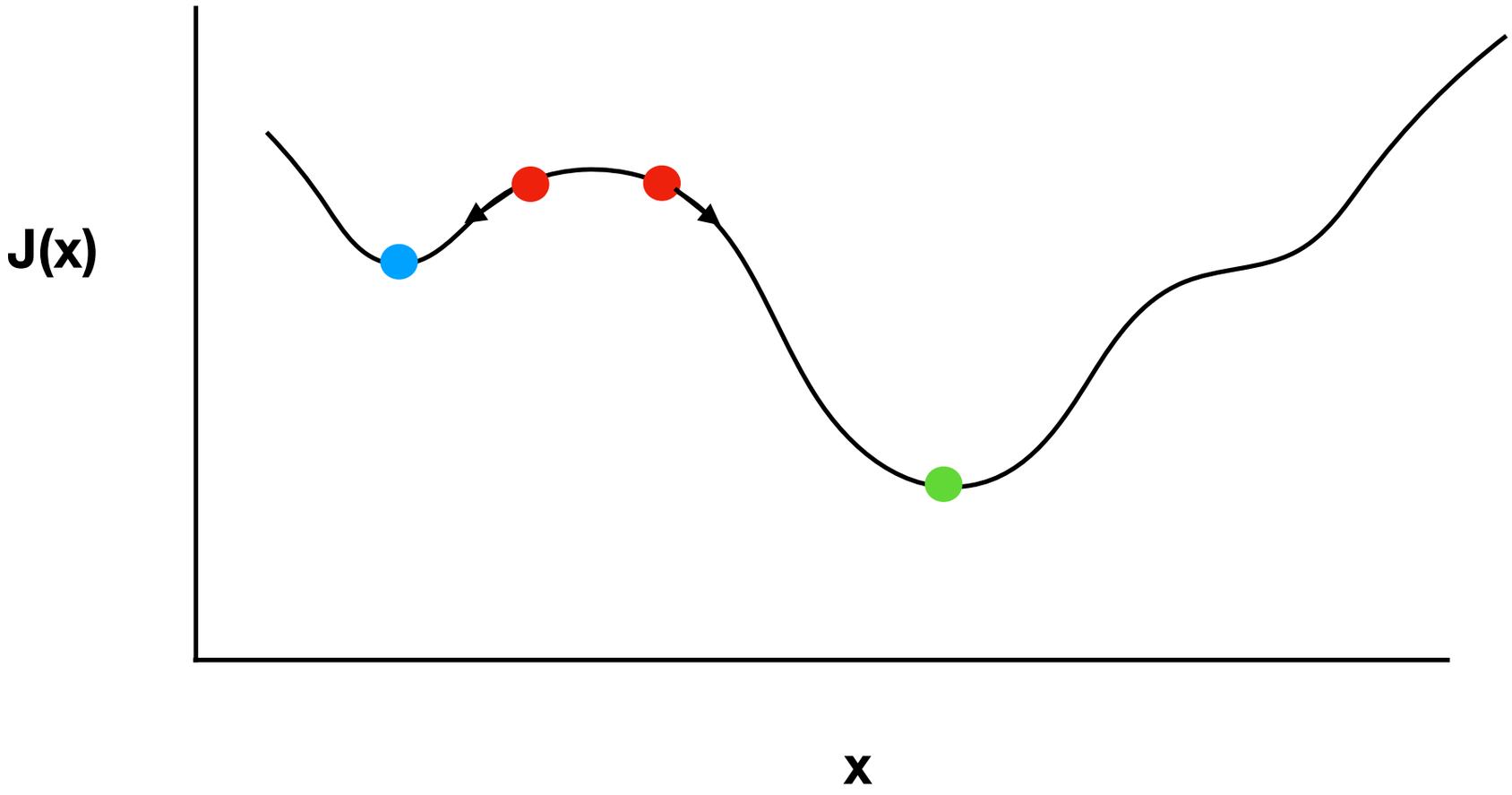
over a **time interval** $t \in [0, T]$

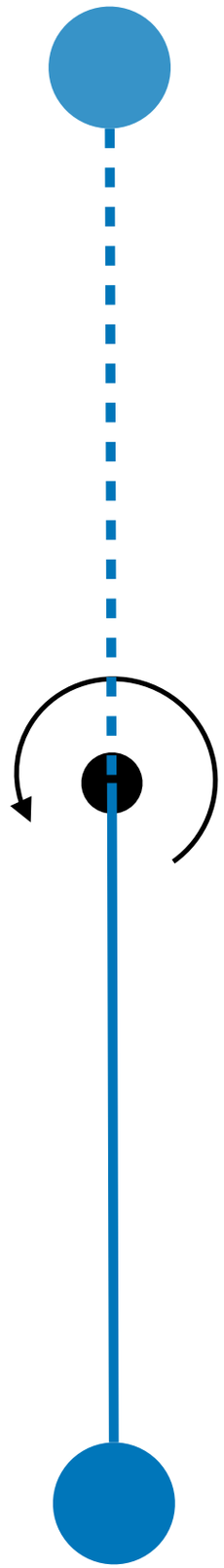
{Open the Brachistochrone folder}





Initial Guess

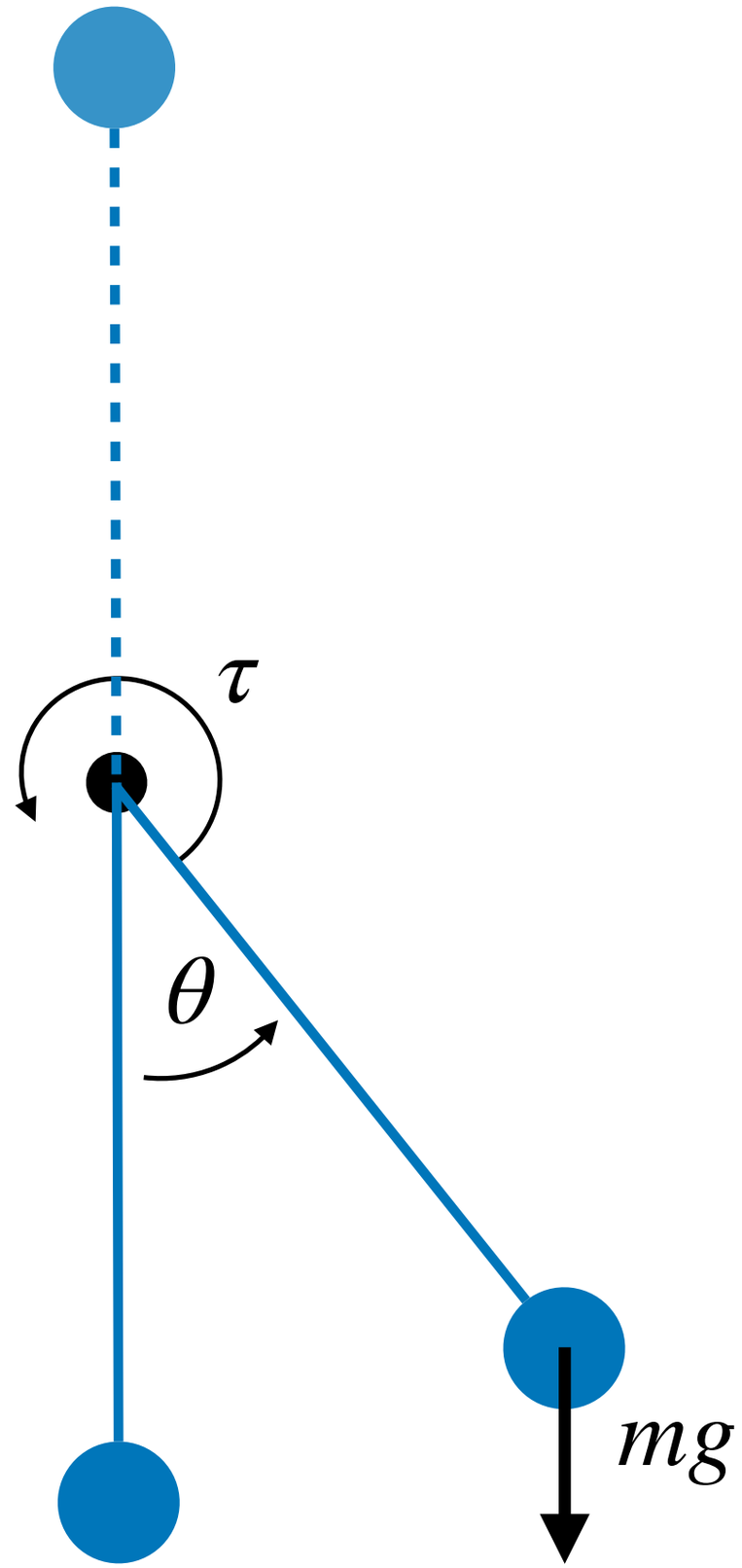




Starting and ending from rest, rotate the arm from a inferior to superior position while minimizing shoulder torque squared

$$J = \int_0^T \tau(t)^2 \, \mathrm{d}t$$

Model the arm as a point of mass *M* located at length *L* from the shoulder.



Step 1: Dynamics!

$$ML^2\ddot{\theta} = \tau - MgL\sin(\theta)$$

$$\Rightarrow \ddot{\theta} = \frac{\tau}{ML^2} - g\sin(\theta)/L$$

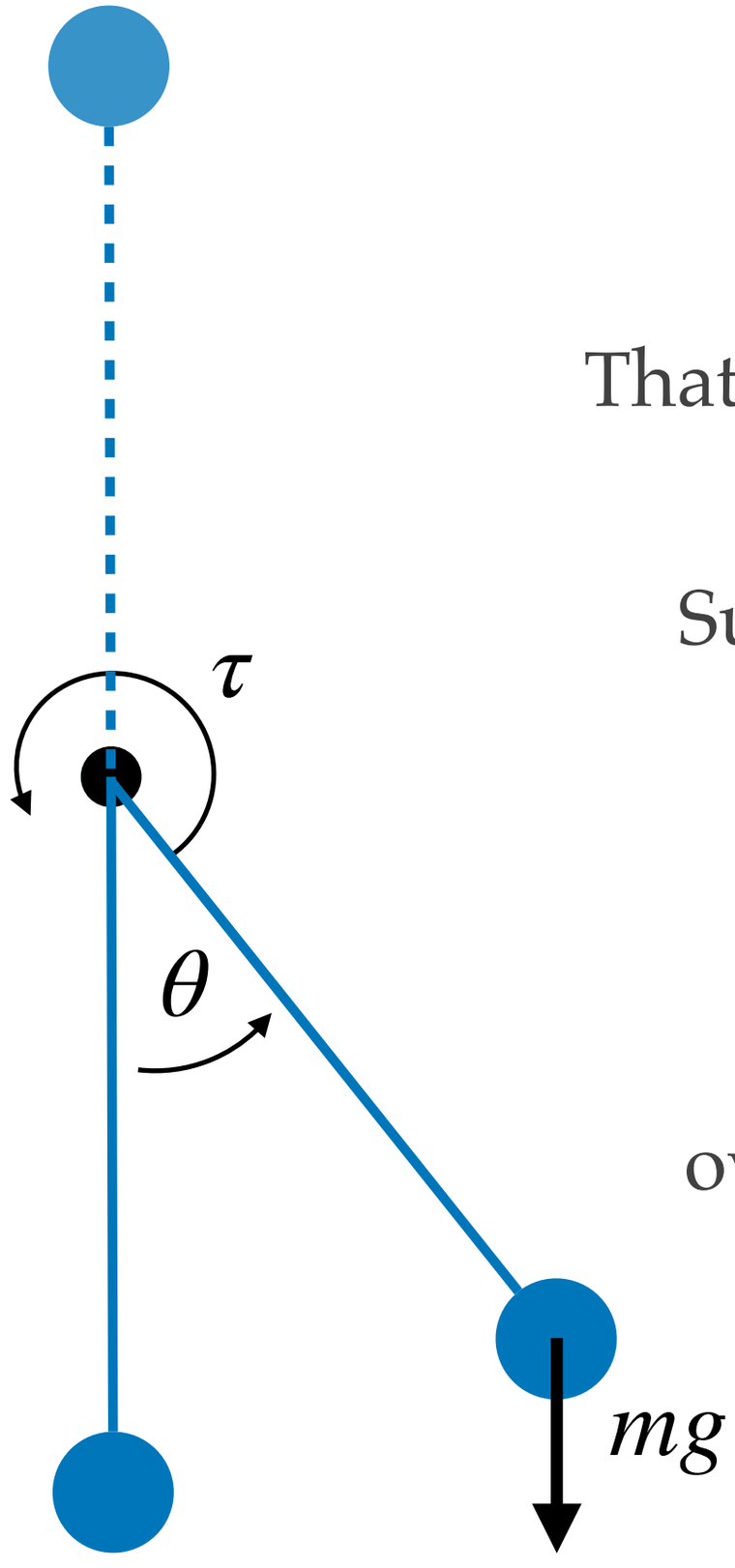
$$\ddot{\theta} = U - g/L\ \sin(\theta)$$

$$\Downarrow$$

Two states:

$$\overrightarrow{X} = [\theta(t), \omega(t)]^T$$

One control: $U(t)$



Given the **states** $\vec{X} = [\theta(t), \omega(t)]^T$

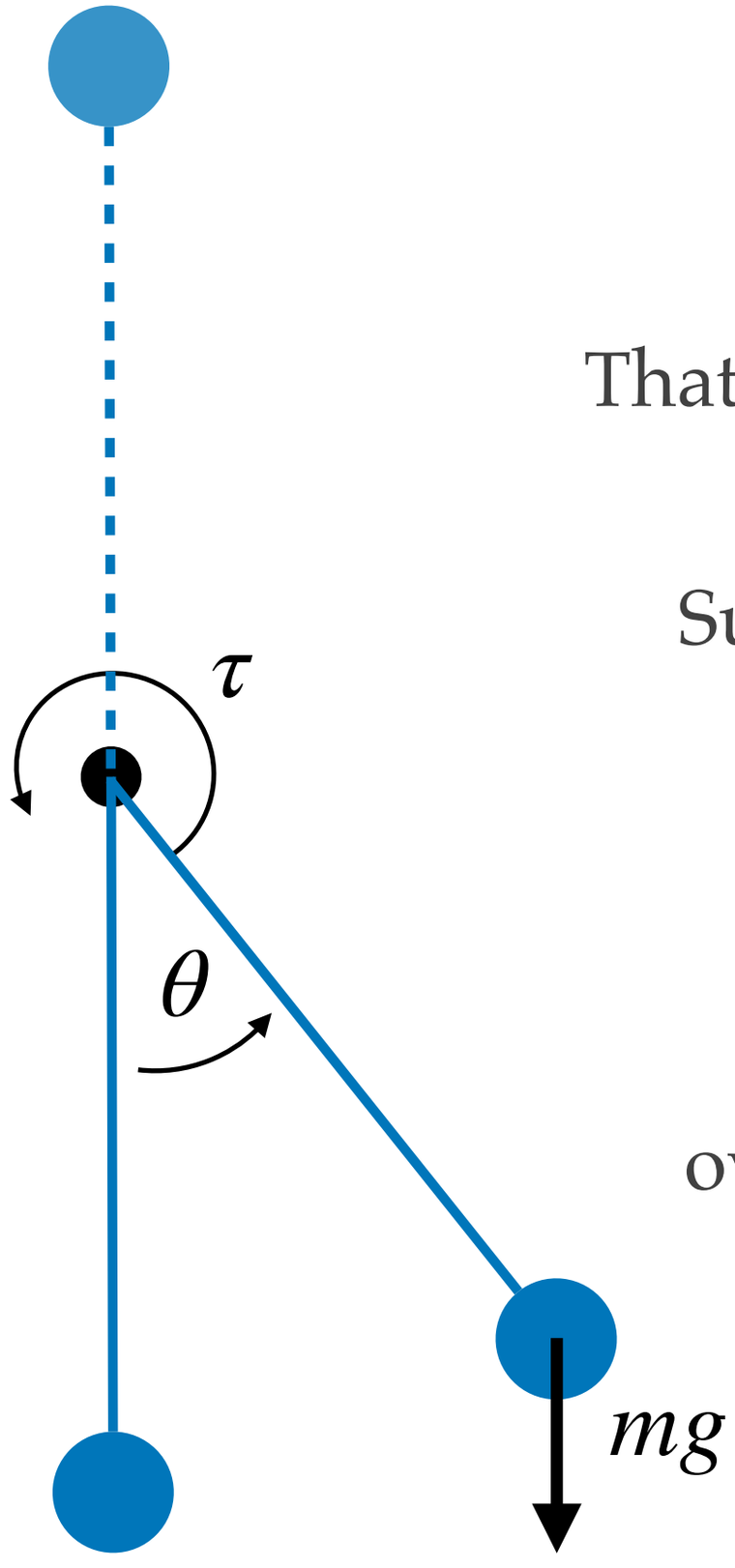
Find the **controls** $U(t)$

That minimize the **cost** $J = \int_0^T \tau(t)^2 \, \mathrm{d}t$

Subject to **dynamics** $\dot{\vec{X}} = \begin{bmatrix} \omega \\ U - g\sin(\theta)/L \end{bmatrix}$

and **constraints**

over a **time interval**



Given the **states** $\vec{X} = [\theta(t), \omega(t)]^T$

Find the **controls** $U(t)$

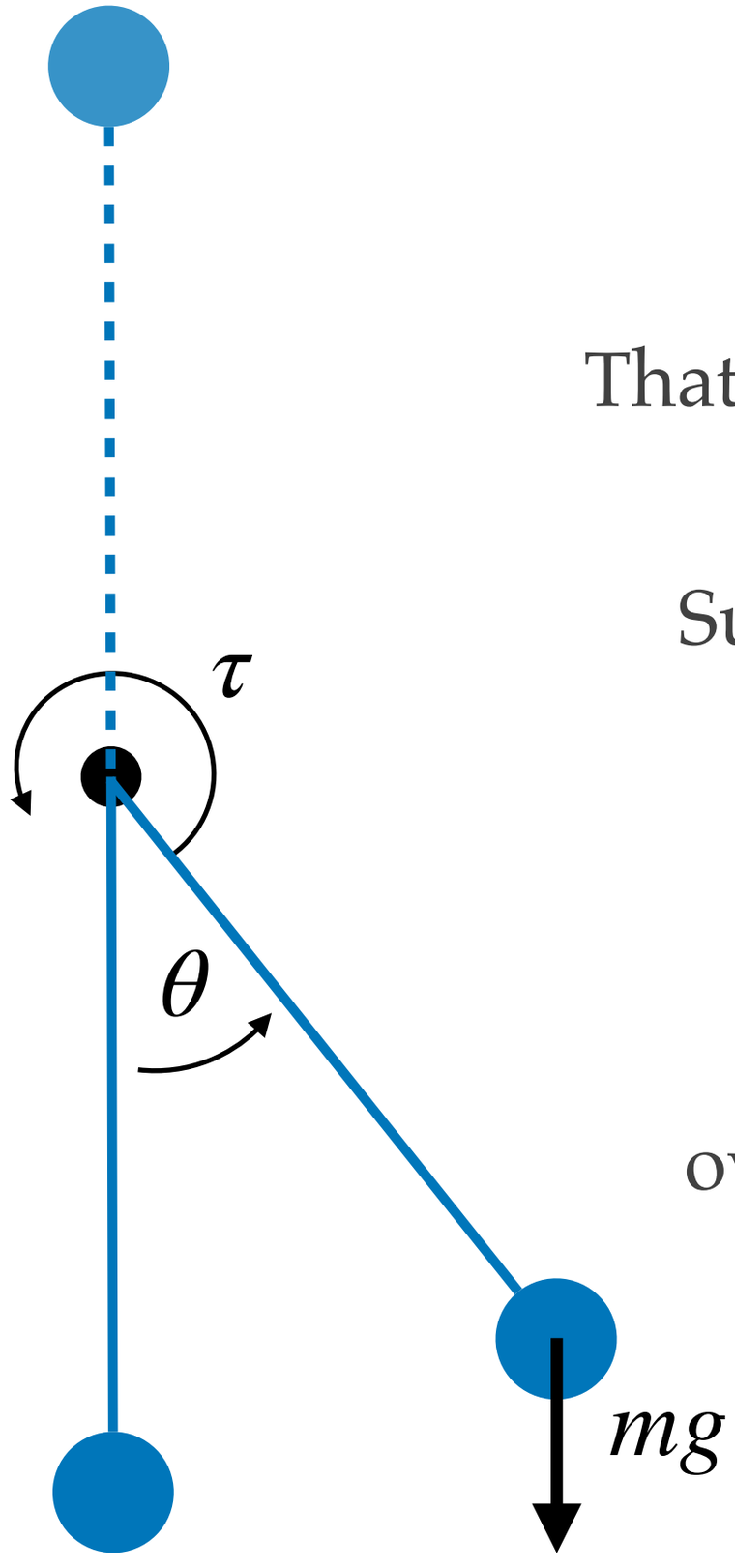
That minimize the **cost** $J = \int_0^T \tau(t)^2 \, \mathrm{d}t$

Subject to **dynamics** $\dot{\vec{X}} = \begin{bmatrix} \omega \\ U - g\sin(\theta)/L \end{bmatrix}$

and **constraints**

$$[\theta(0), \omega(0)] = \vec{0}$$
$$[\theta(T), \omega(T)] = [\pi, 0]$$

over a **time interval**



Given the **states** $\vec{X} = [\theta(t), \omega(t)]^T$

Find the **controls** $U(t)$

That minimize the **cost** $J = \int_0^T \tau(t)^2 \, \mathrm{d}t$

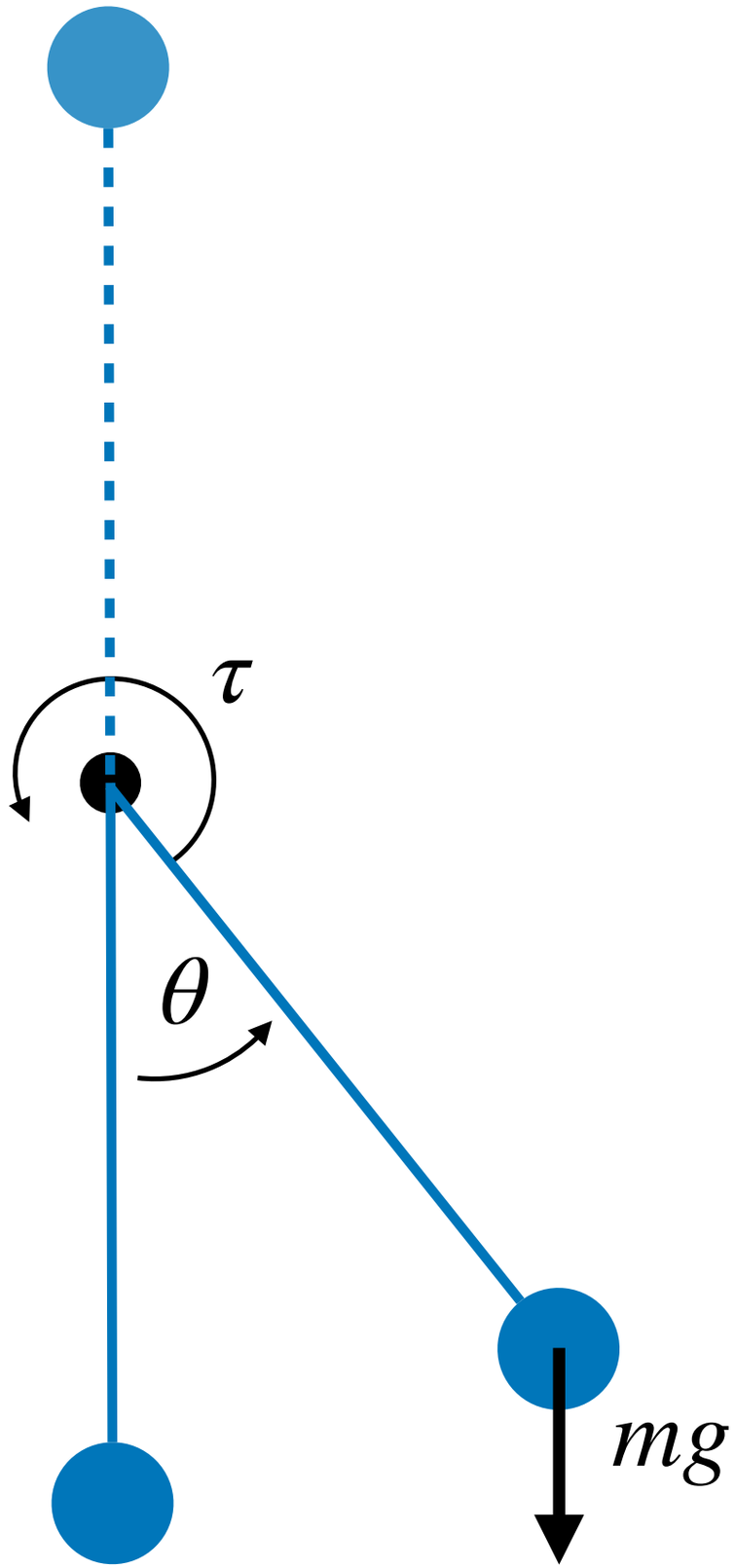
Subject to **dynamics** $\dot{\vec{X}} = \begin{bmatrix} \omega \\ U - g\sin(\theta)/L \end{bmatrix}$

and **constraints** $[\theta(0), \omega(0)] = \vec{0}$

$[\theta(T), \omega(T)] = [\pi, 0]$

over a **time interval** $0 \le t \le T_{\max}$

{Code and solve the problem!}



Given the **states** $\vec{X} = [\theta(t), \omega(t)]^T$

Find the **controls** $U(t)$

That minimize the **cost** $$J = \int_0^T |\, U(t) \cdot \omega(t) \,|\; \mathrm{d}t$$

Subject to **dynamics** $$\dot{\vec{X}} = \begin{bmatrix} \omega \\ U - g\sin(\theta)/L \end{bmatrix}$$

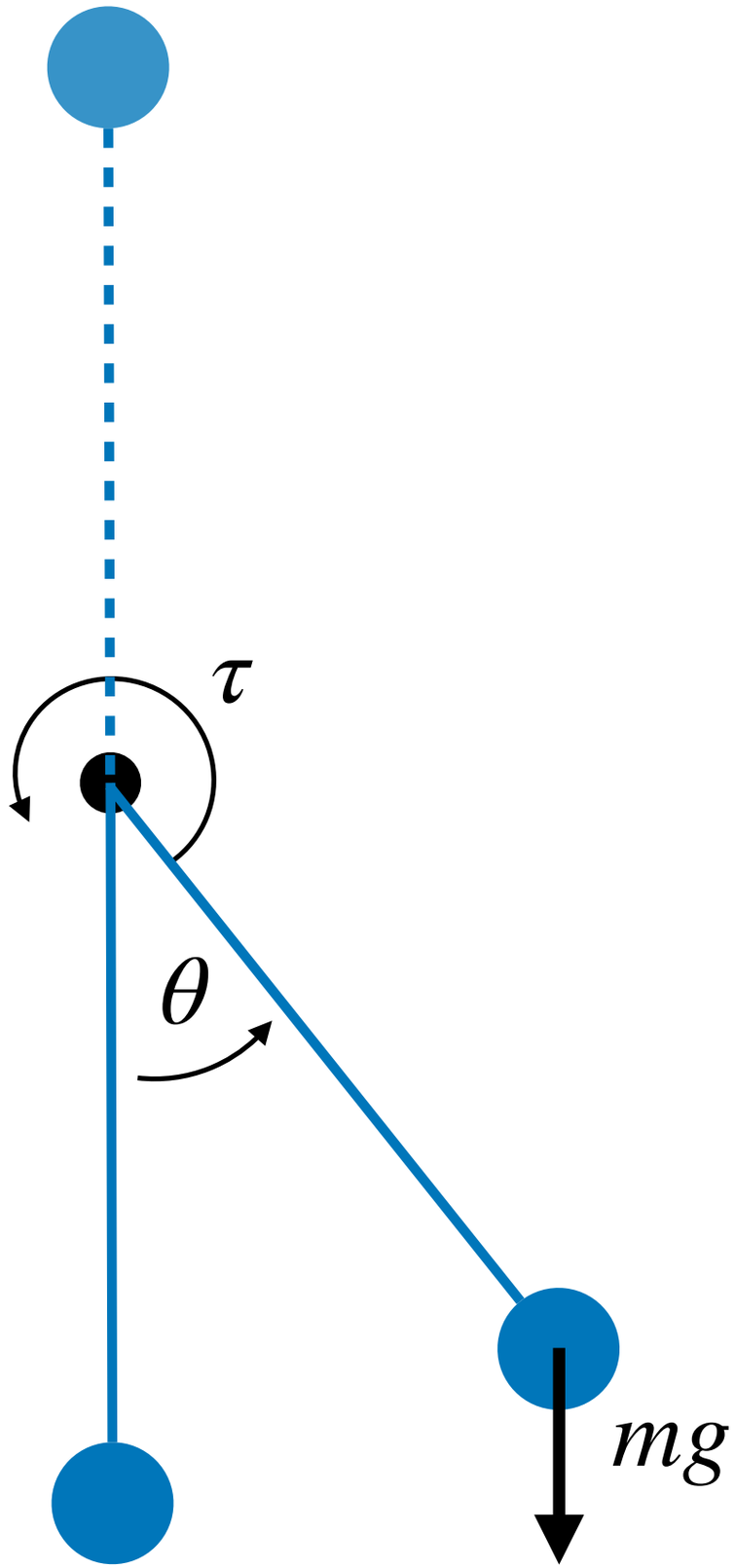
and **constraints**

$$[\theta(0), \omega(0)] = \vec{0}$$

$$[\theta(T), \omega(T)] = [\pi, 0]$$

$$? \leq U \leq ?$$

over a **time interval** $0 \leq t \leq T_{\max}$



Given the **states** $\vec{X} = [\theta(t), \omega(t)]^T$

Find the **controls** $U(t)$

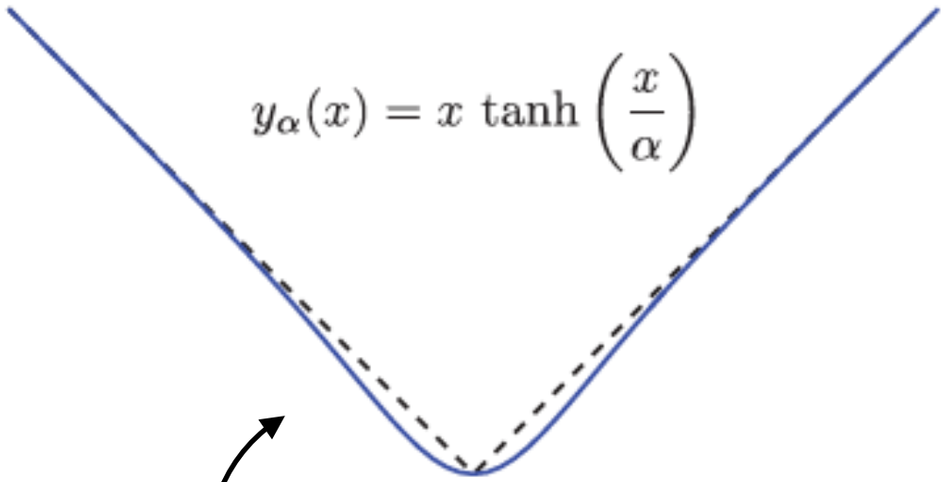
That minimize the **cost** $$J = \int_0^T |\, U(t) \cdot \omega(t) \,|\; \mathrm{d}t$$

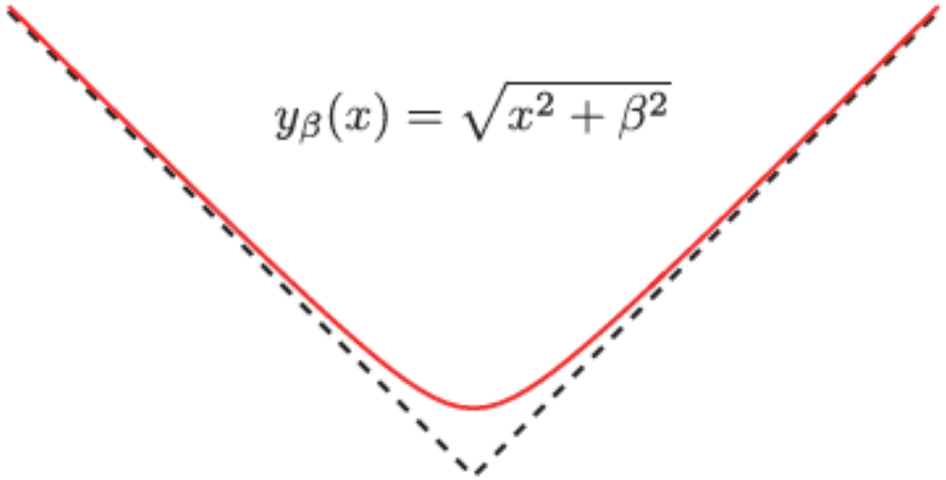
Subject to **dynamics** $$\dot{\vec{X}} = \begin{bmatrix} \omega \\ U - g\sin(\theta)/L \end{bmatrix}$$

and **constraints**
$$[\theta(0), \omega(0)] = \vec{0}$$
$$[\theta(T), \omega(T)] = [\pi, 0]$$
$$-U_{\max} \le U \le U_{\max}$$

over a **time interval** $$0 \le t \le T_{\max}$$

{Code and solve the problem!}



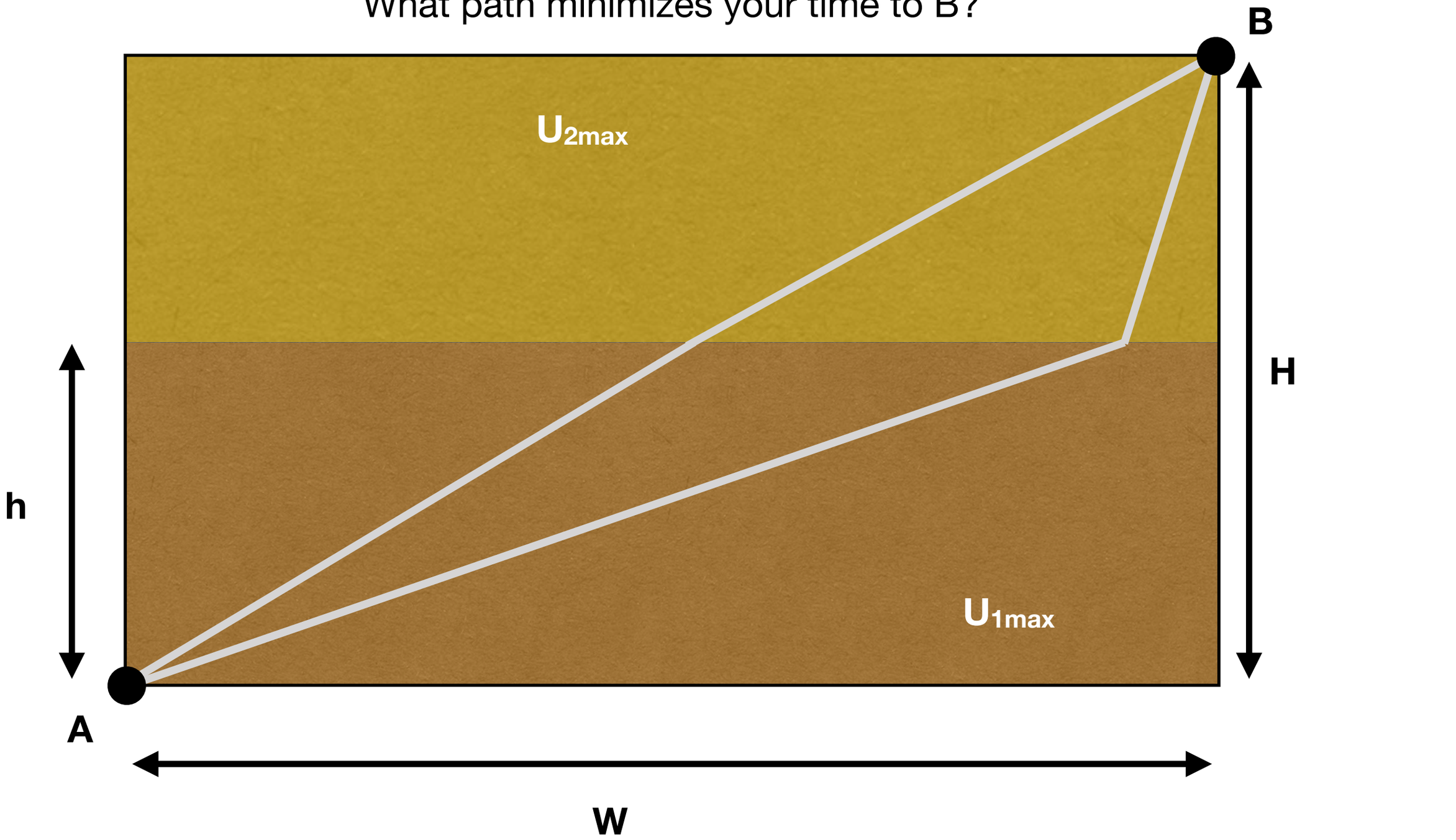


- **Smoothing approaches**
- **“Exact” way: Slack variables**

Kelly 2017 *SIAM Review* 59: 849-904

Multi-phase optimization

You're on a sandy beach. You're at A and want to get to B in minimum time
You can run at U1max on the wet sand, and U2max on the dry sand.
What path minimizes your time to B?

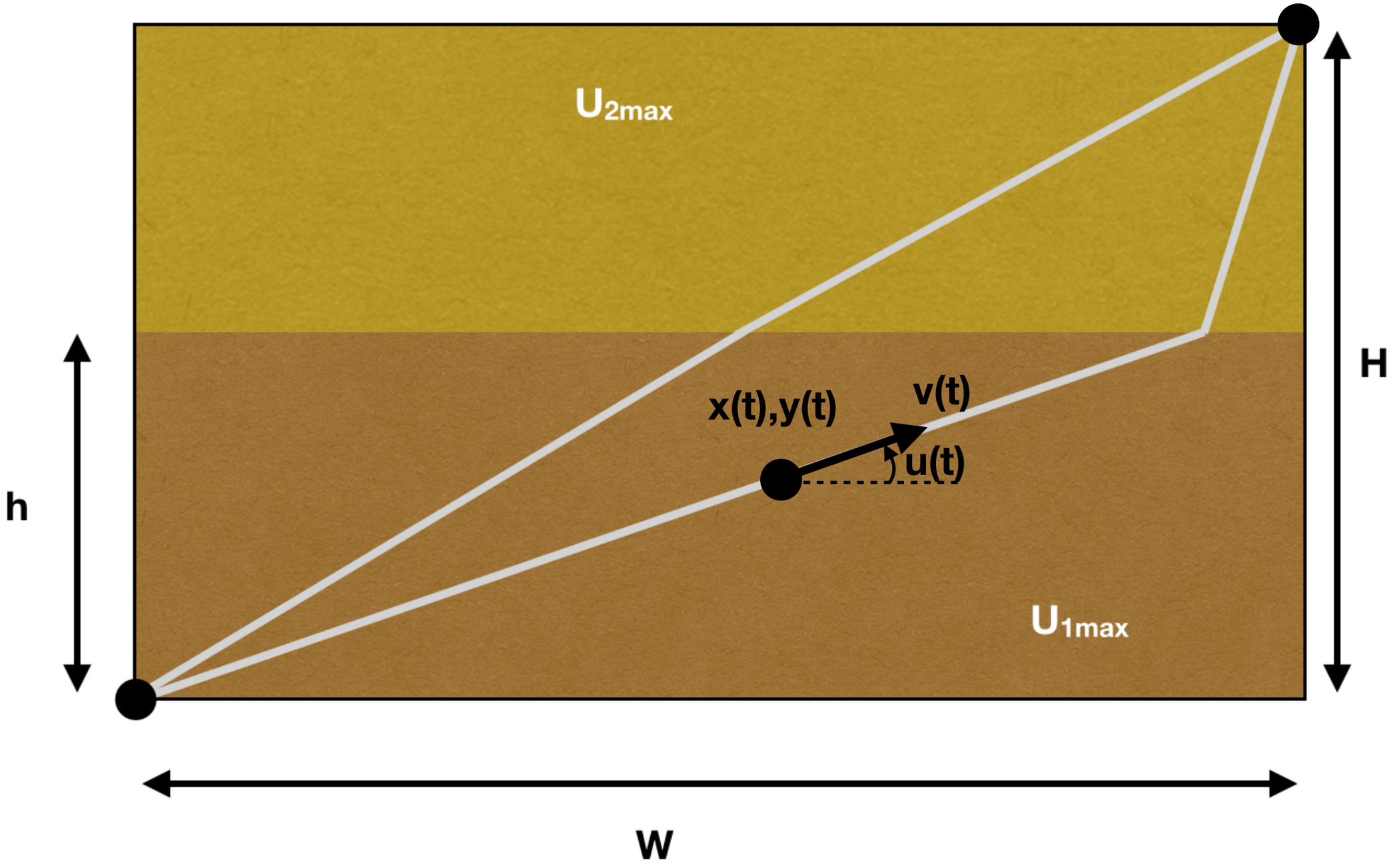


Analogous to Snell's law of light refraction

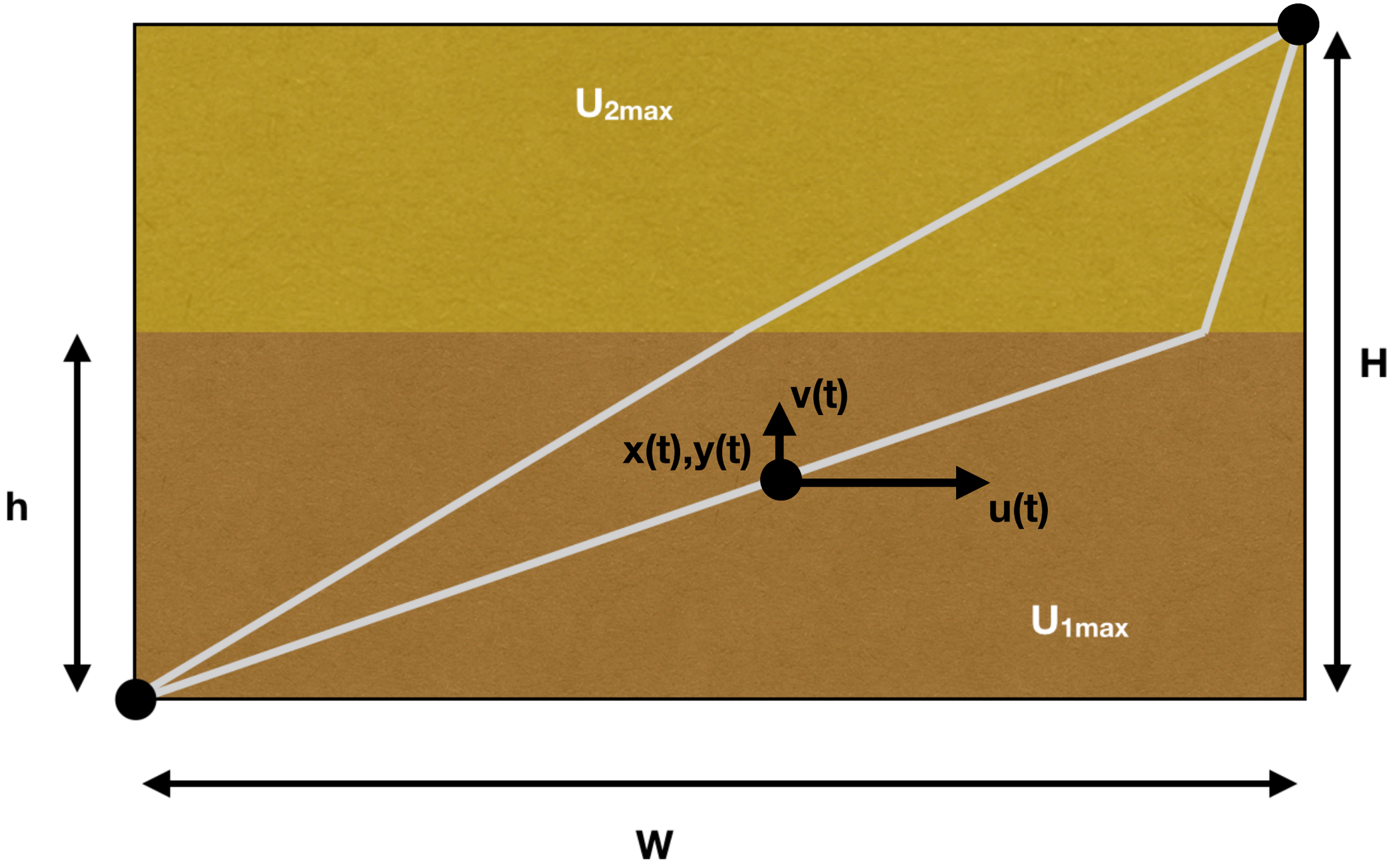
- In multiple-phase optimization, each phase has different “rules” (dynamics, costs, constraints)

- Often, endpoint constraints will be added: these restrict values at the start and end of each phase

Step 1: Dynamics!



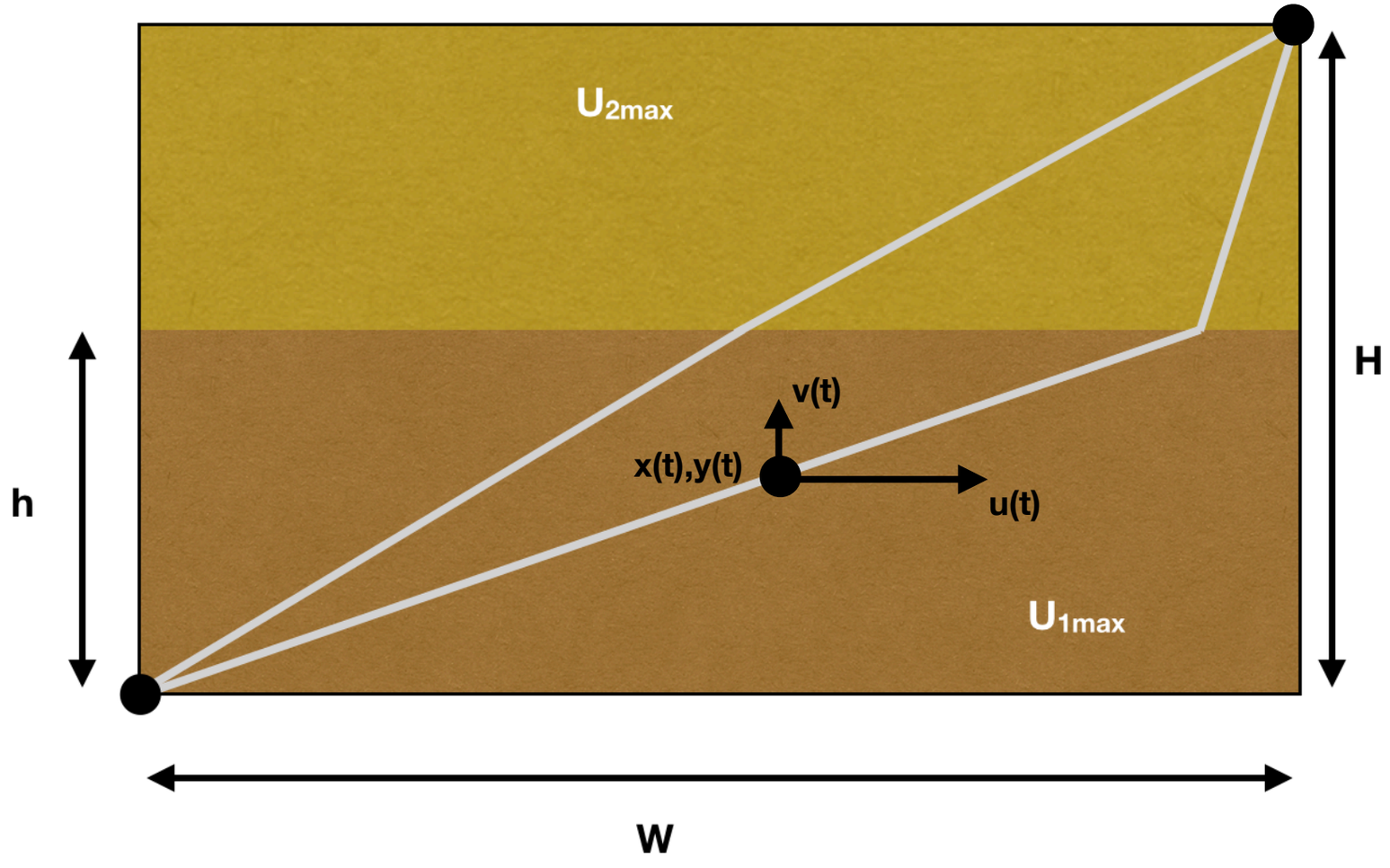
Step 1: Dynamics!



Step 1: Dynamics!

$$\dot{x} = u$$

$$\dot{y} = v$$



Dynamics

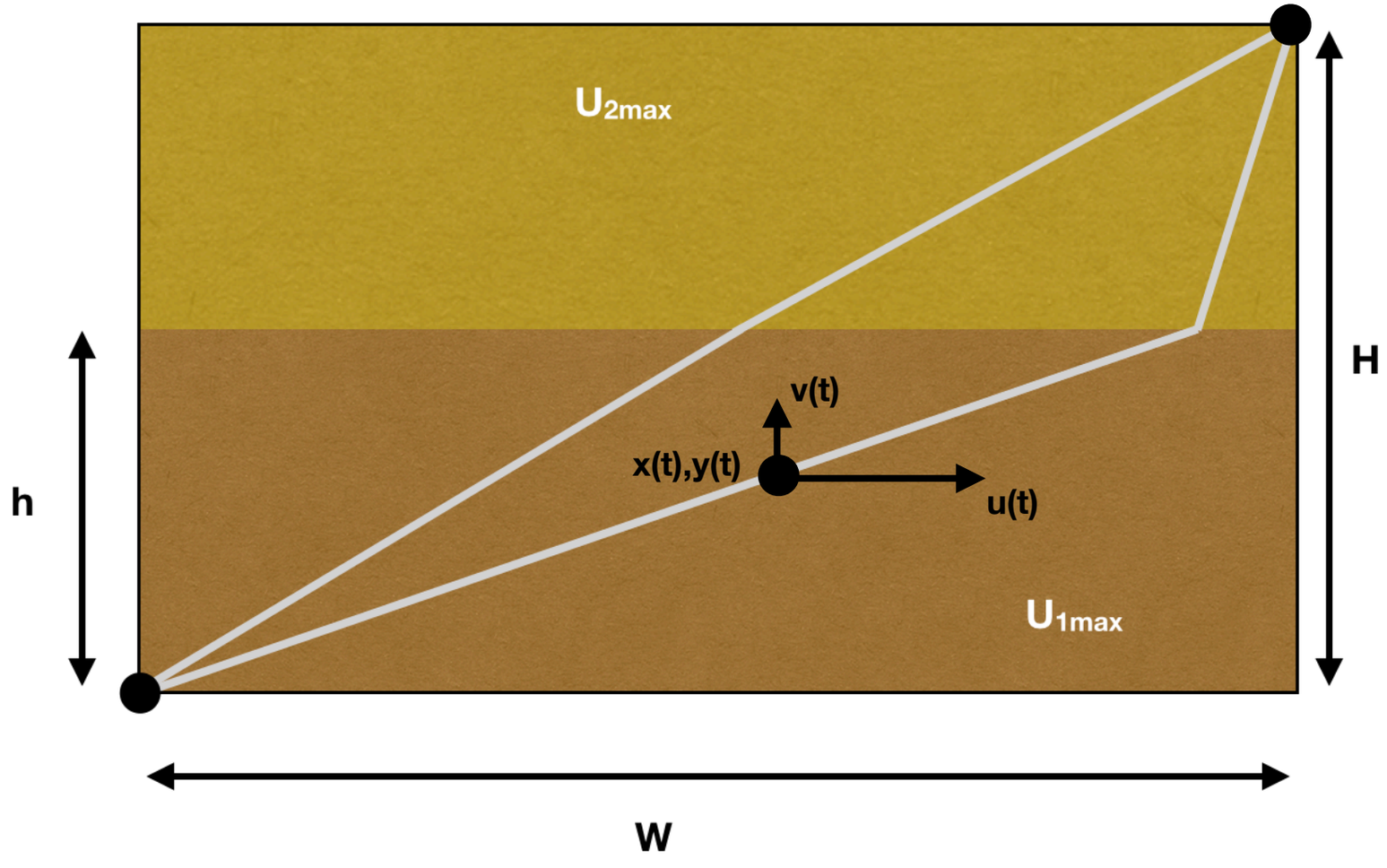
$$\dot{x} = u$$

$$\dot{y} = v$$

Path constraint:

(nonlinear) constraints on variables in time

$$0 \leq u_i(t)^2 + v_i(t)^2 \leq U_{i\max}^2 \quad \forall t$$



Dynamics

$$\dot{x} = u$$

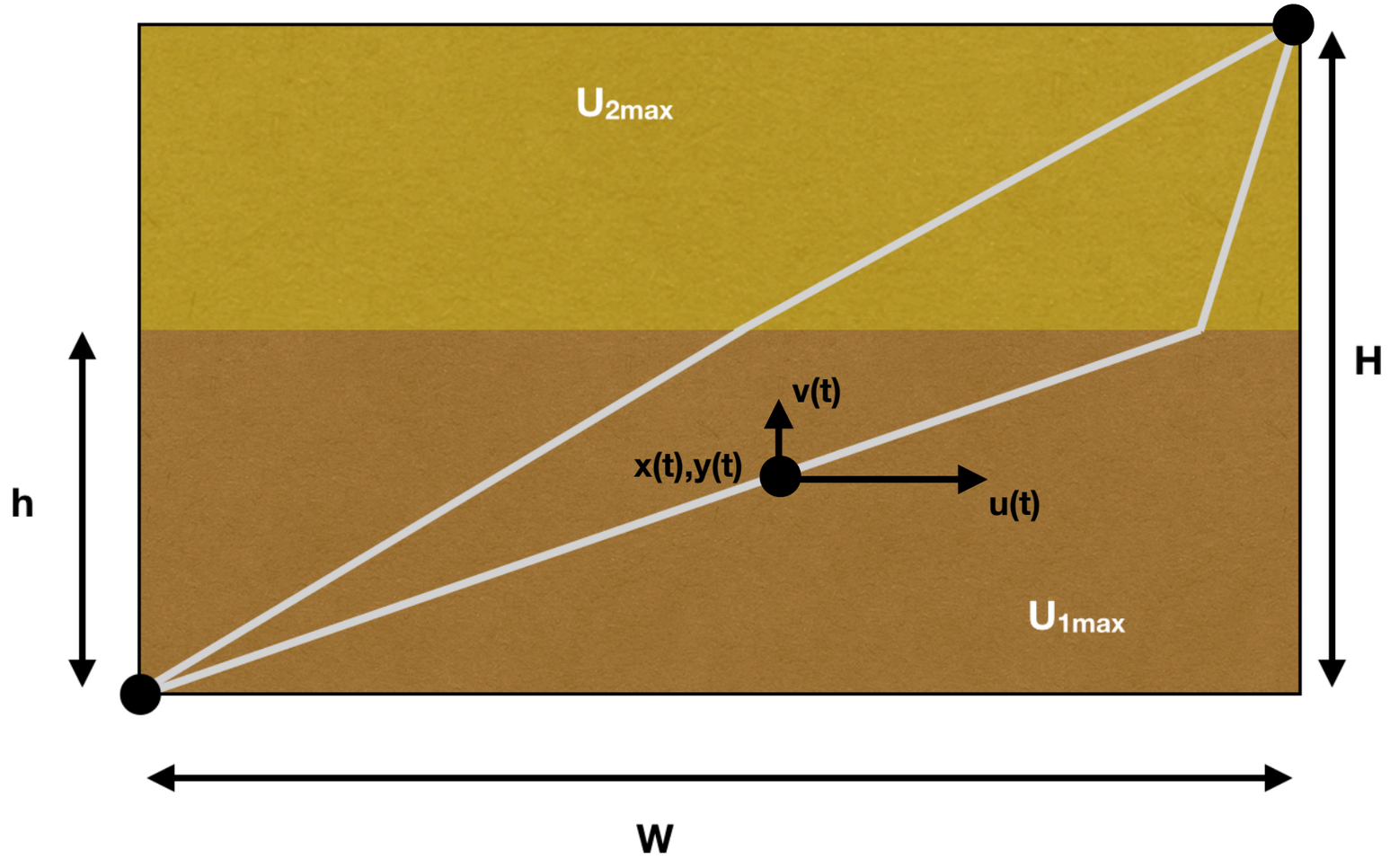
$$\dot{y} = v$$

Path constraint:

(nonlinear) constraints on variables in time

$$u_i(t)^2 + v_i(t)^2 \leq U_{i\max}^2 \quad \forall t$$

Bounds in each phase?

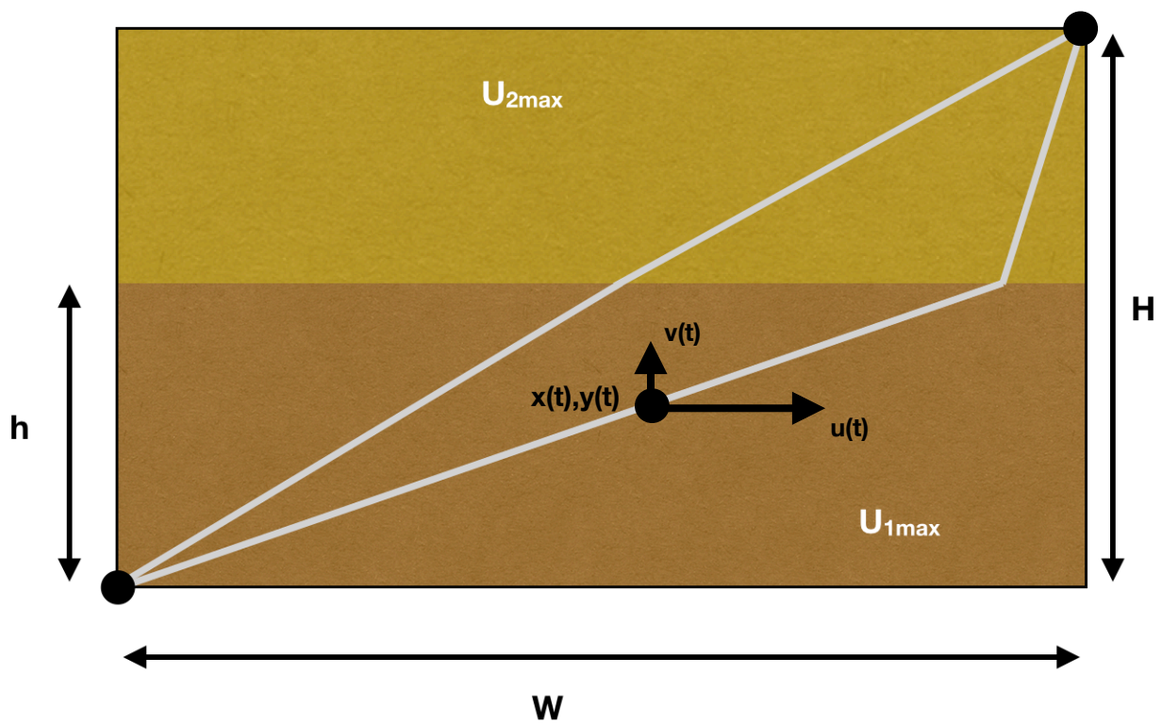


Phase 1

$$0 \le t_1 \le \infty$$
$$x_1(0) = 0$$
$$0 \le x_1(t) \le W$$
$$0 \le x_1(T) \le W$$
$$y_1(0) = 0$$
$$0 \le y_1(t) \le h$$
$$y_1(T) = h$$
$$0 \le u_1 \le U_{1\max}$$

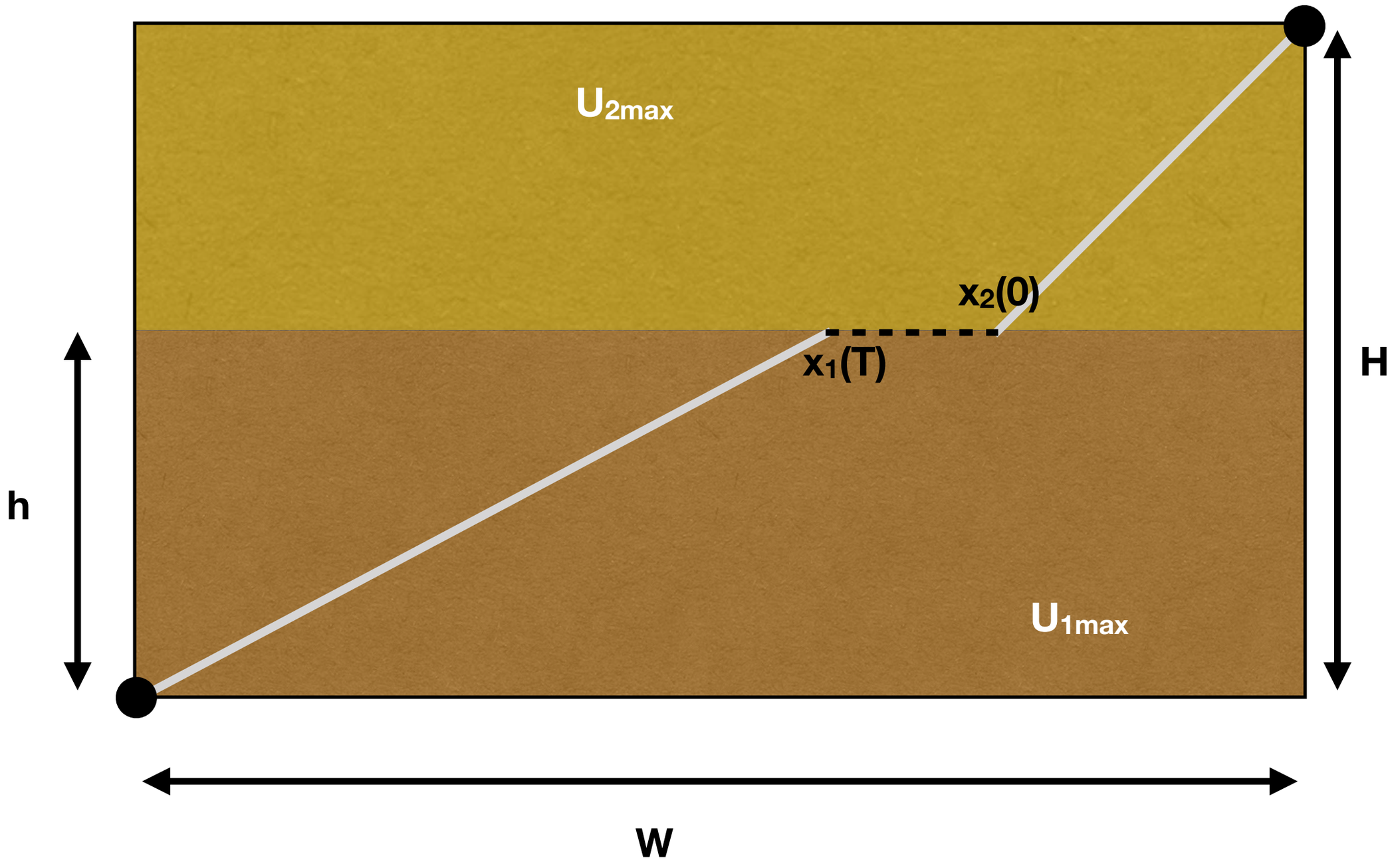
Phase 2

$$0 \le t_2 \le \infty$$
$$0 \le x_2(0) \le W$$
$$0 \le x_2(t) \le W$$
$$x_2(T) = W$$
$$y_1(0) = h$$
$$h \le y_1(t) \le H$$
$$y_2(T) = H$$
$$0 \le u_2 \le U_{2\max}$$



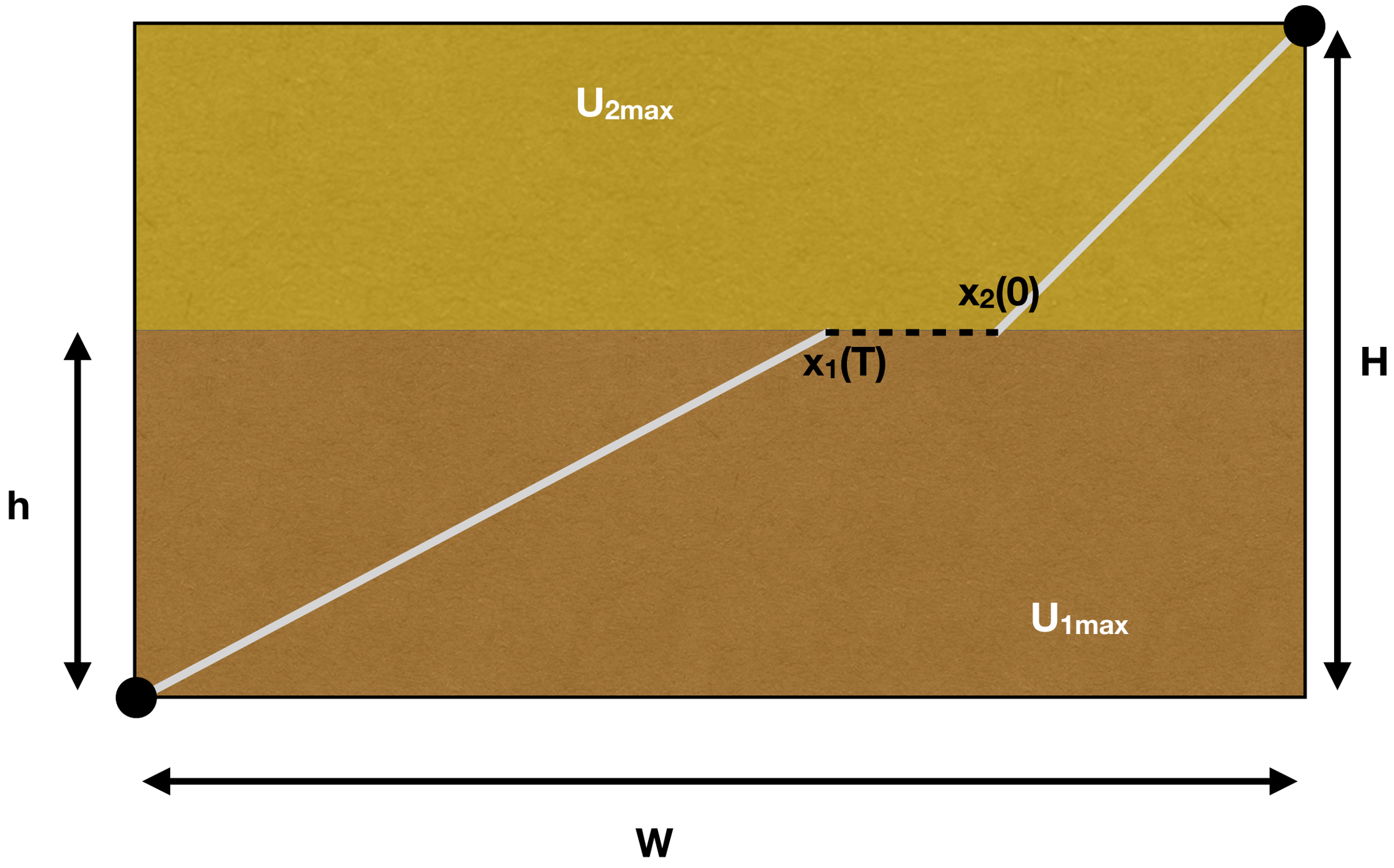
{Fill in the bounds for minTime2Phase.m}

Endpoint constraints



Endpoint constraints

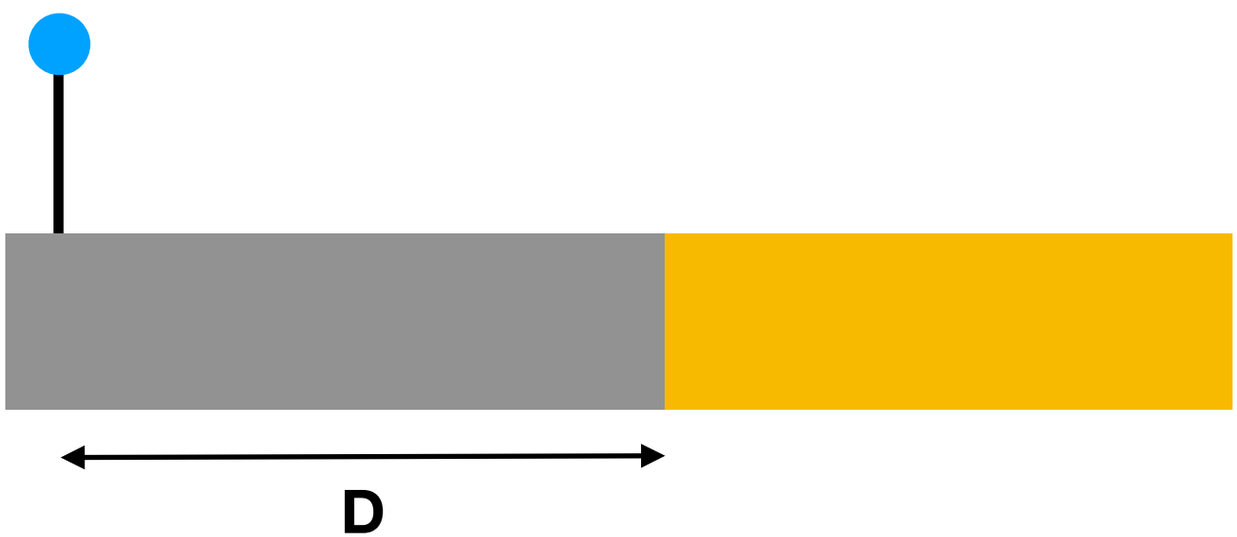
$$x_1(T) - x_2(0) = 0$$

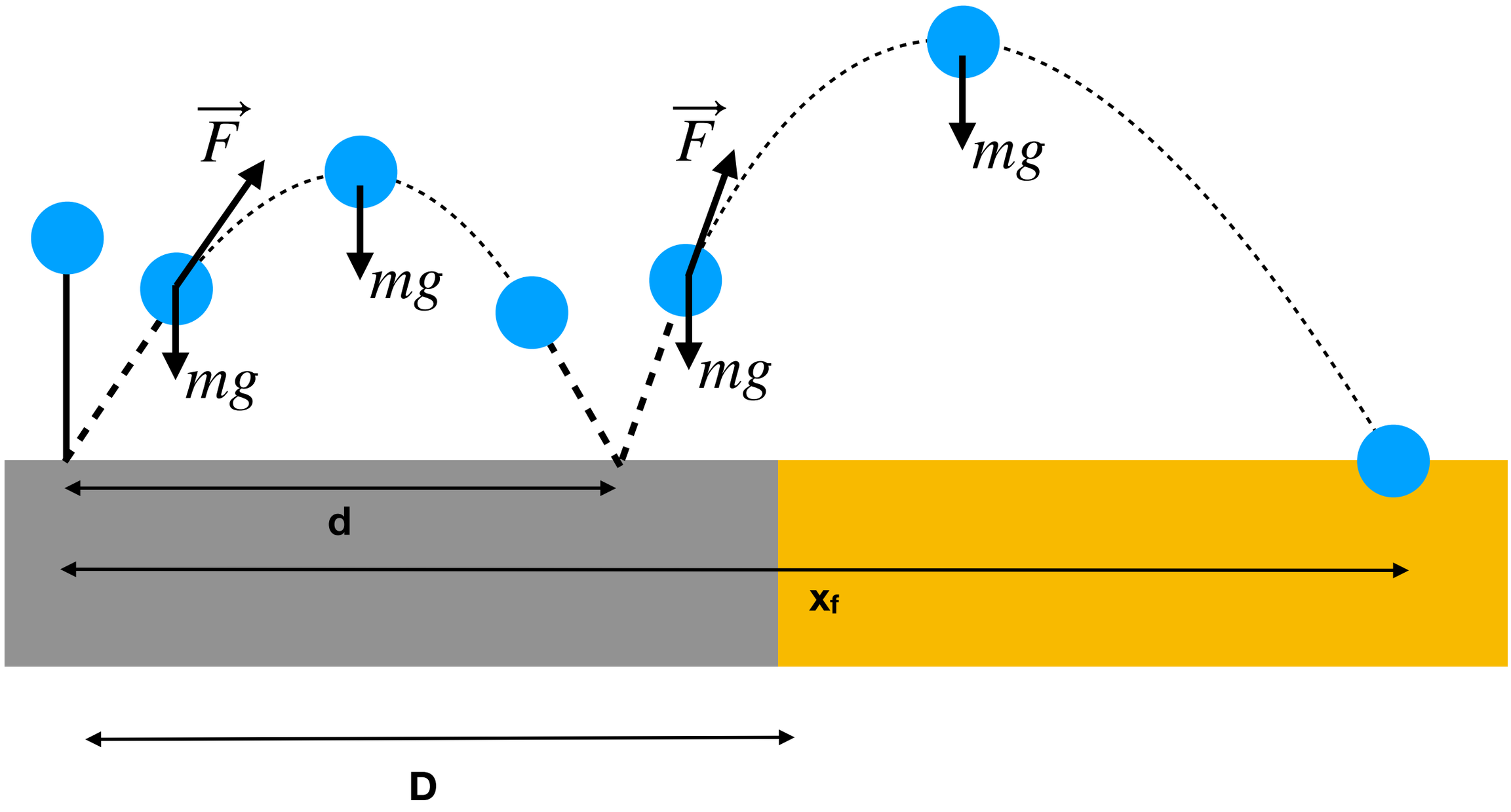


Long Jump!

- You're standing at rest on a track, a distance D from the sand pit. You want to jump as far horizontally as you can, but you can only take one step *before* the sand pit. What is the optimal jumping strategy?

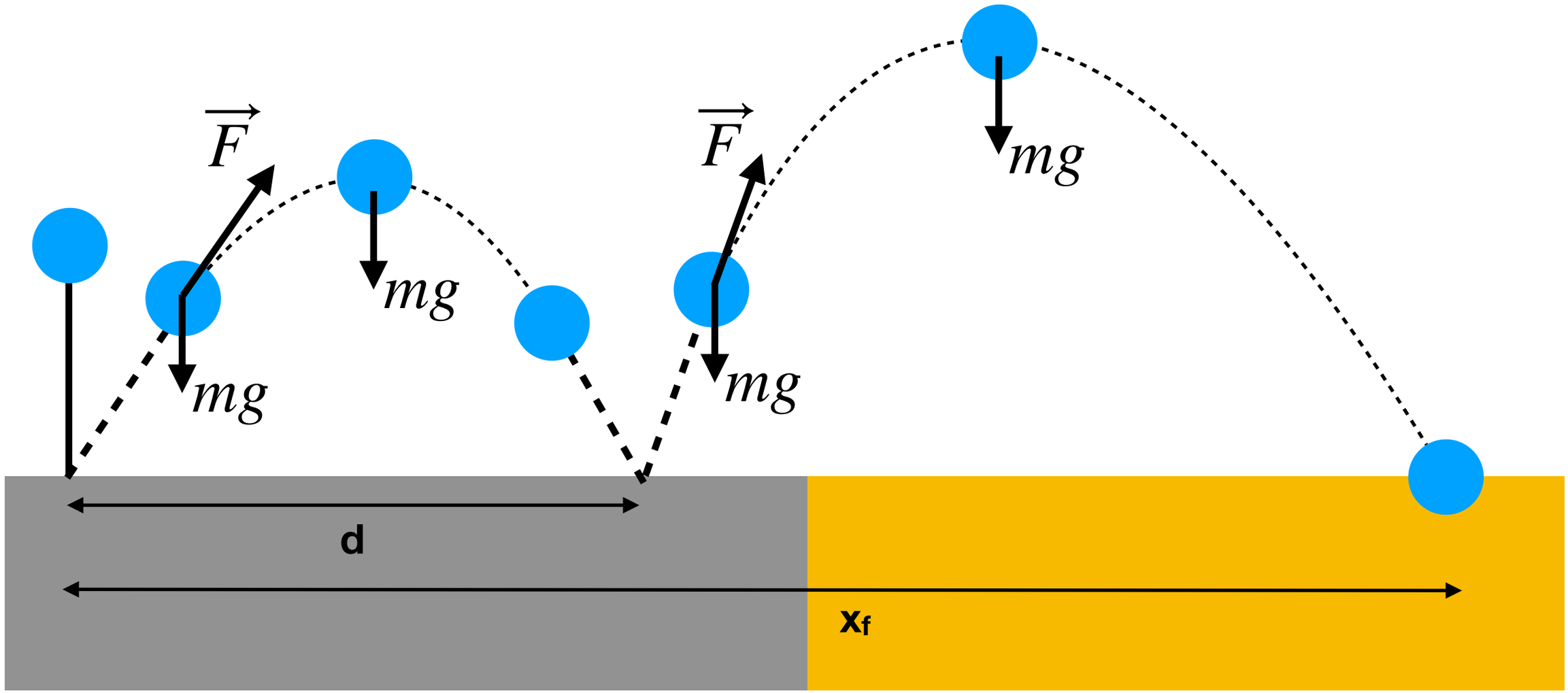
- Assume a point mass with massless legs, maximum and minimum leg length, and maximal leg force of 4*mg*





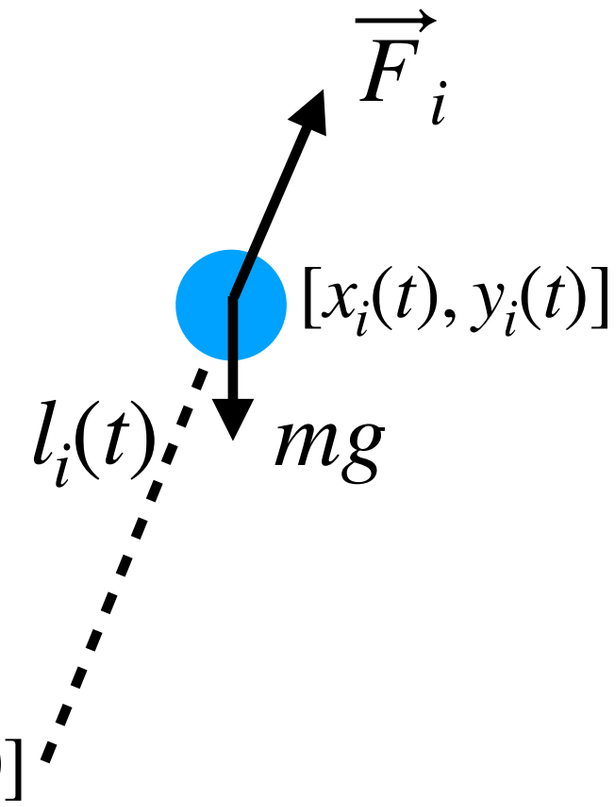
Objective?

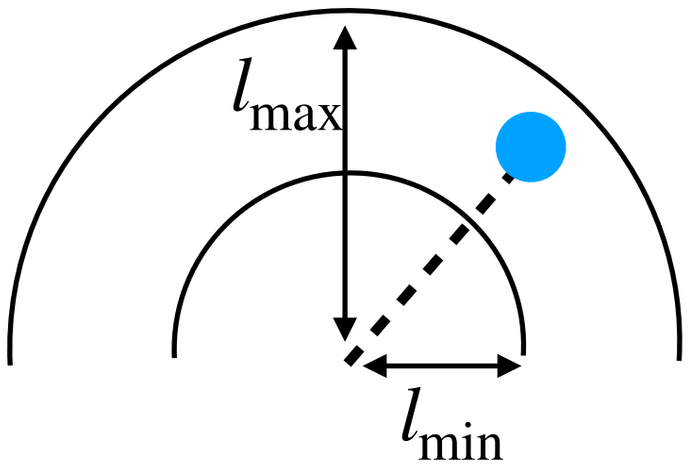
How many phases?



Objective? $\max x_f = \boxed{\min -x_f}$

How many phases?





Dynamics

$$\ddot{x}_i = F_i x_i / l_i$$

$$\ddot{y}_i = F_i y_i / l_i - g$$

$$l_i^2 = x_i^2 + y_i^2$$

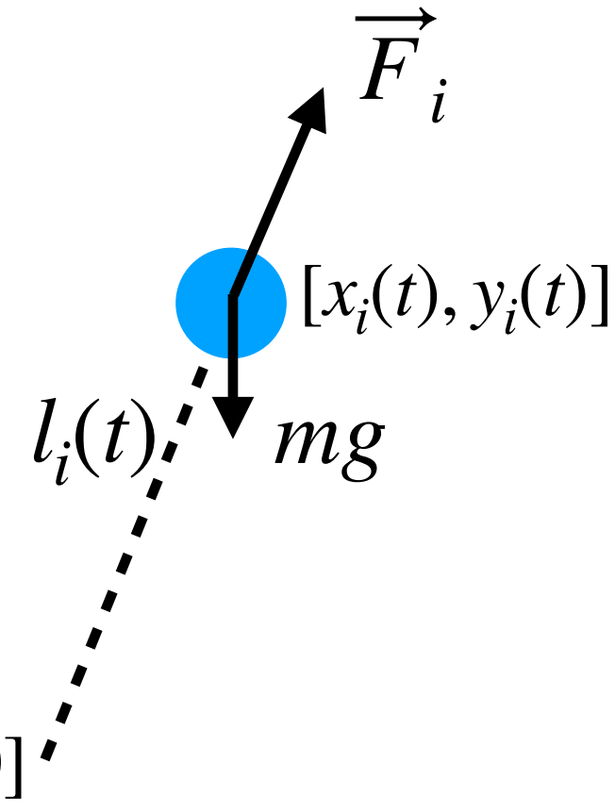
(Note: *F* is force per mass)

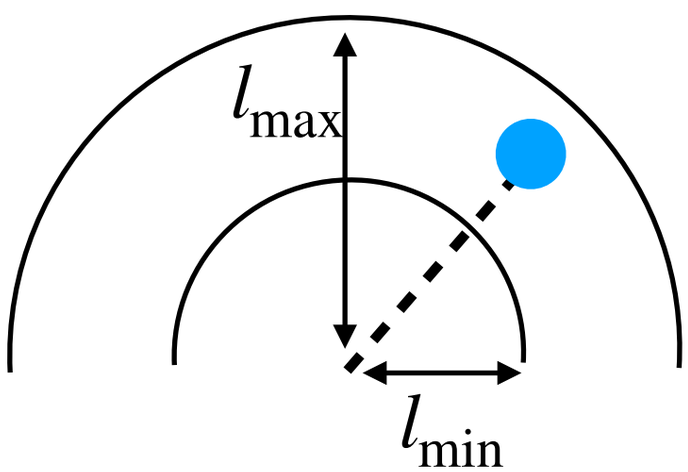
States are?

Controls are?

Path constraints

Fill in the dynamic function





Dynamics

$$\ddot{x}_i = F_i x_i / l_i$$

$$\ddot{y}_i = F_i y_i / l_i - g$$

$$l_i^2 = x_i^2 + y_i^2$$

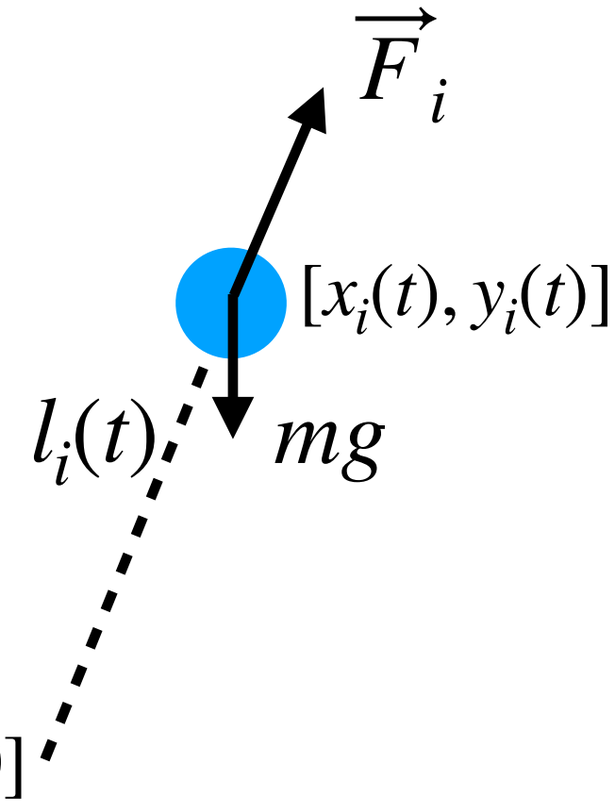
(Note: *F* is force per mass)

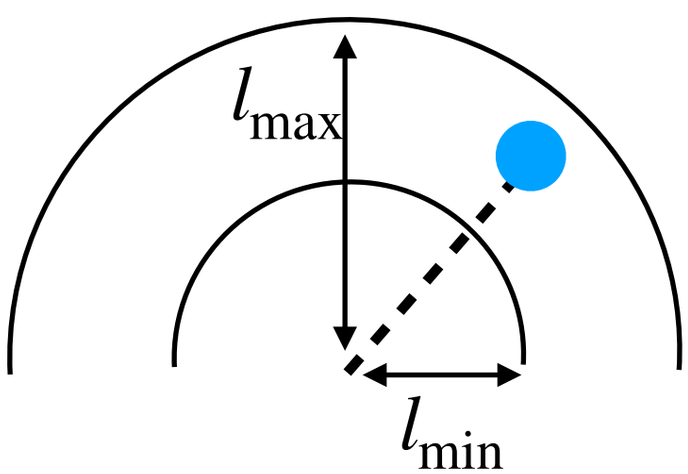
States are? $\vec{x}_i = [x_i, y_i, u_i, v_i]$

Controls are? $U_i = F_i$

Path constraints

Fill in the dynamic function





Dynamics

$$\ddot{x}_i = F_i x_i / l_i$$

$$\ddot{y}_i = F_i y_i / l_i - g$$

$$l_i^2 = x_i^2 + y_i^2$$

(Note: *F* is force per mass)

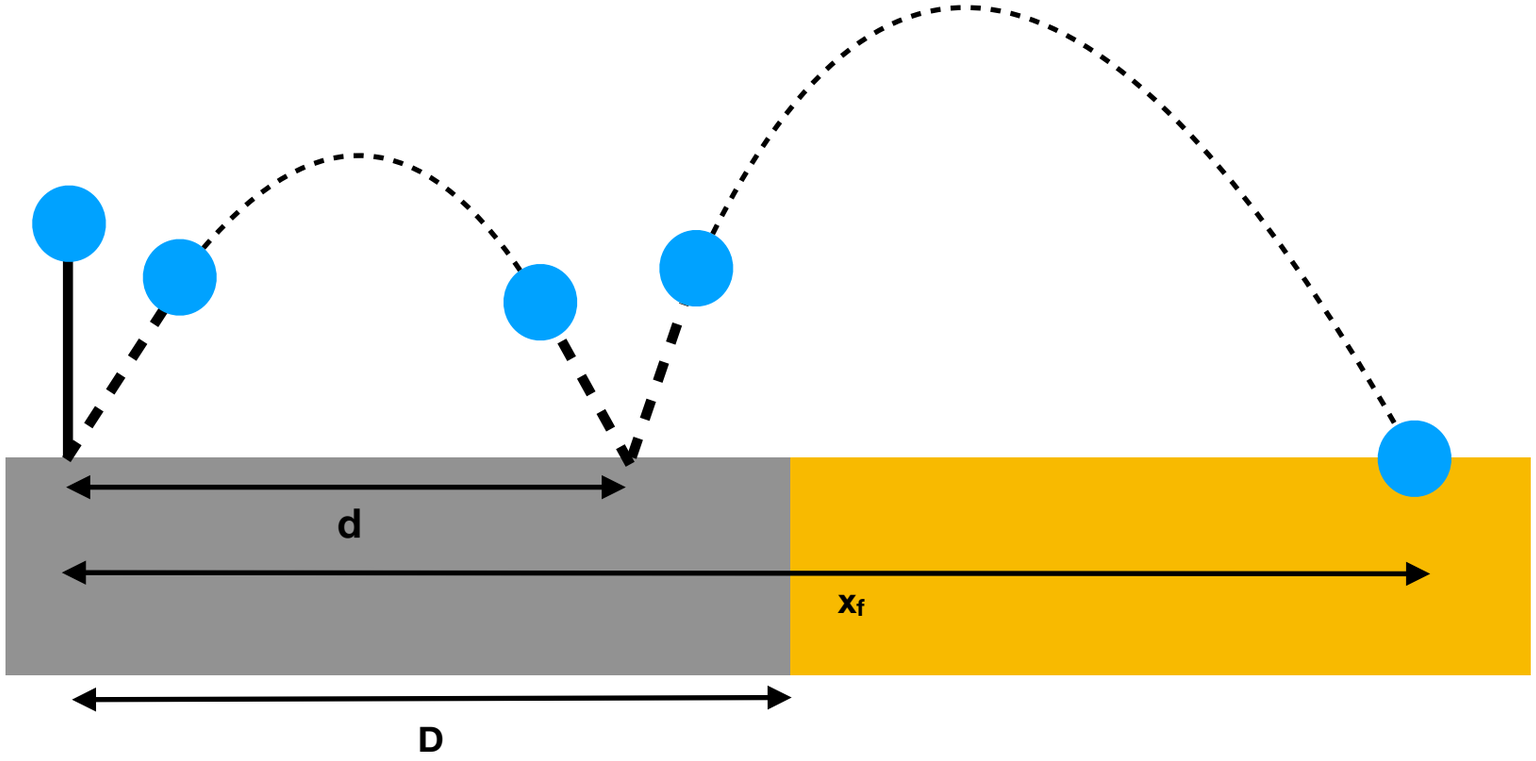
States are? $\vec{x}_i = [x_i, y_i, u_i, v_i]$

Controls are? $U_i = F_i$

Path constraints

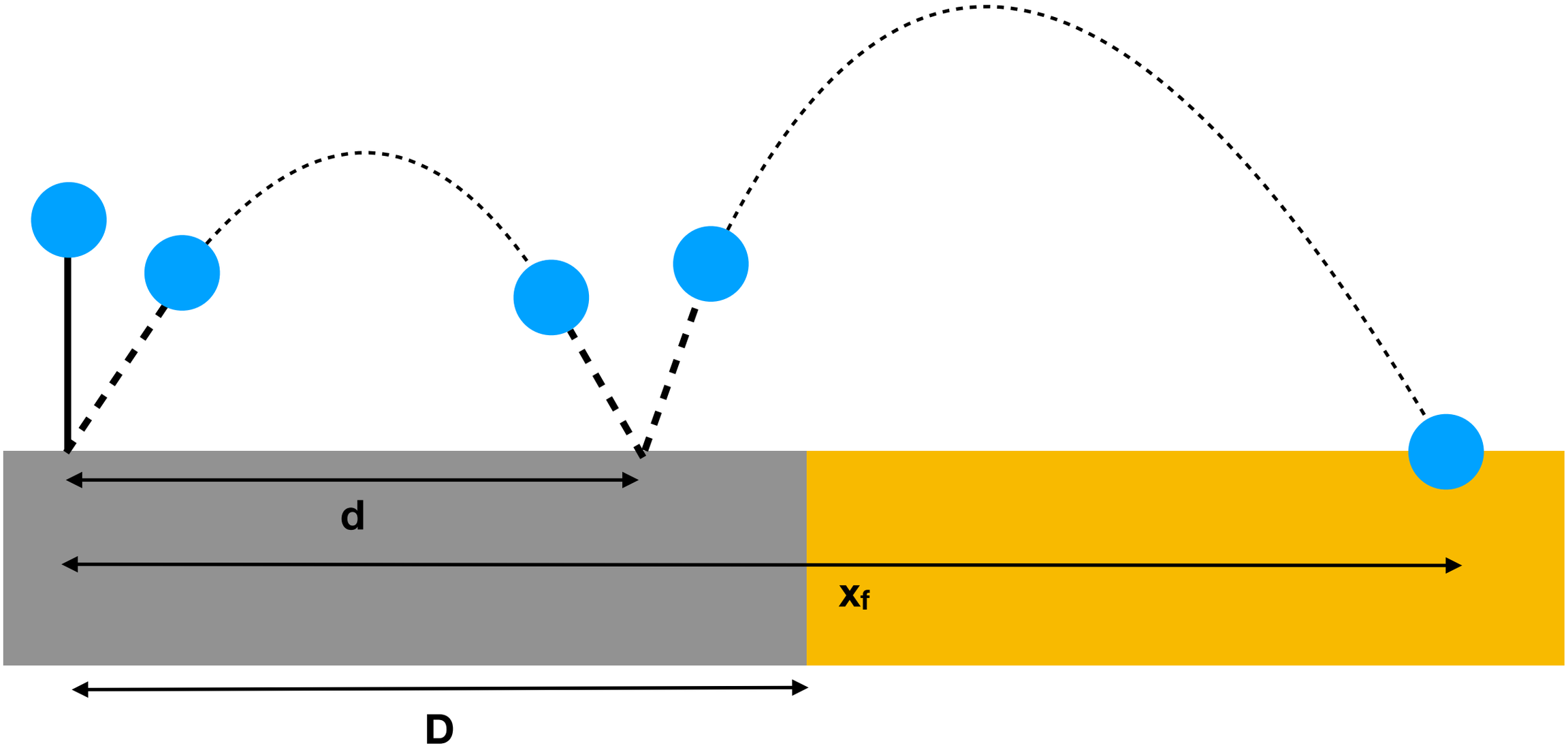
$$l_{\min}^2 \leq x_i^2 + y_i^2 \leq l_{\max}^2$$

Fill in the dynamic function



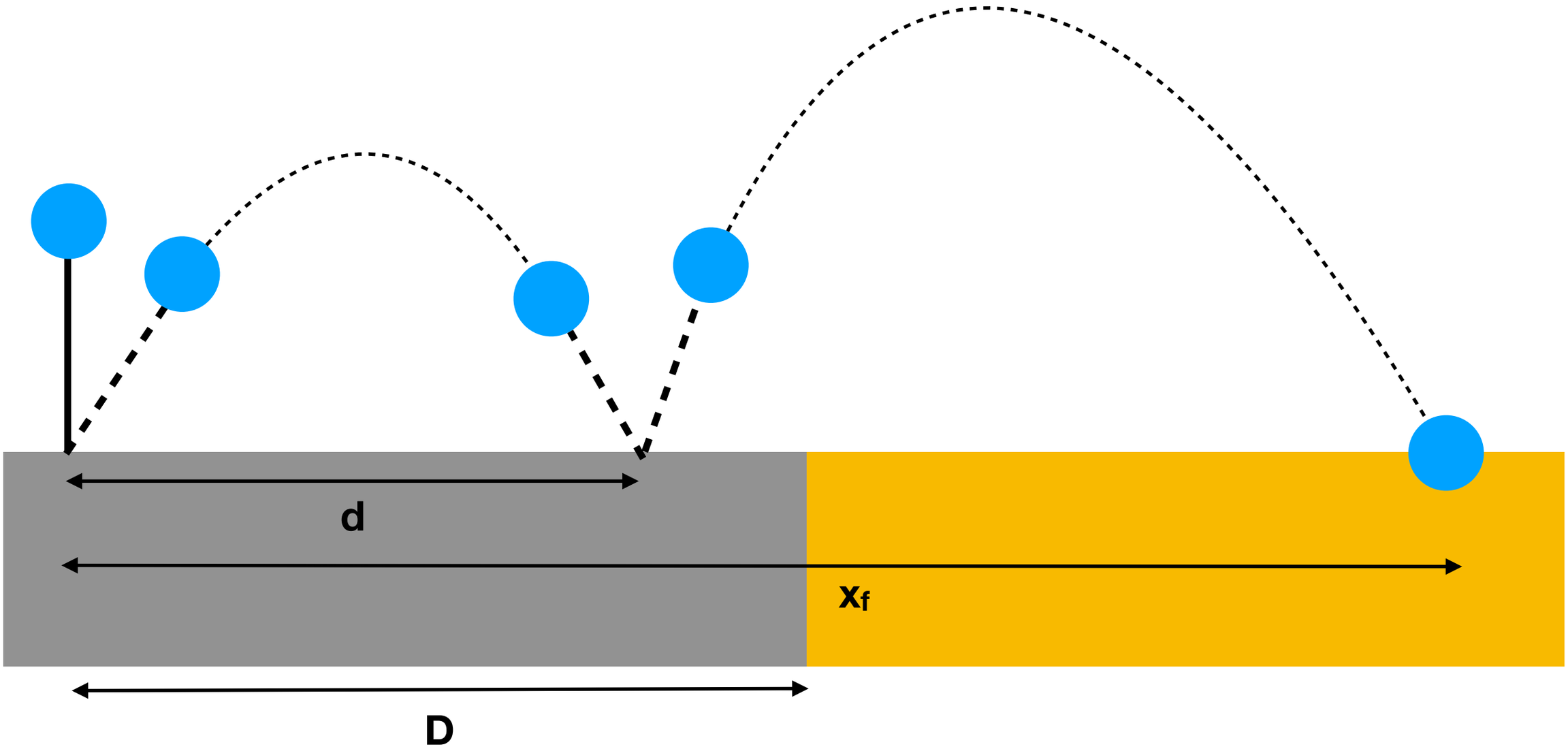
- Given D, Fmax, lmin, lmax, write the variable bounds for this problem in `longjump.m`

- Assume that the second step is renormalized, so that the contact occurs at $x_2 = y_2 = 0$



Is *d* a decision variable?
Is it a state, control, or something else?

d* is a *parameter!
Let's fill in the bounds



What are the appropriate endpoint conditions at landing, after the first ballistic phase?

Fill in the endpoint function

Software -- Trajectory Optimization

GPOPS II **http://www.gpops2.com/**

--> Matlab library -- $30/yr -- closed source
--> all of the bells and whistles

TrajOpt **https://github.com/MatthewPeterKelly/TrajOpt**

--> Matlab library -- open-source
--> direct collocation, multiple shooting, orthogonal collocation

PSOPT **http://www.psopt.org/**

--> C++ -- open source
--> all of the bells and whistles

Software -- Nonlinear Programming

FMINCON

--> Matlab optimization toolbox

IPOPT

--> Open source, C++

SNOPT

--> Industry Standard
--> Cornell MAE has a department license

^ From Matt Kelly's website

Also

- Betts 2010: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming <- The Bible of Trajectory Optimization

- matthewpeterkelly.com <- Excellent resource with interactive tutorials

- Matt Kelly 2017 SIAM Review <- Accessible tutorial / beginner's guide to direct collocation

- Manoj Srinivasan: Tutorial slides from Dynamic Walking 2010