Planning and control

A three-part story

Action plan

- Model predictive control
  - Backprop through kinematic equation
  - Minimisation wrt the “latent”
- Truck backer-upper
  - Learning an emulator of the kinematics from observations
  - Training a policy (this no one made it work)
- PPUU
  - Stochastic environment
  - Uncertainty minimisation
  - Latent decoupling

State transition equations

Evolution of the state

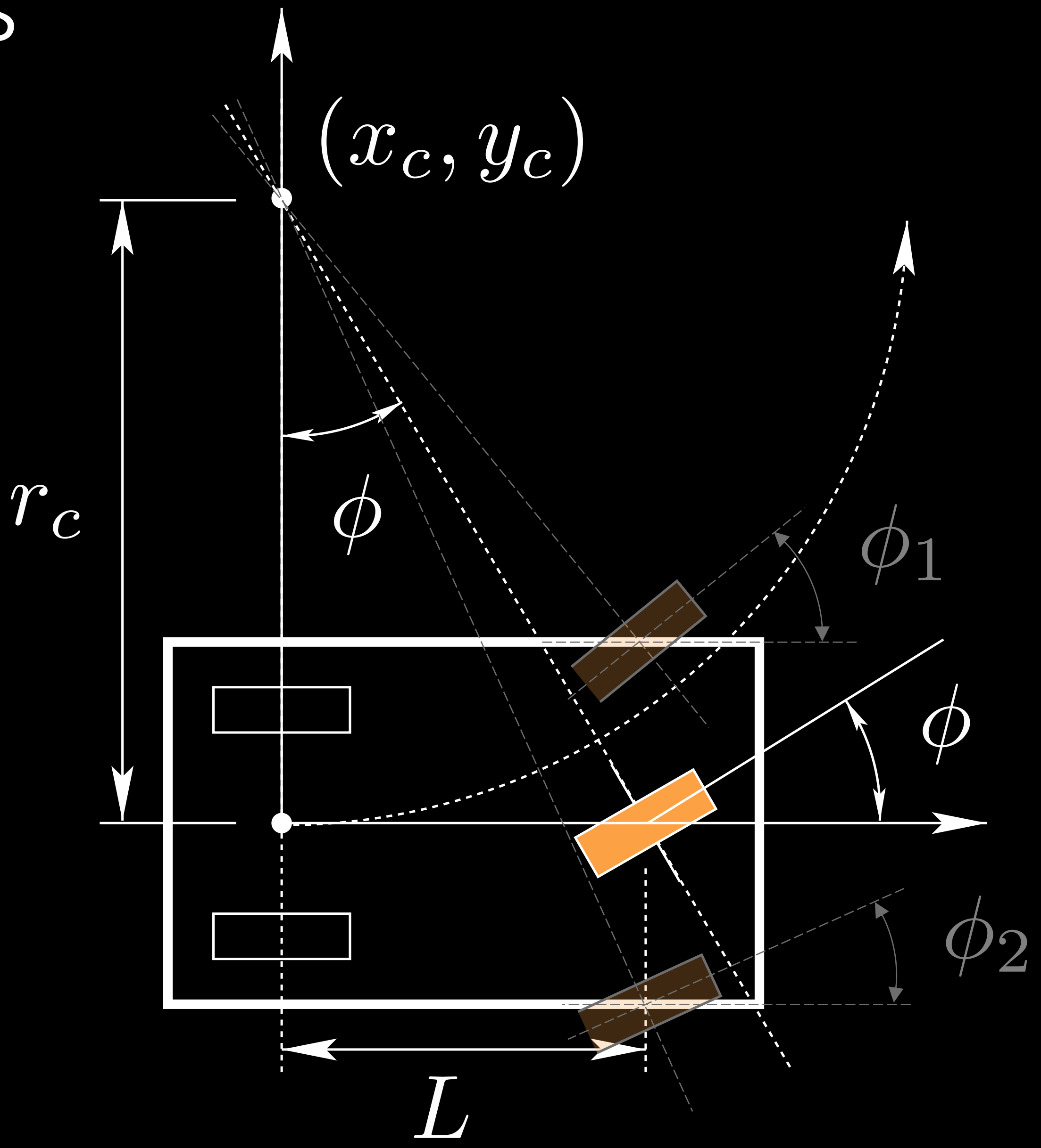
$$\boldsymbol{x} = (x \; y \; \theta \; s) \qquad \boldsymbol{u} = (\phi \; a)$$

State transition equations

$$\dot{\boldsymbol{x}} = f(\underset{\text{state}}{\boldsymbol{x}}, \underset{\text{control}}{\boldsymbol{u}}) \qquad r_c = L / \tan\phi$$

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = f(\boldsymbol{x}(t), \boldsymbol{u}(t))$$

$$\begin{cases} \dot{x} = s \cos\theta \\ \dot{y} = s \sin\theta \\ \dot{\theta} = \frac{s}{L} \tan\phi \\ \dot{s} = a \end{cases}$$



$$\boldsymbol{x} = (x\ y\ \theta\ s) \qquad \boldsymbol{u} = (\phi\ a)$$

State transition equations

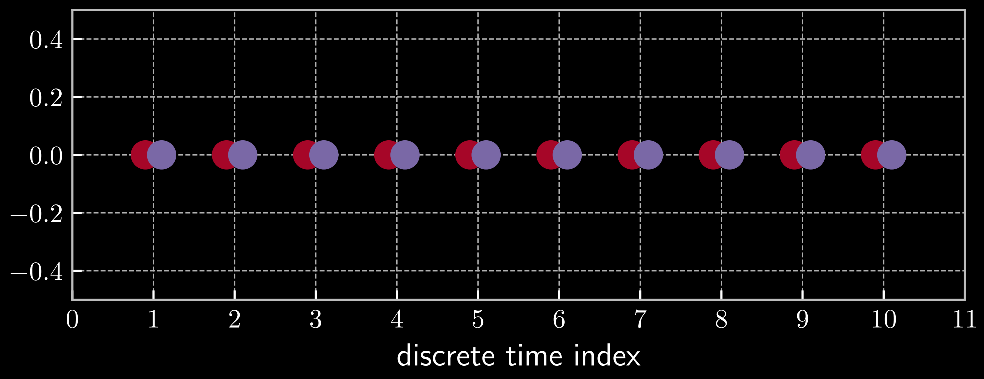
differential equation

difference equation

$$\dot{\boldsymbol{x}} = f(\underset{\text{state}}{\boldsymbol{x}}, \underset{\text{control}}{\boldsymbol{u}}) \qquad \boldsymbol{x}[n] = \boldsymbol{x}[n-1] + f(\boldsymbol{x}[n-1], \boldsymbol{u}[n])$$

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \qquad \boldsymbol{x}[n] \doteq \boldsymbol{x}(nT),\ \boldsymbol{u}[n] \doteq \boldsymbol{u}(nT)$$

$$\begin{cases} \dot{x} = s\cos\theta \\ \dot{y} = s\sin\theta \\ \dot{\theta} = \frac{s}{L}\tan\phi \\ \dot{s} = a \end{cases} \qquad \begin{cases} x[n] = x[n-1] + s\cos\theta[n-1] \\ y[n] = y[n-1] + s\sin\theta[n-1] \\ \theta[n] = \theta[n-1] + \frac{s}{L}\tan\phi[n] \\ s[n] = s[n-1] + a[n] \end{cases}$$



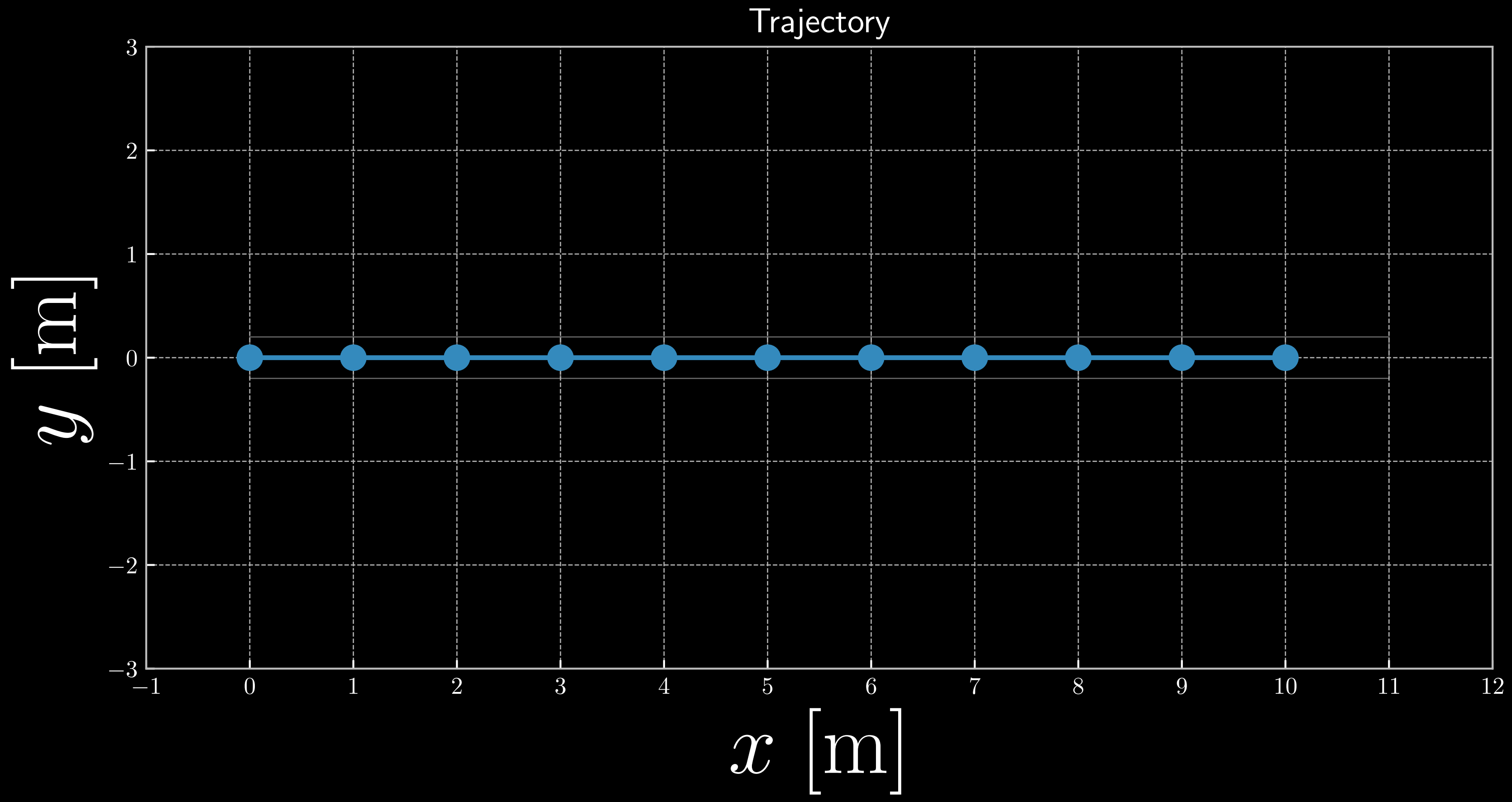
$$[\boldsymbol{u}] = (\mathrm{rad}\ \frac{\mathrm{m}}{\mathrm{s}^2})$$

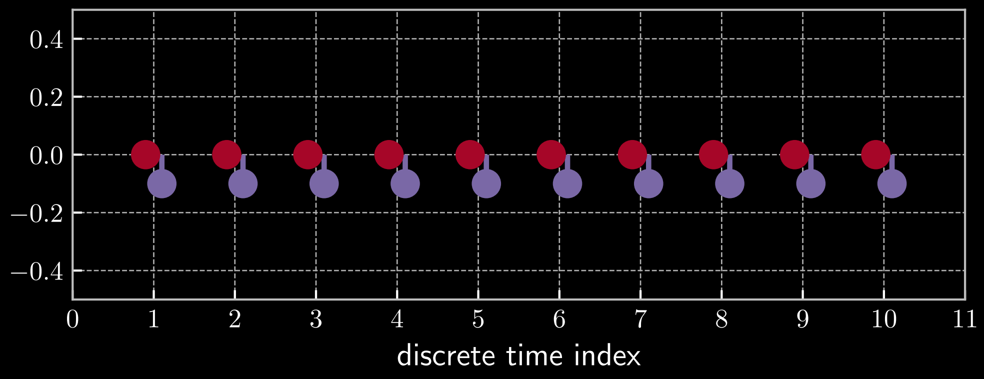
$$[\boldsymbol{x}] = (\mathrm{m}\ \mathrm{m}\ \mathrm{rad}\ \frac{\mathrm{m}}{\mathrm{s}})$$

$$\boldsymbol{u} = (\phi\ a)$$

$$\boldsymbol{x} = (x\ y\ \theta\ s)$$

$$\boldsymbol{x}_0 \doteq (0\ 0\ 0\ 1)$$





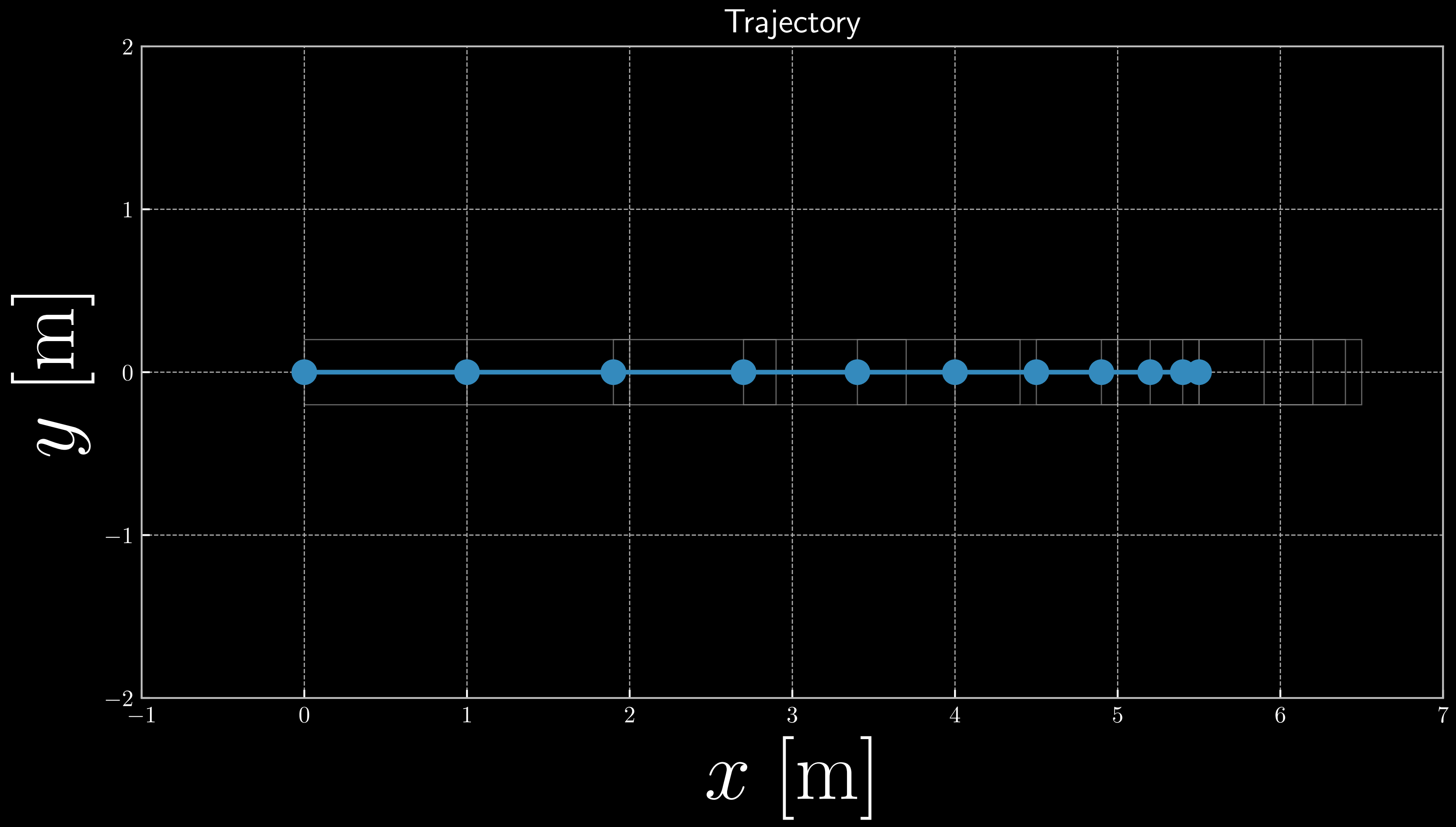
$$[\boldsymbol{u}] = (\mathrm{rad}\ \tfrac{\mathrm{m}}{\mathrm{s}^2})$$

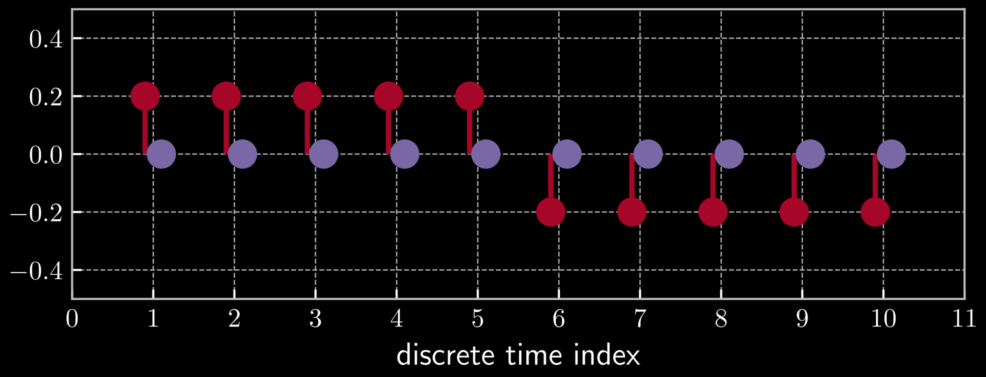
$$[\boldsymbol{x}] = (\mathrm{m}\ \mathrm{m}\ \mathrm{rad}\ \tfrac{\mathrm{m}}{\mathrm{s}})$$

$$\boldsymbol{u} = (\phi\ a)$$

$$\boldsymbol{x} = (x\ y\ \theta\ s)$$

$$\boldsymbol{x}_0 \doteq (0\ 0\ 0\ 1)$$





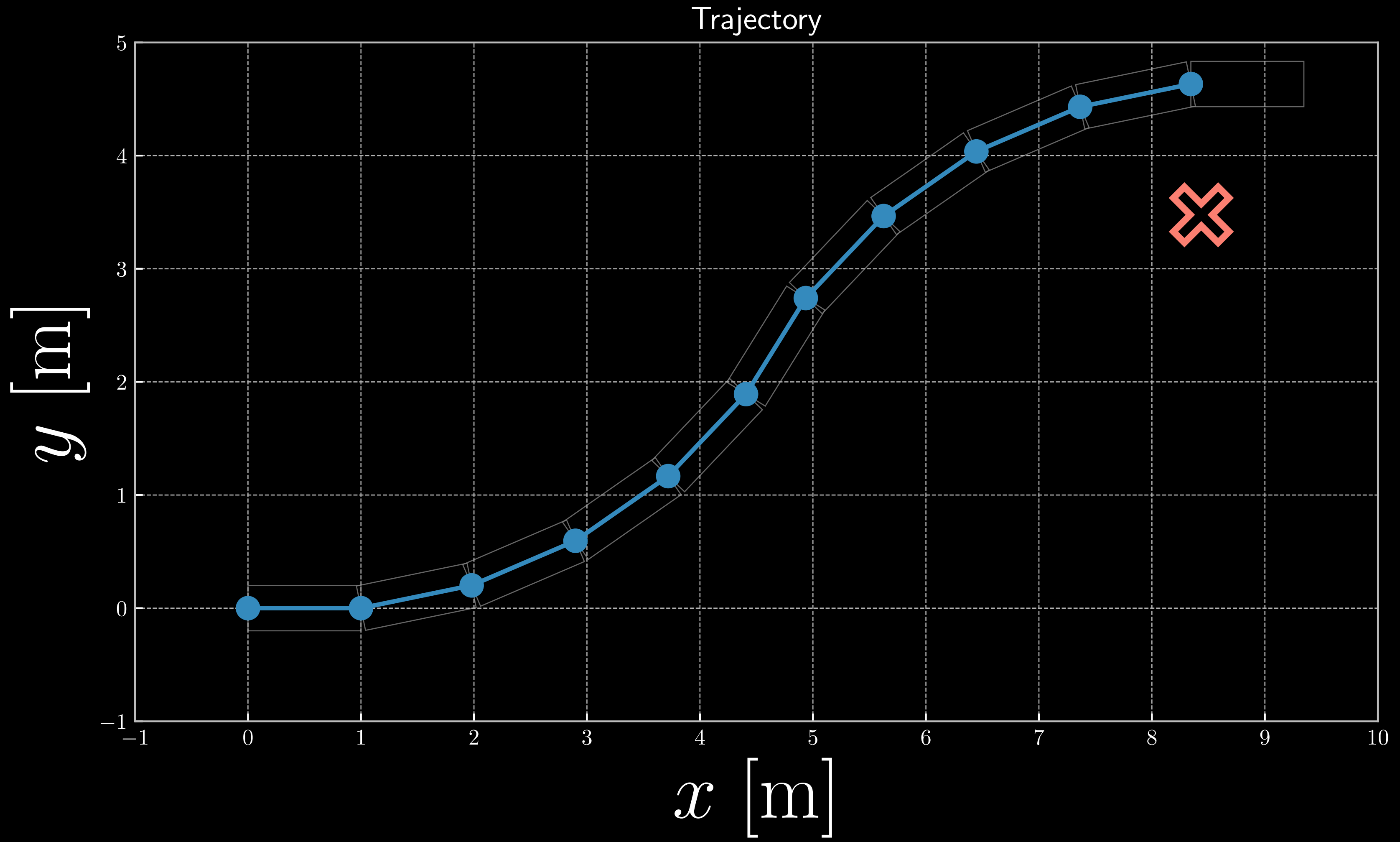
$$[\boldsymbol{u}] = (\mathrm{rad}\ \tfrac{\mathrm{m}}{\mathrm{s}^2})$$

$$[\boldsymbol{x}] = (\mathrm{m}\ \mathrm{m}\ \mathrm{rad}\ \tfrac{\mathrm{m}}{\mathrm{s}})$$

$$\boldsymbol{u} = (\phi\ a)$$

$$\boldsymbol{x} = (x\ y\ \theta\ s)$$

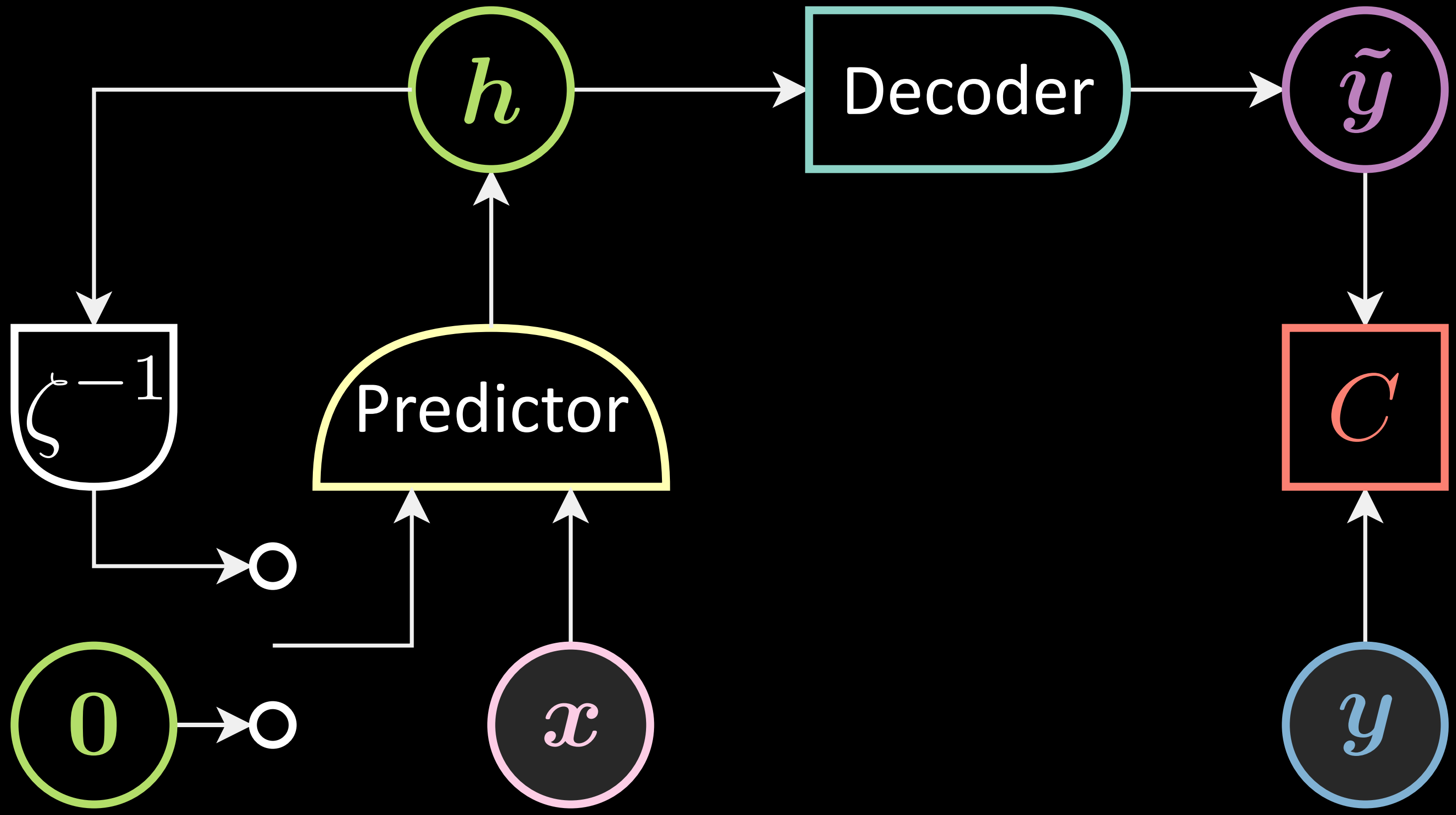
$$\boldsymbol{x}_0 \doteq (0\ 0\ 0\ 1)$$



Kelley-Bryson algorithm

Backprop through time + gradient descent

RNN recap



RNN equations

$$h[0] \doteq \mathbf{0}$$

$$h[t] = \mathrm{Pred}(h[t-1], x[t])$$

$$\tilde{y}[t] = \mathrm{Dec}(h[t])$$

RNN training

- backprop through time
- SGD wrt predictor's params to match ***x*** and ***y***

Control

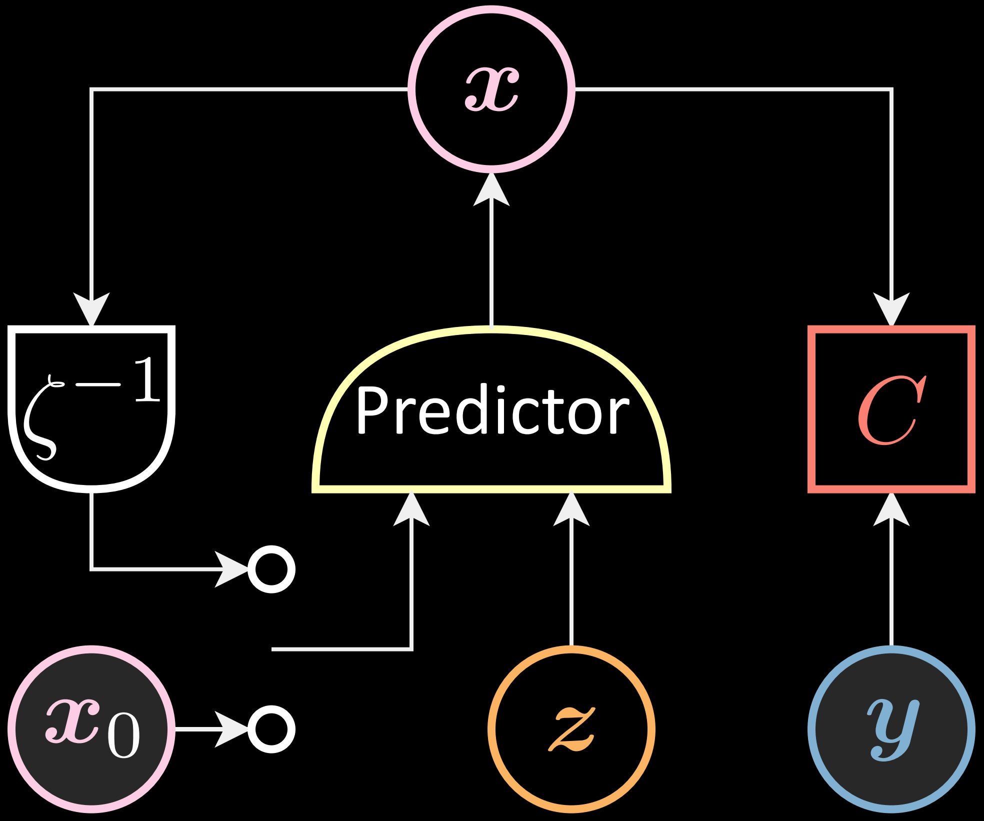
Optimal control (inference)

- backprop through time
- GD wrt $\boldsymbol{z}$ to go from $\boldsymbol{x}_0$ to $\boldsymbol{y}$

$$\boldsymbol{x}[0] \doteq \boldsymbol{x}_0$$

$$\boldsymbol{x}[t] = \text{Pred}(\boldsymbol{x}[t-1], \boldsymbol{z}[t])$$

control $\boldsymbol{u}[t]$



RNN equations

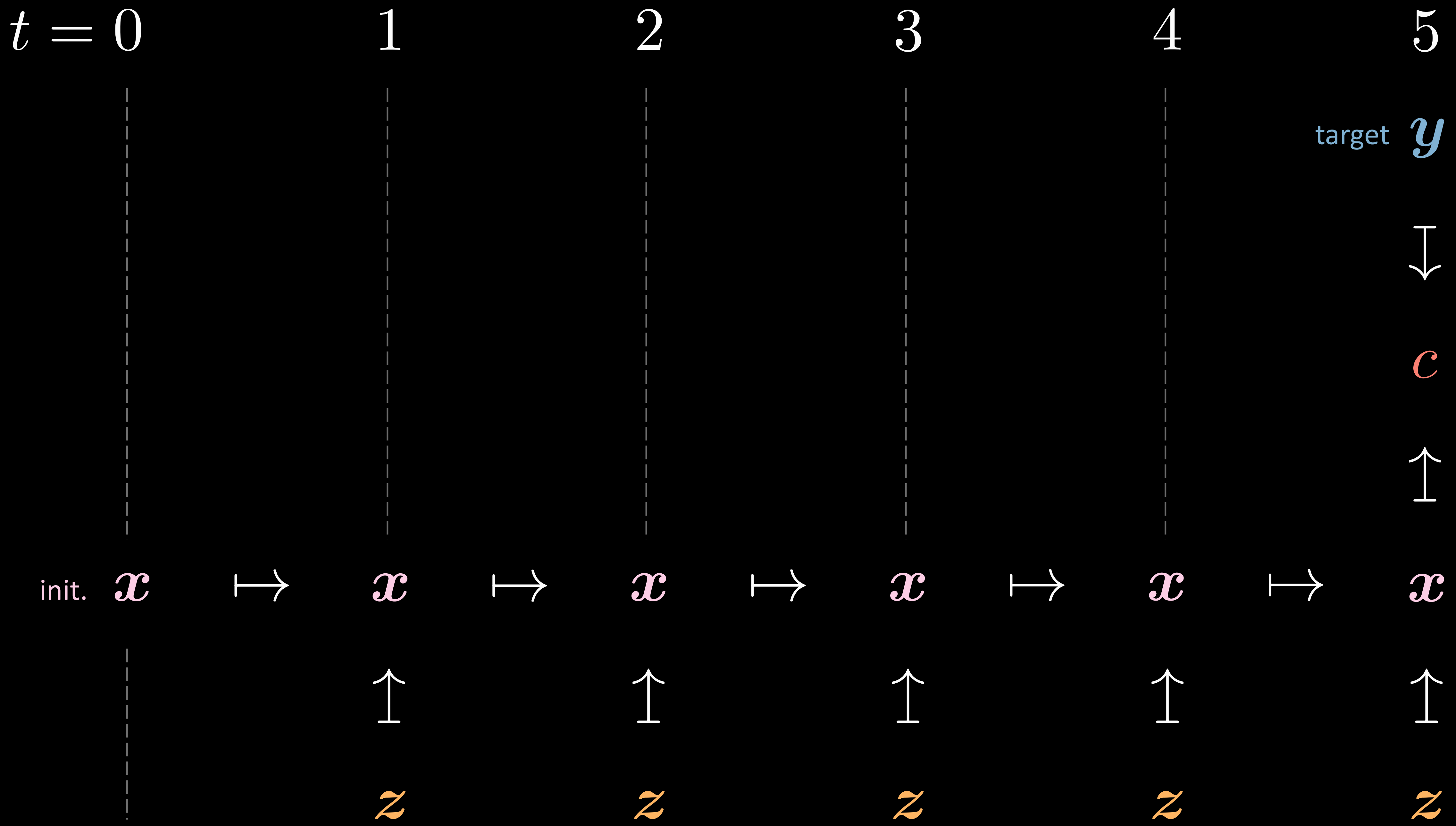
$$\boldsymbol{h}[0] \doteq \boldsymbol{0}$$

$$\boldsymbol{h}[t] = \text{Pred}(\boldsymbol{h}[t-1], \boldsymbol{x}[t])$$

$$\tilde{\boldsymbol{y}}[t] = \text{Dec}(\boldsymbol{h}[t])$$

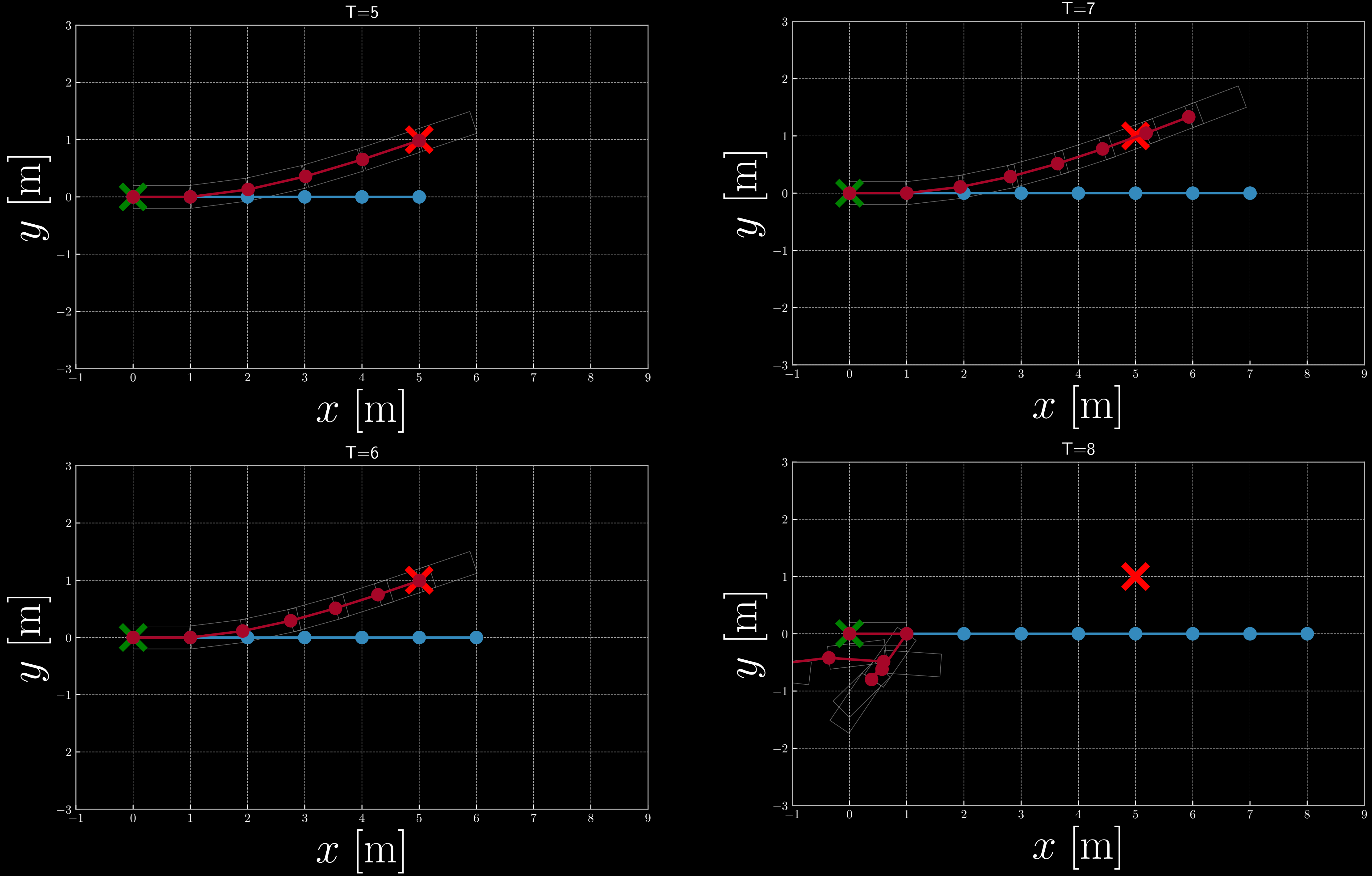
RNN training

- backprop through time
- SGD wrt predictor's params to match $\boldsymbol{x}$ and $\boldsymbol{y}$

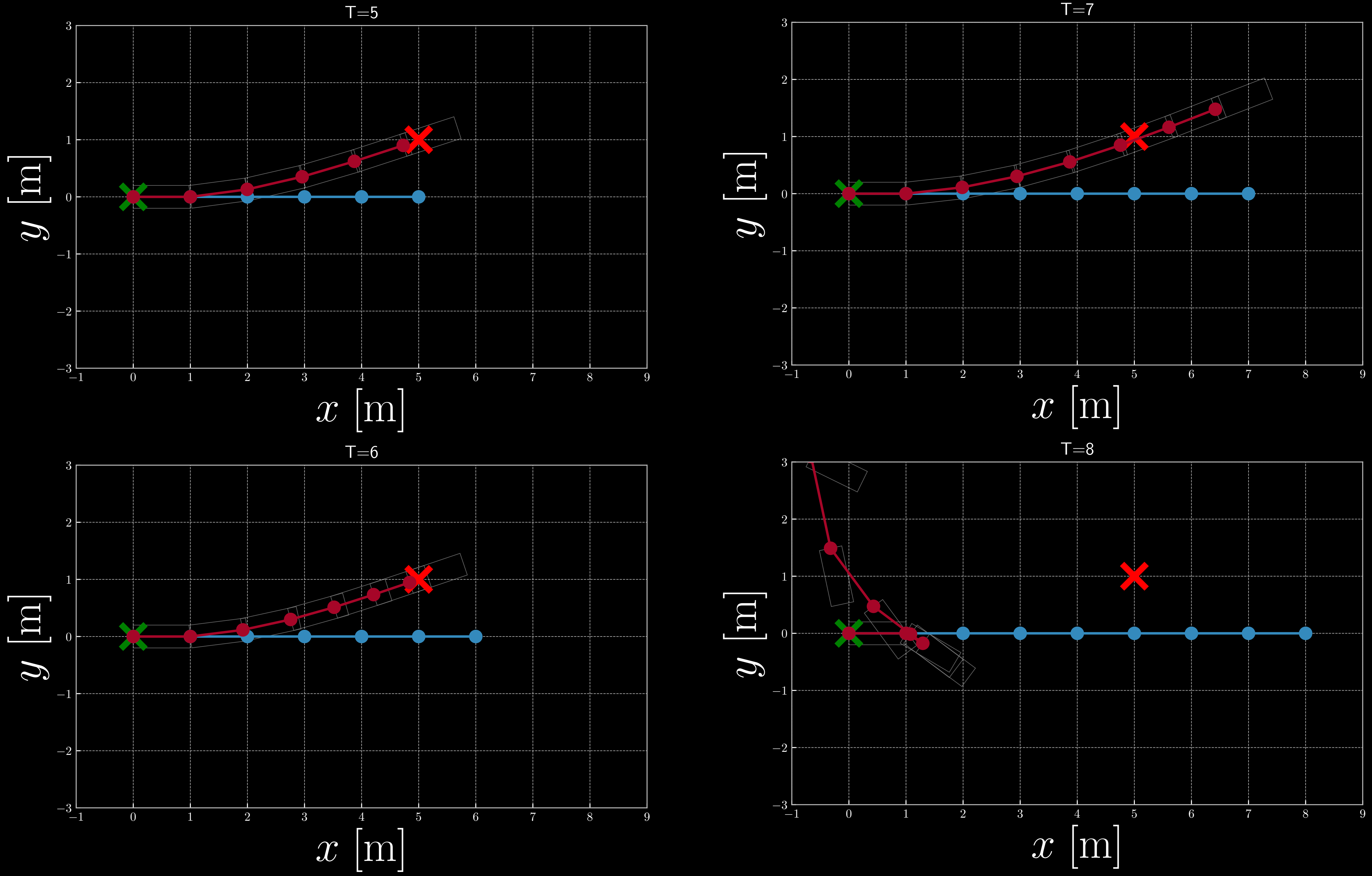


$t = 0$
1
2
3
4
5
target y
c
init. x
x
x
x
x
x
z
z
z
z
z

Final position only

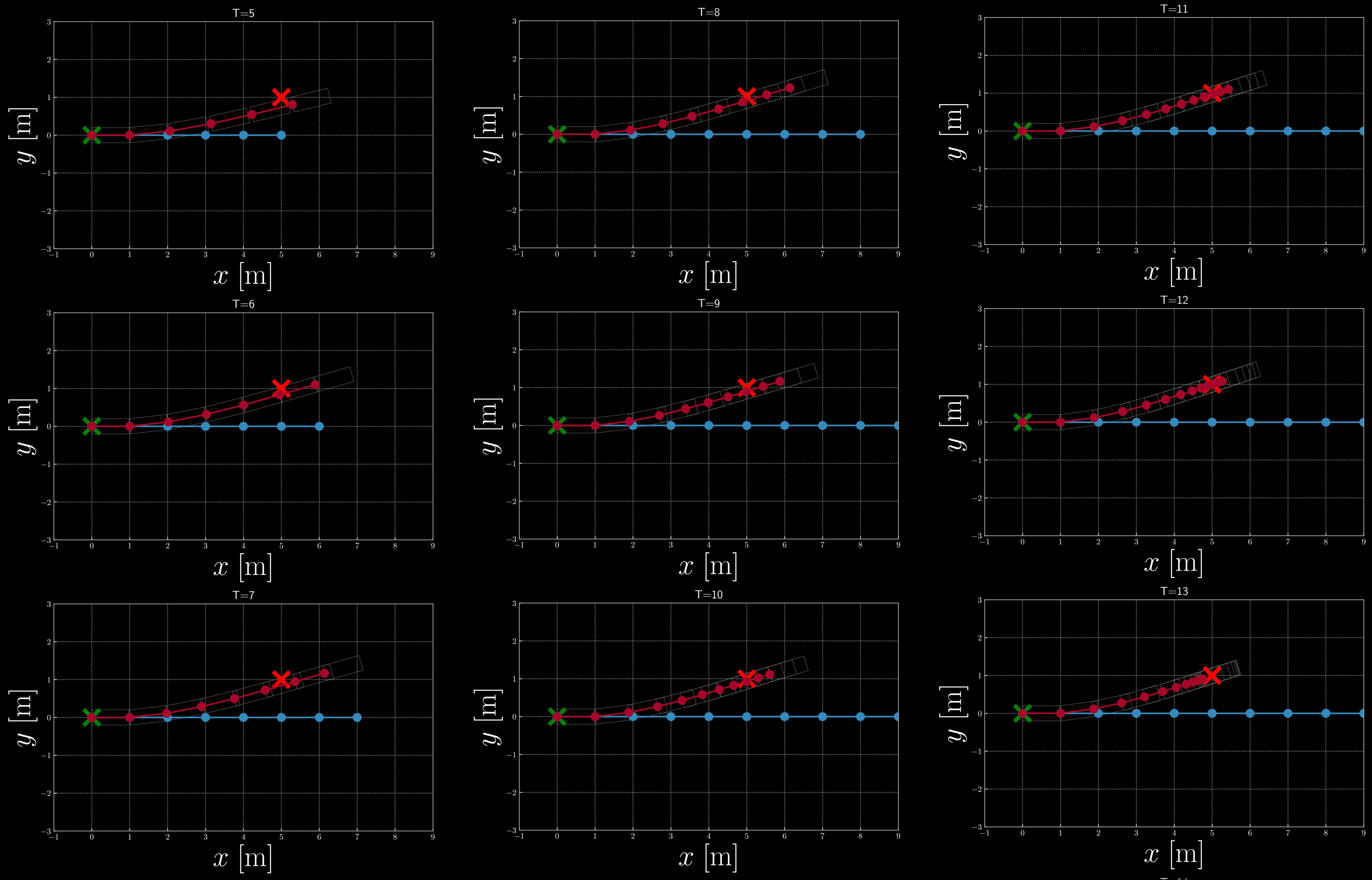


Final position only and zero speed



$$\begin{tikzcd}
t = 0 & & 1 & & 2 & & 3 & & 4 & & 5 \\
 & & y \arrow[d, mapsto] & & y \arrow[d, mapsto] & & y \arrow[d, mapsto] & & y \arrow[d, mapsto] & & y \arrow[d, mapsto] \\
 & & c & & c & & c & & c & & c \\
x \arrow[rr, mapsto] & & x \arrow[u, mapsto] \arrow[rr, mapsto] & & x \arrow[u, mapsto] \arrow[rr, mapsto] & & x \arrow[u, mapsto] \arrow[rr, mapsto] & & x \arrow[u, mapsto] \arrow[rr, mapsto] & & x \arrow[u, mapsto] \\
 & & z \arrow[u, mapsto] & & z \arrow[u, mapsto] & & z \arrow[u, mapsto] & & z \arrow[u, mapsto] & & z \arrow[u, mapsto]
\end{tikzcd}$$

Average distance



Softmin