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

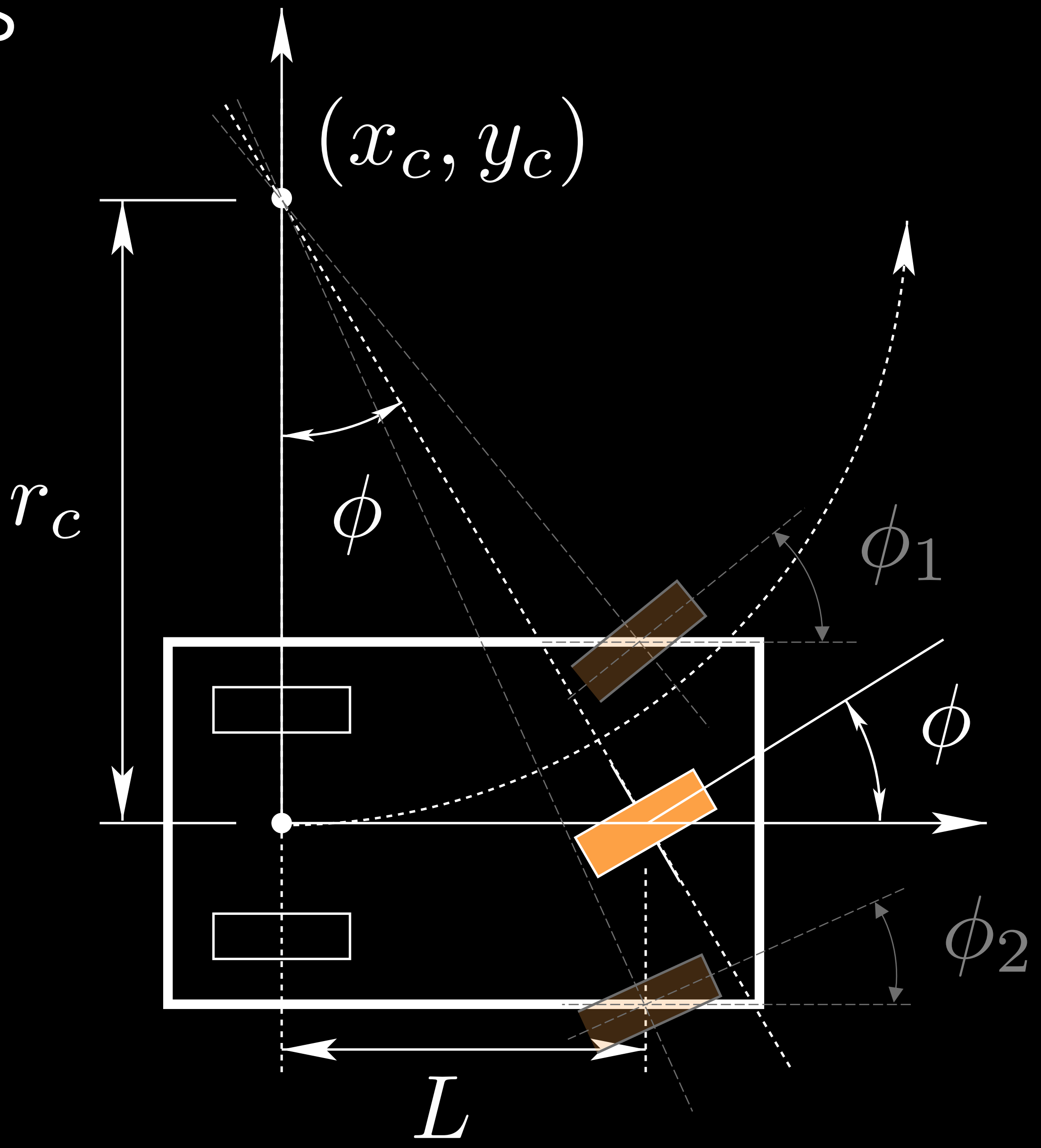
$$\boldsymbol{x} = (x \; y \; \theta \; s) \qquad \boldsymbol{u} = (\phi \; a)$$

State transition equations

$$\dot{\boldsymbol{x}} = f(\underbrace{\boldsymbol{x}}_{\text{state}}, \underbrace{\boldsymbol{u}}_{\text{control}})$$

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

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



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

State transition equations

differential equation

$$\dot{\boldsymbol{x}} = f(\underset{\text{state}}{\boldsymbol{x}}, \underset{\text{control}}{\boldsymbol{u}})$$

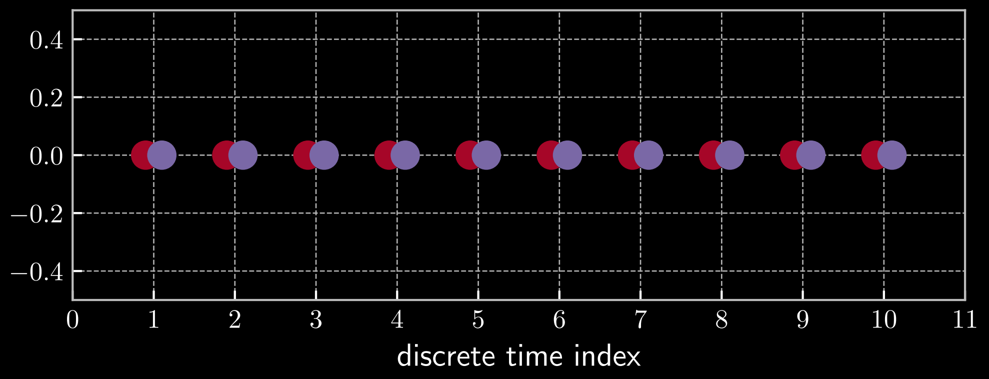
$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = f(\boldsymbol{x}(t), \boldsymbol{u}(t))$$

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

difference equation

$$\boldsymbol{x}[t] = \boldsymbol{x}[t-1] + f(\boldsymbol{x}[t-1], \boldsymbol{u}[t])\,\mathrm{d}t$$

$$\begin{cases} x[t] = x[t-1] + s\cos\theta[t-1]\,\mathrm{d}t \\ y[t] = y[t-1] + s\sin\theta[t-1]\,\mathrm{d}t \\ \theta[t] = \theta[t-1] + \frac{s}{L}\tan\phi[t]\,\mathrm{d}t \\ s[t] = s[t-1] + a[t]\,\mathrm{d}t \end{cases}$$



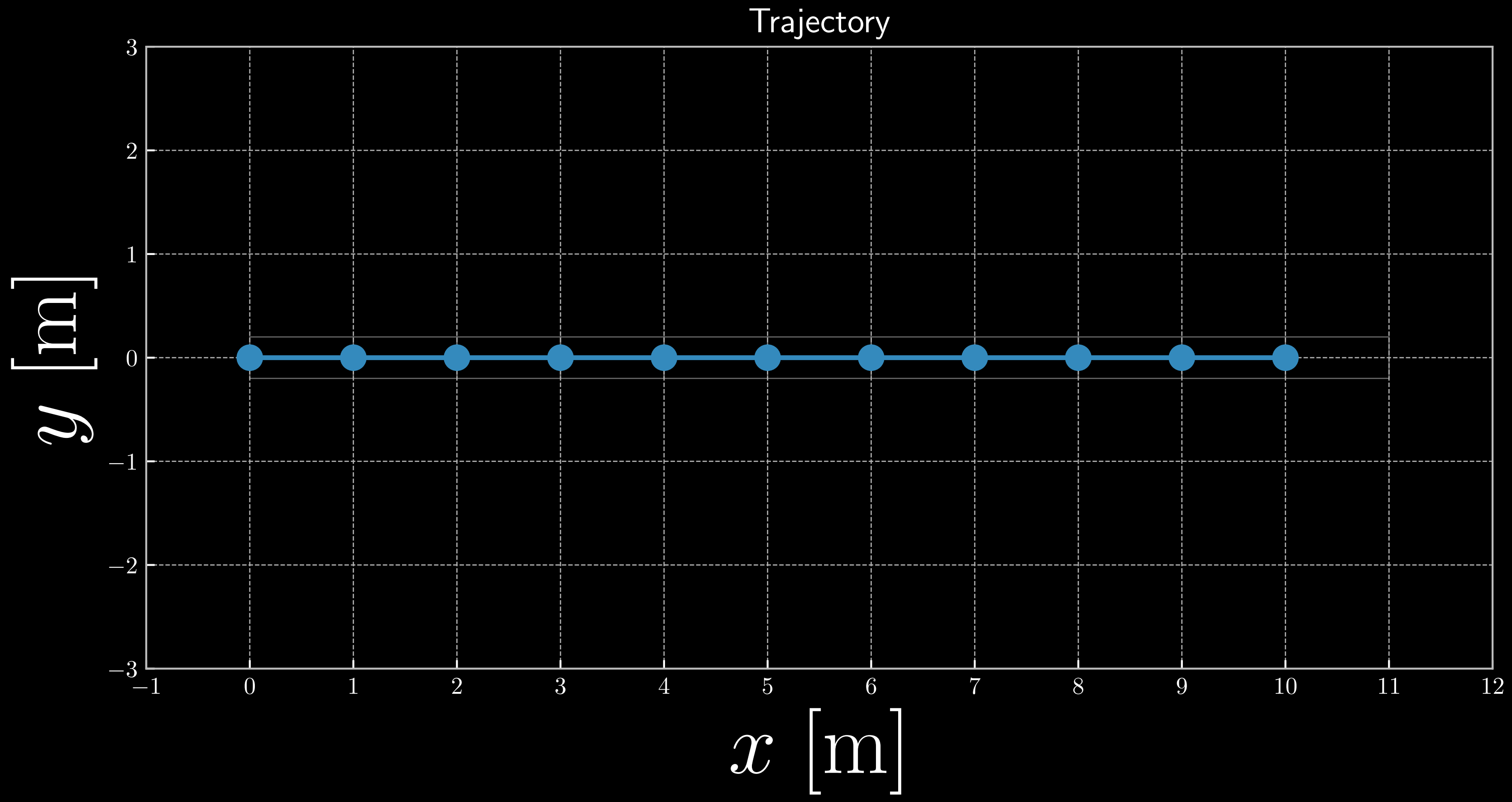
$$[\boldsymbol{u}] = (\mathrm{rad}\ \tfrac{\mathrm{m}}{\mathrm{s}^2})$$

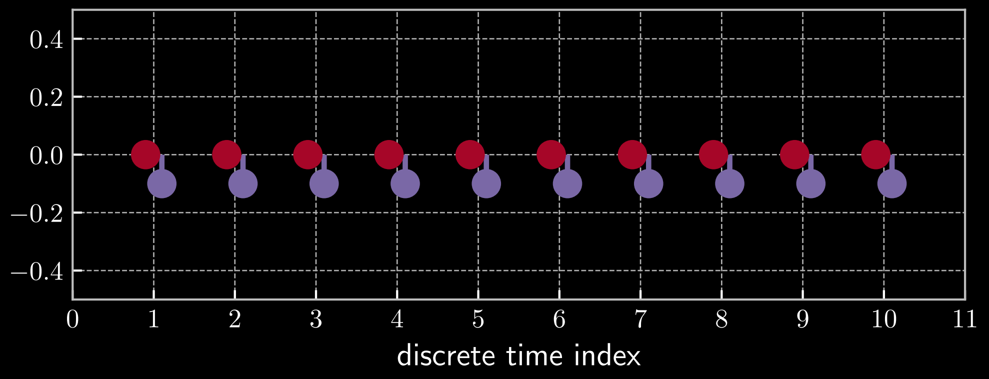
$$[\boldsymbol{x}] = (\mathrm{m}\ \mathrm{m}\ \mathrm{rad}\ \tfrac{\mathrm{m}}{\mathrm{s}})$$

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

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

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





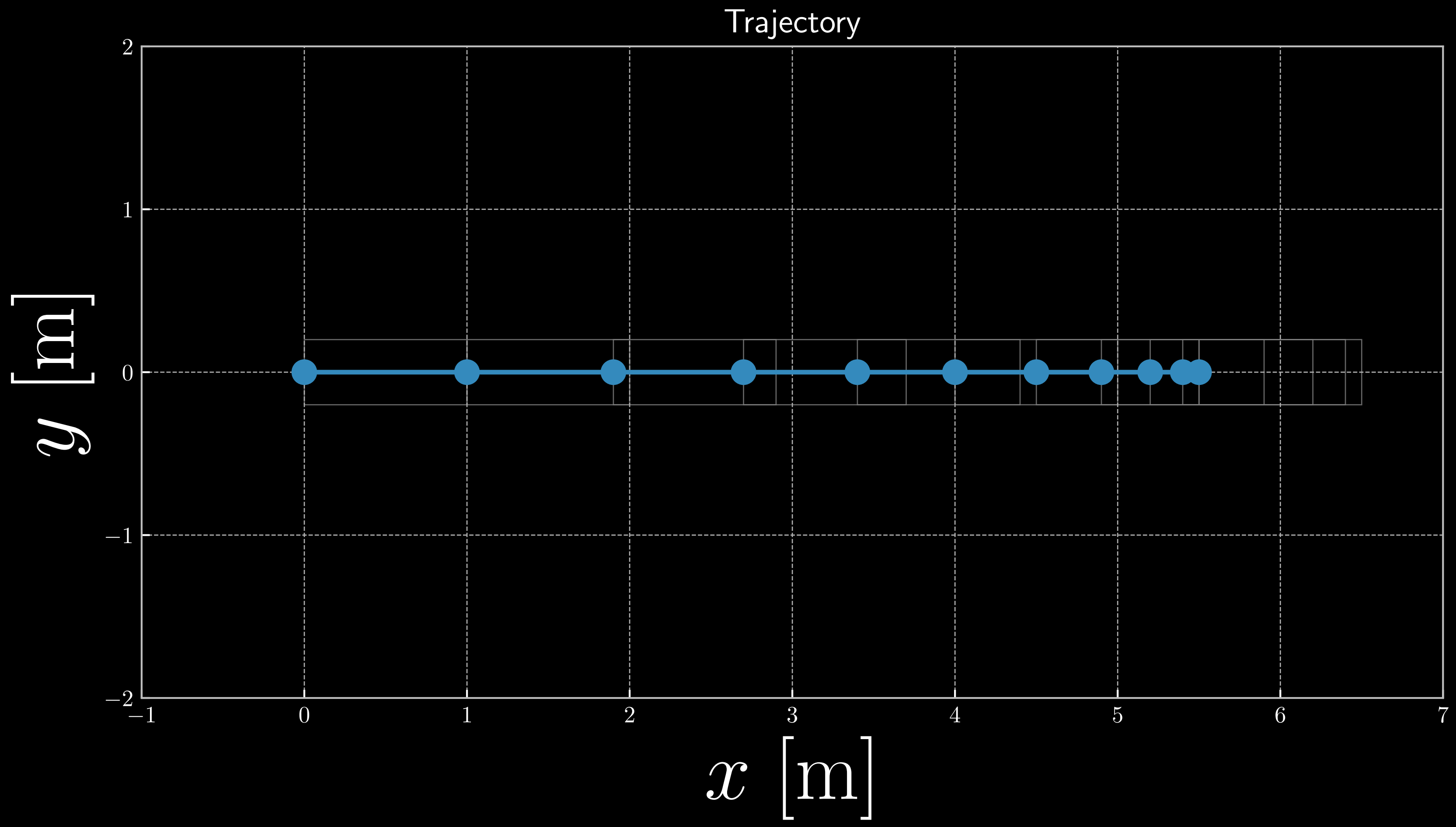
$$[\boldsymbol{u}] = \left(\mathrm{rad}\ \frac{\mathrm{m}}{\mathrm{s}^2}\right)$$

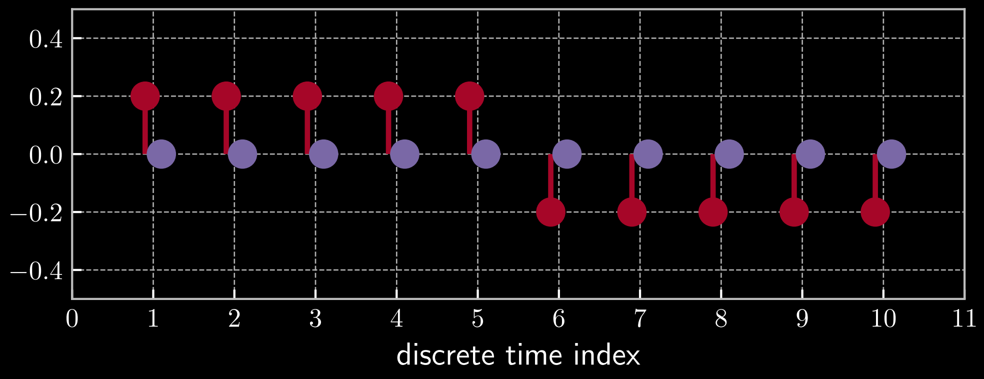
$$[\boldsymbol{x}] = \left(\mathrm{m}\ \mathrm{m}\ \mathrm{rad}\ \frac{\mathrm{m}}{\mathrm{s}}\right)$$

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

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

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





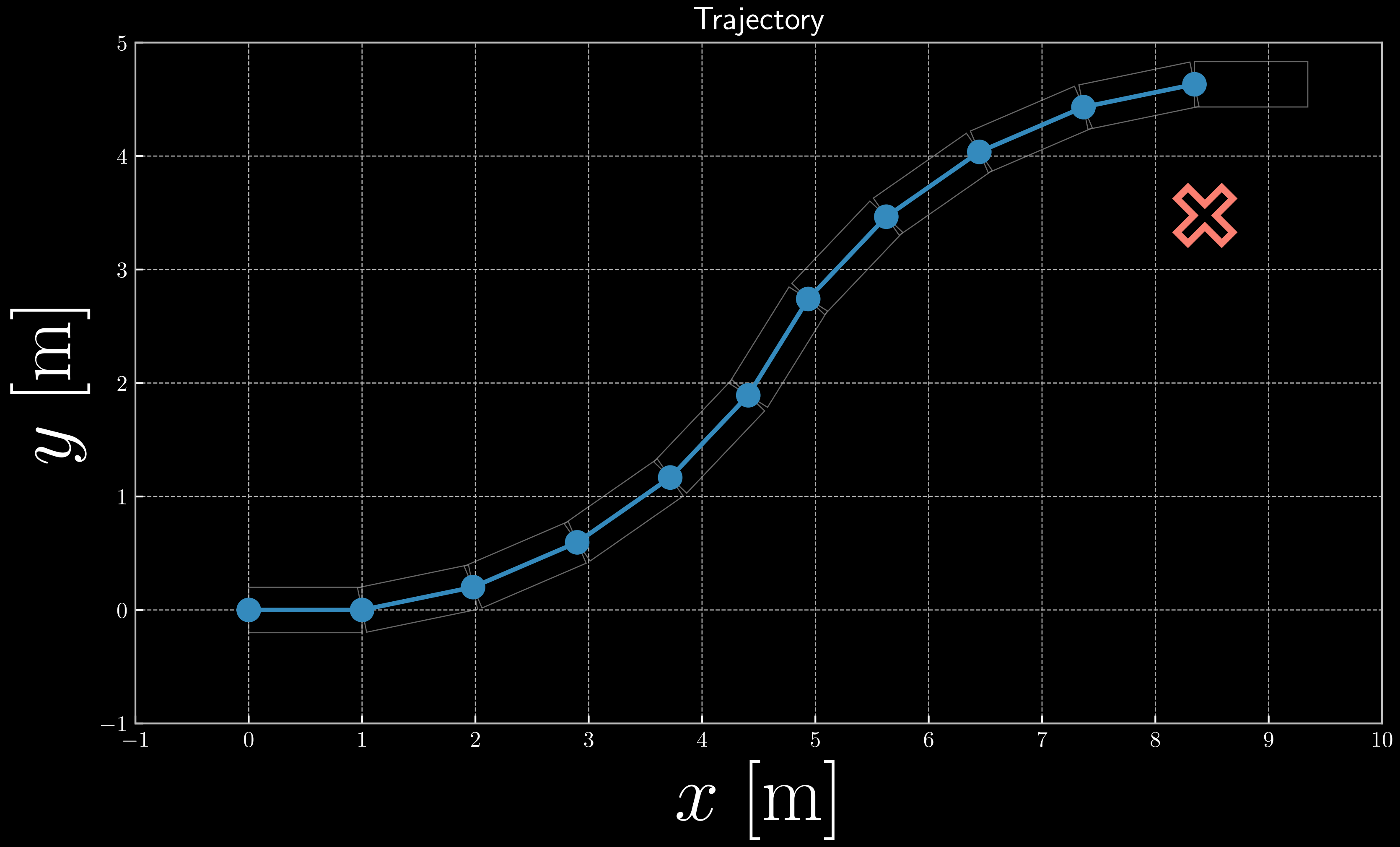
$$[\boldsymbol{u}] = (\mathrm{rad}\ \tfrac{\mathrm{m}}{\mathrm{s}^2})$$

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

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

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

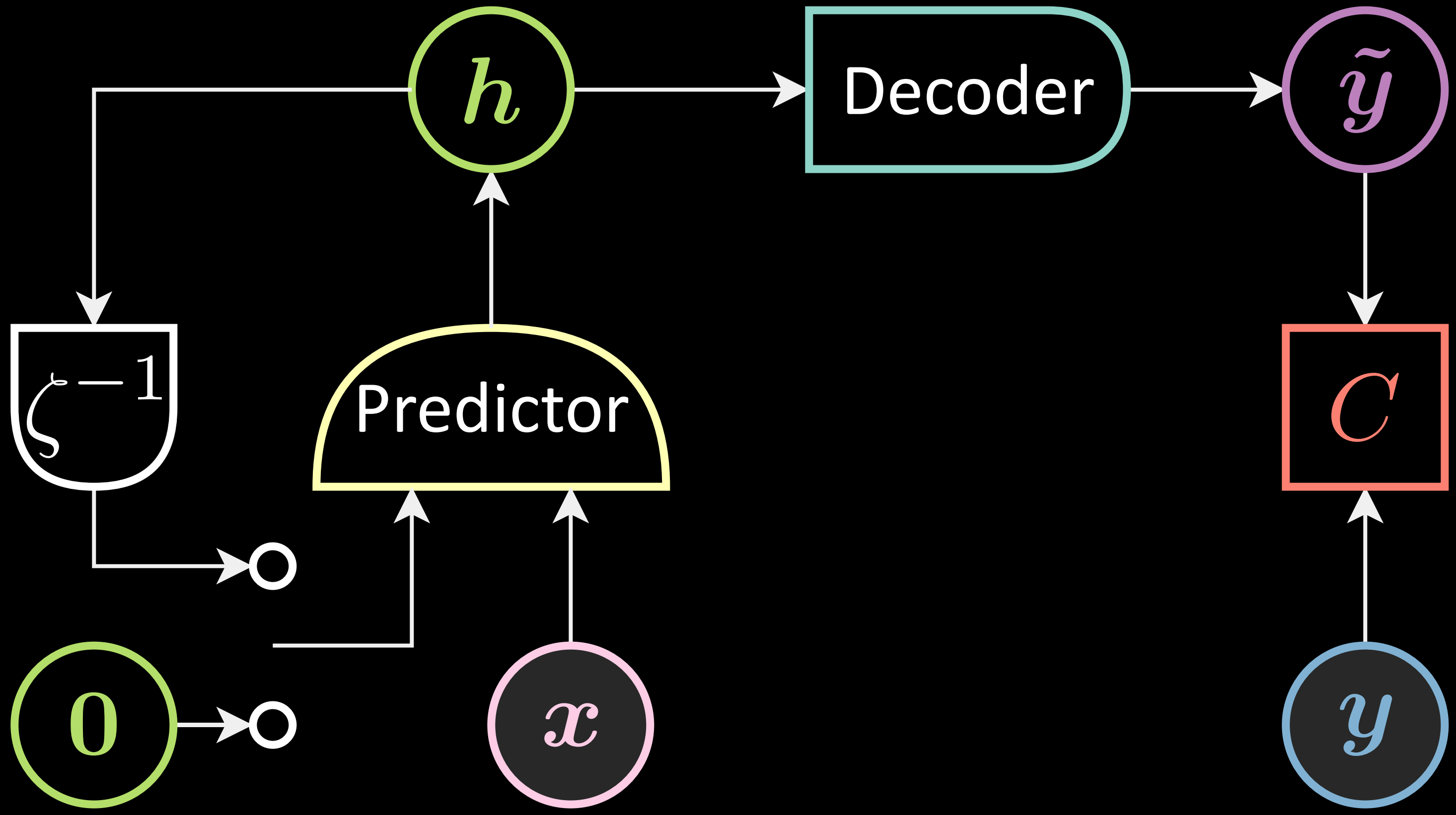
$$\boldsymbol{x}_0 \doteq (0\ 0\ 0\ 1)$$



Kelley-Bryson algorithm

Backprop through time + gradient descent

RNN recap



RNN equations

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

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

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

RNN training

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

Control

Optimal control (inference)

- backprop through time
- GD wrt ***z*** to go from ***x***$_0$ to ***y***

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

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

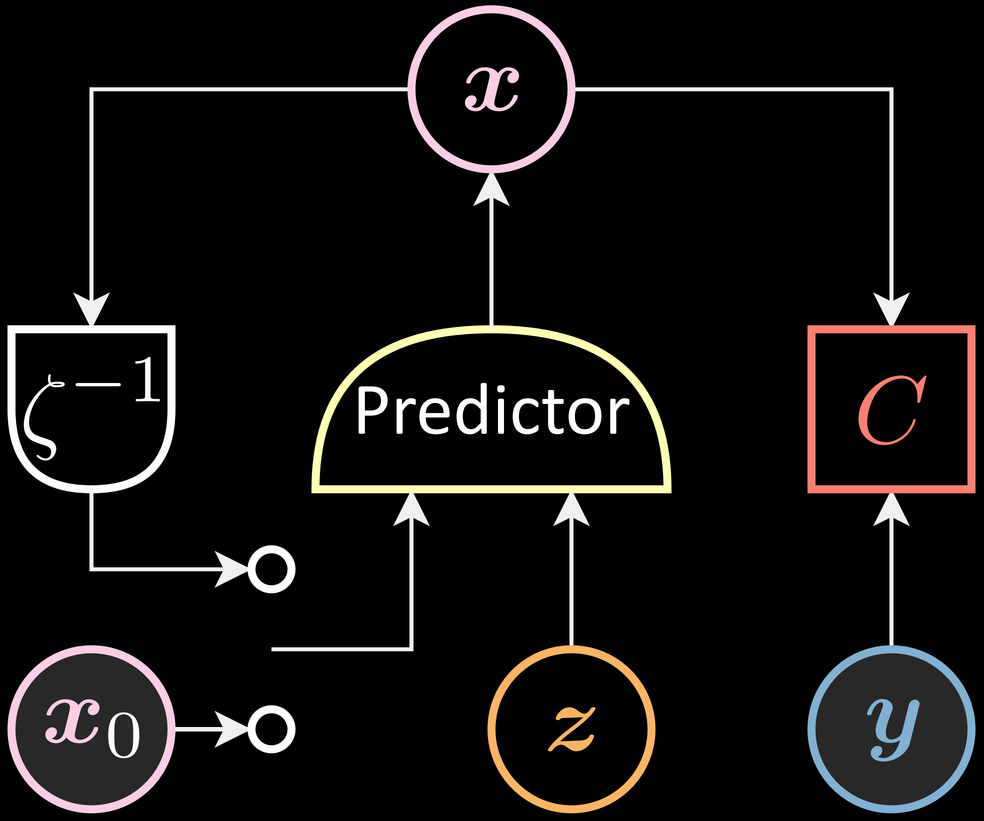
control $\boldsymbol{u}[t]$

RNN equations

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

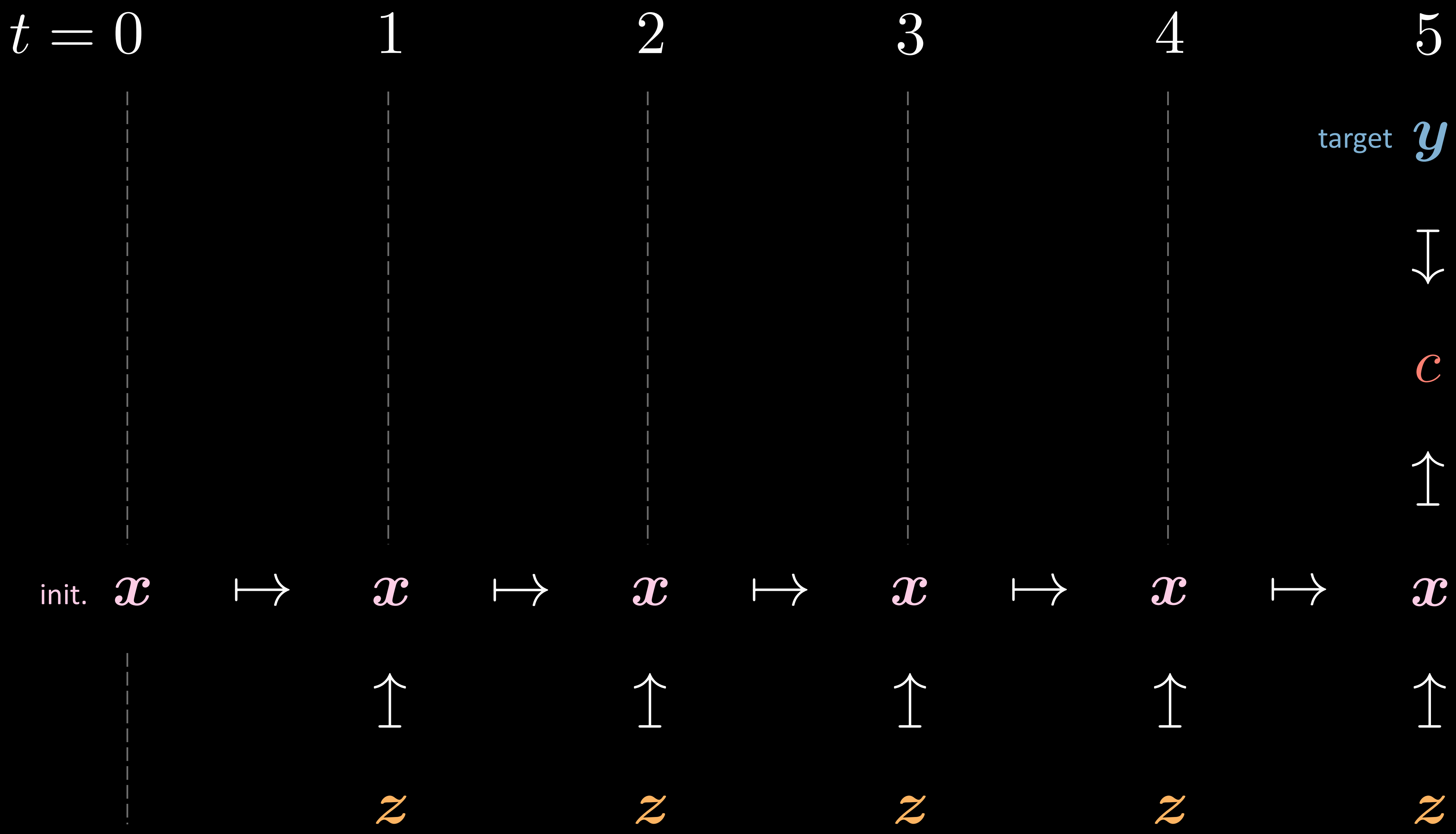
$$\boldsymbol{h}[t] = \mathrm{Pred}(\boldsymbol{h}[t-1], \boldsymbol{x}[t])$$

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



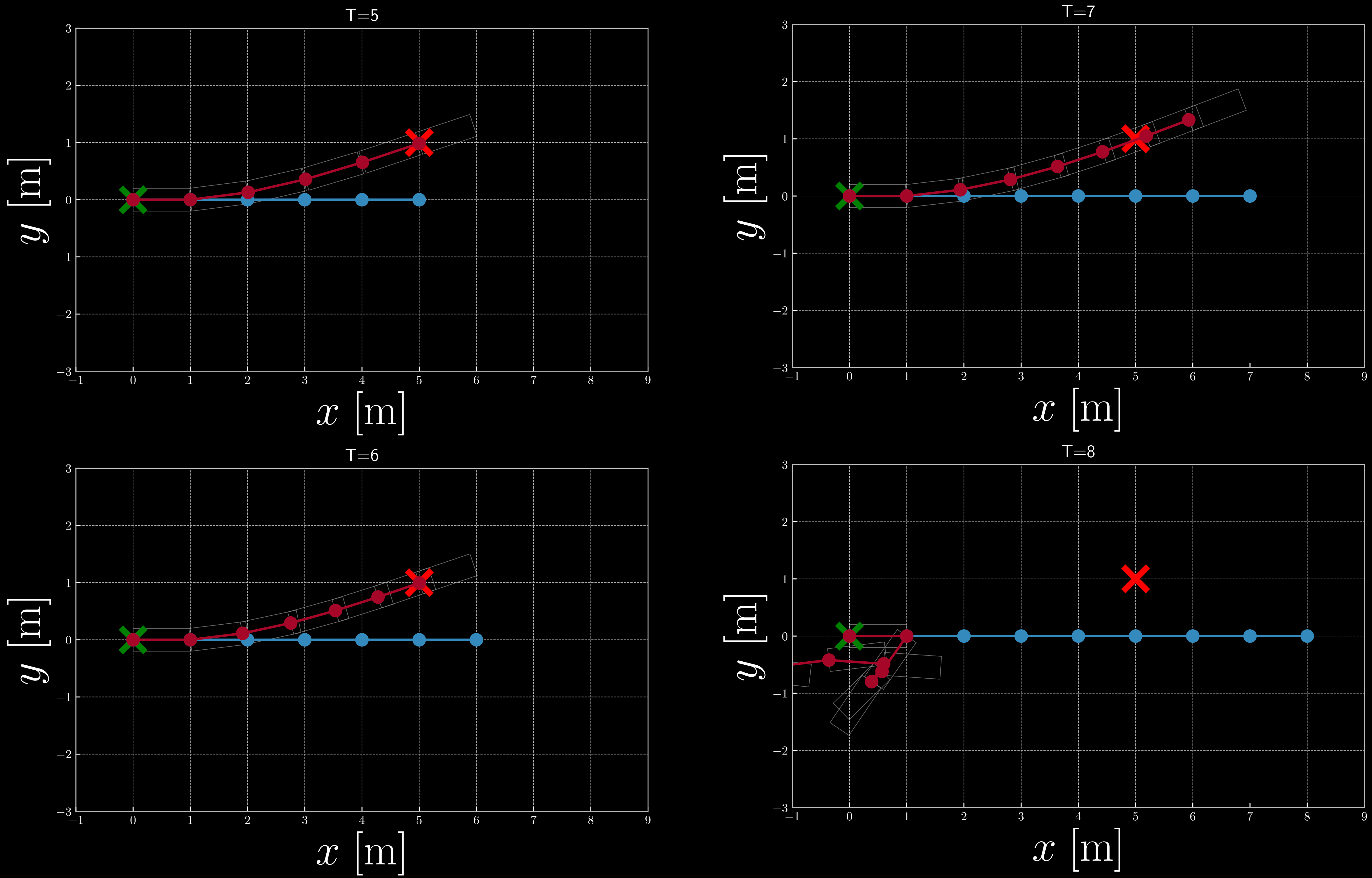
RNN training

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

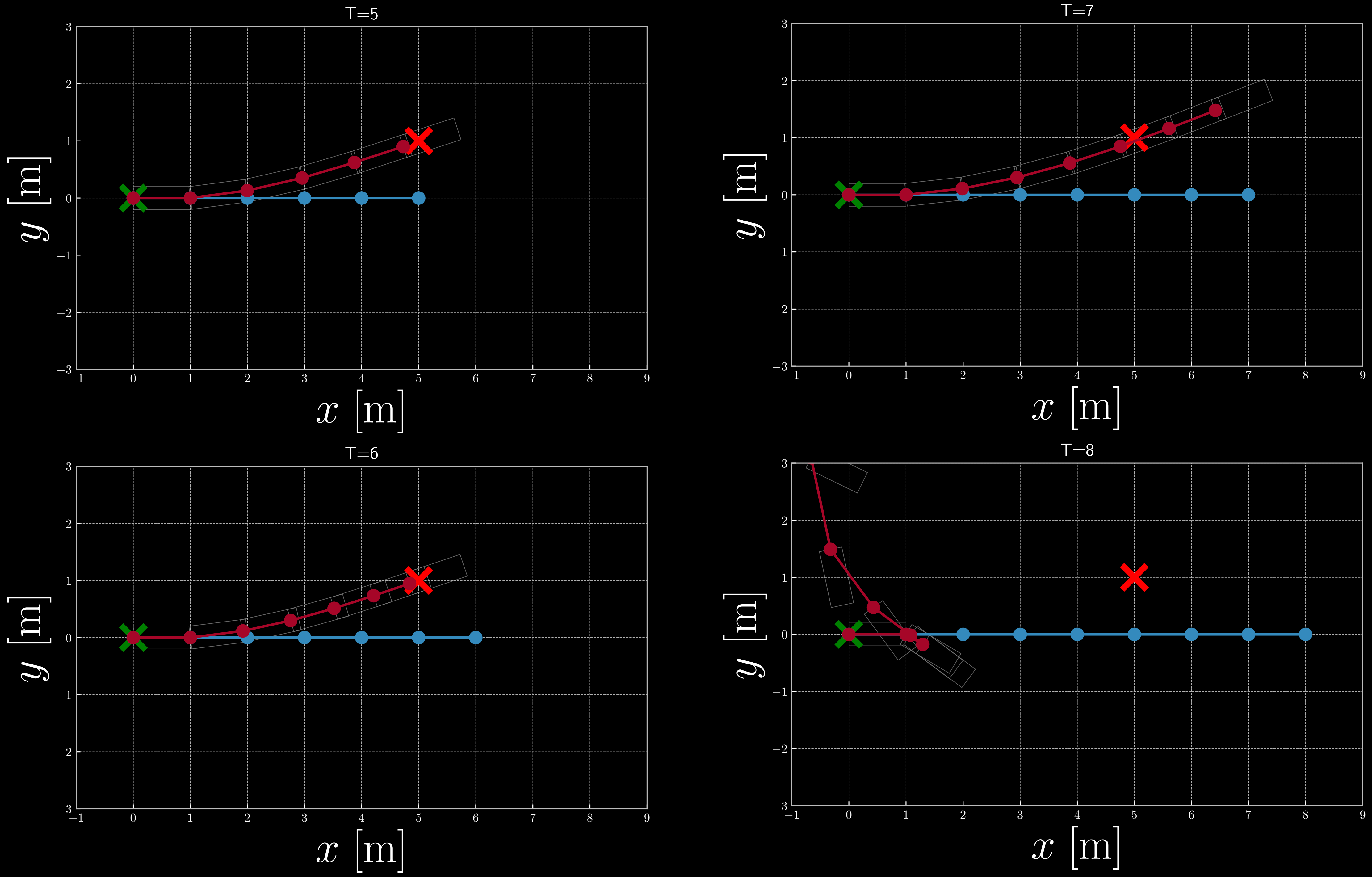


$t = 0$
1
2
3
4
5
target y
c
init. x
x
x
x
x
x
z
z
z
z
z

Final position only

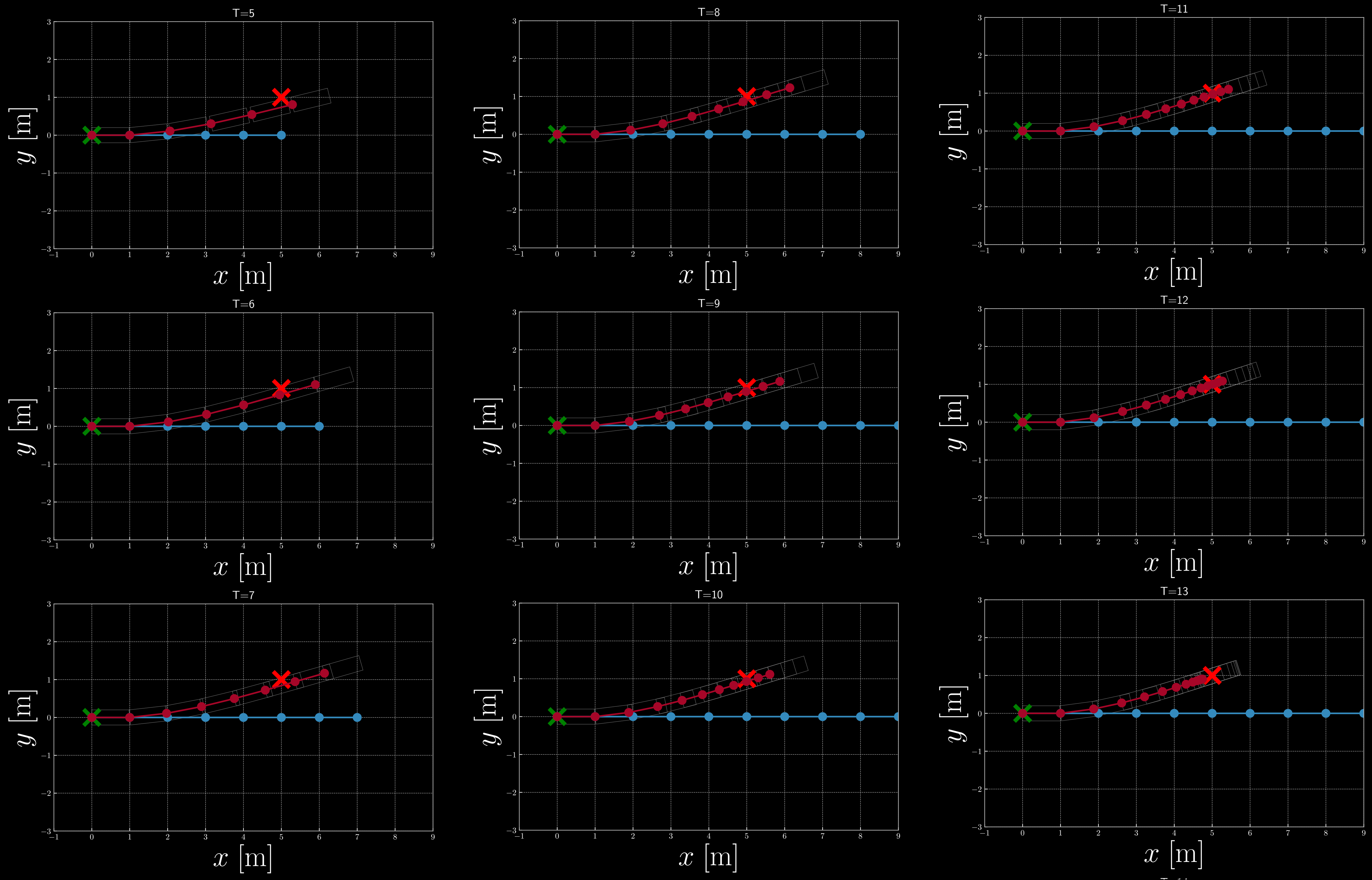


Final position only and zero speed



$$\begin{array}{ccccccccccc}
t=0 & & 1 & & 2 & & 3 & & 4 & & 5 \\
 & & y & & y & & y & & y & & y \\
 & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 & & c & & c & & c & & c & & c \\
 & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
x & \mapsto & x & \mapsto & x & \mapsto & x & \mapsto & x & \mapsto & x \\
 & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 & & z & & z & & z & & z & & z
\end{array}$$

Average distance



Softmin