Where the hell am I and where is the stuff around me?

SLAM in SE(2) with (i) measurement models of 2D/3D marker detectors, UWB, GPS/GNSS, odometry, and (ii) differential drive motion model

Karel Zimmermann

Problem definition

Complete states: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau\sim\pi}\{\sum_{r_t\sim\tau} \gamma^t r_t\} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Algorithm: $\mathbf{z}_0, \mathbf{u}_1, \mathbf{z}_1, \ldots$ => estimate $p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ $\overset{\pi(\mathbf{x}_t)}{=>}$ decide following action $\mathbf{u}_{t+1}$

perception (local, SLAM, object detection) control (planning, RL, opt.control

Localisation problem definition

Previous lecture only 1D/2D translations (no rotations)

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$ .... ~~6DOF~~ robot's poses (no map for now)

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$ .... generated by external source

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$ .... comes from variety of sensors

MAP: $\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x} \,|\, \mathbf{z}, \mathbf{u}) = \arg\max_{\mathbf{x}_0 \ldots \mathbf{x}_t} p(\mathbf{x}_0 \ldots \mathbf{x}_t \,|\, \mathbf{z}_1 \ldots \mathbf{z}_t, \mathbf{u}_1 \ldots \mathbf{u}_t)$

Unknown

1. Construct p(x|z)
2. Optimize poses

Localisation problem definition

Today only 2D translations + 1D rotation

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$ .... ~~6DOF~~ robot's poses (no map for now)

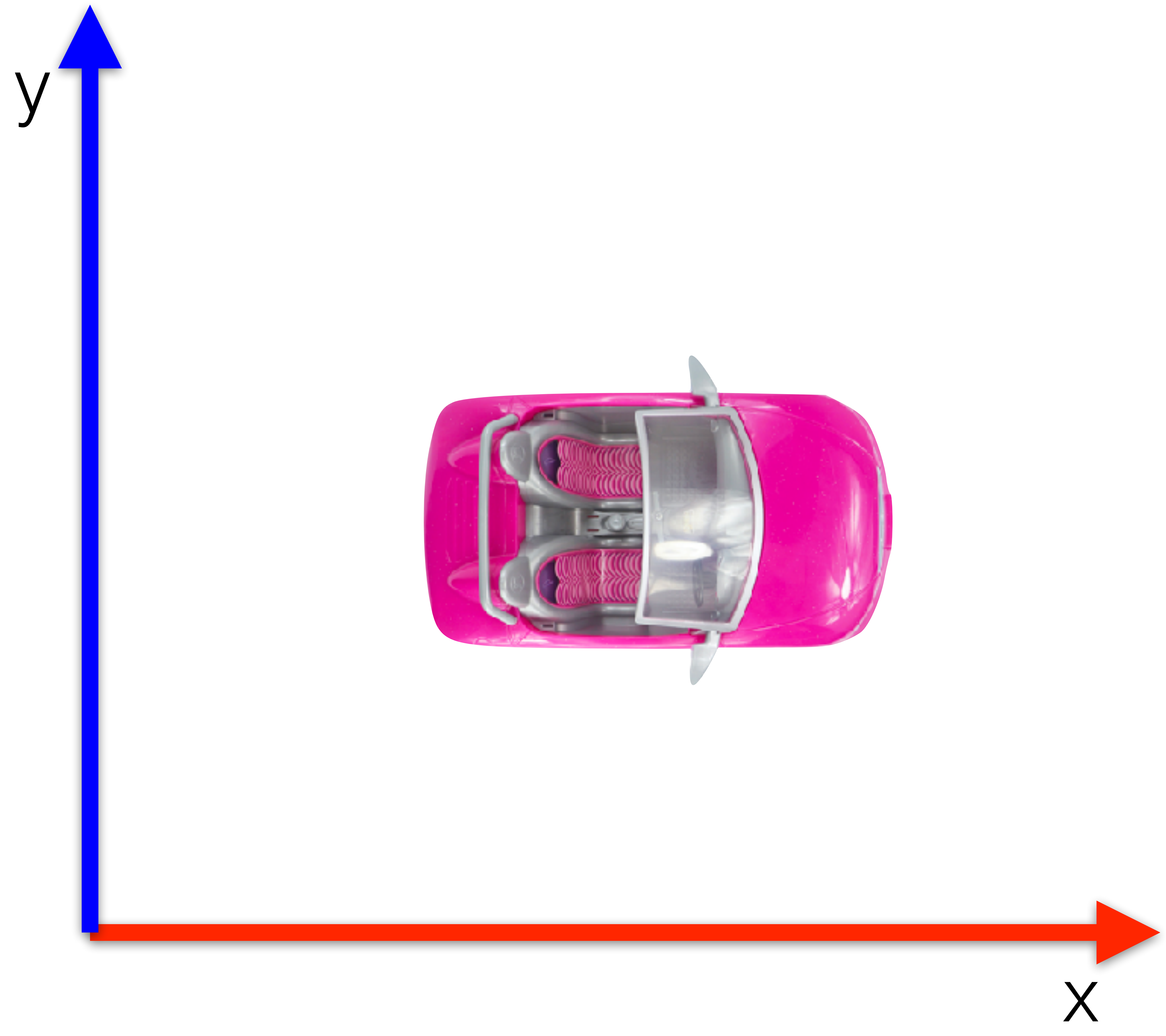
Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$ .... generated by external source

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$ .... comes from variety of sensors

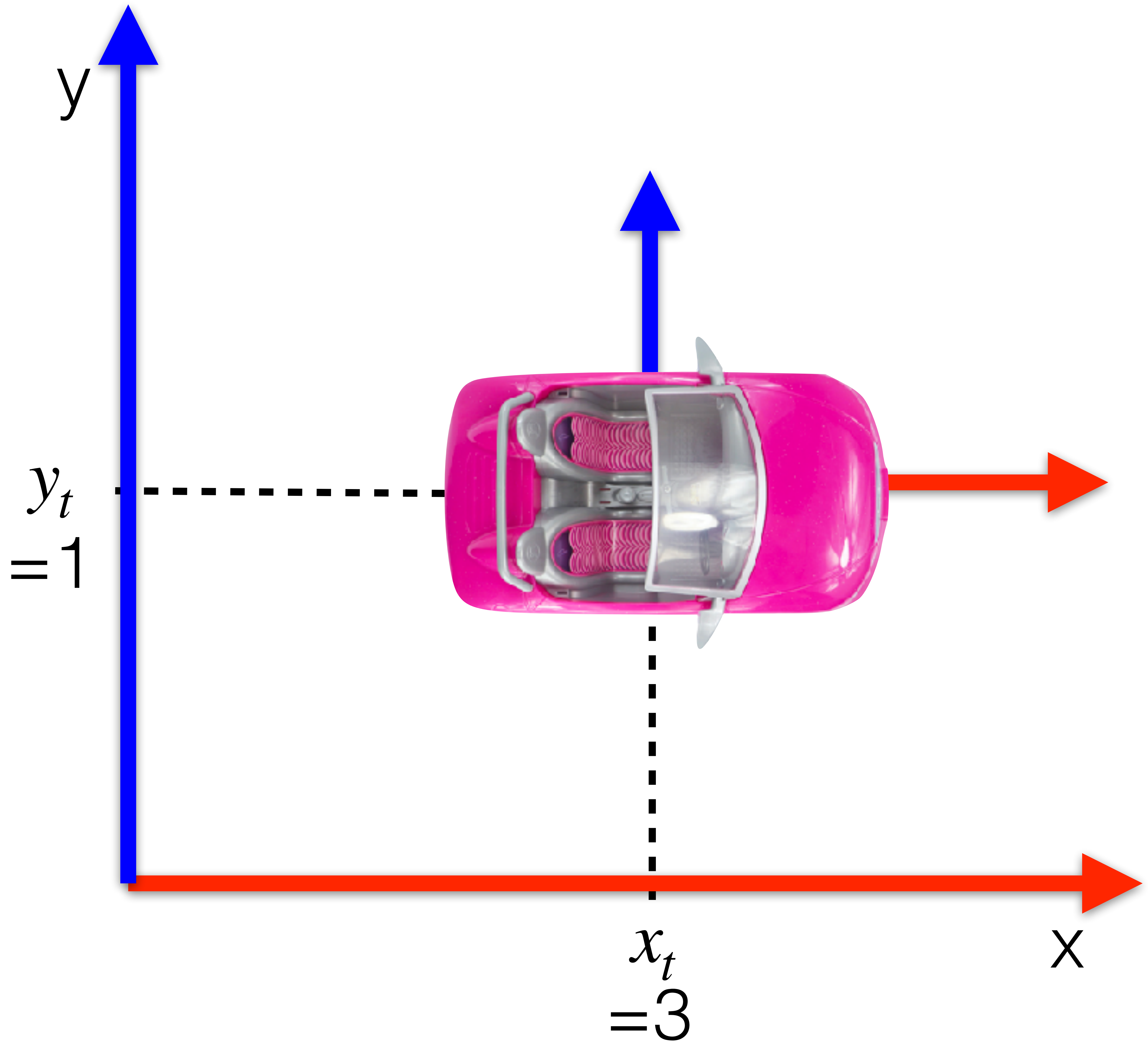
Unknown

MAP: $\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x} \,|\, \mathbf{z}, \mathbf{u}) = \arg\max_{\mathbf{x}_0 \ldots \mathbf{x}_t} p(\mathbf{x}_0 \ldots \mathbf{x}_t \,|\, \mathbf{z}_1 \ldots \mathbf{z}_t, \mathbf{u}_1 \ldots \mathbf{u}_t)$

1. Construct p(x|z)
2. Optimize poses

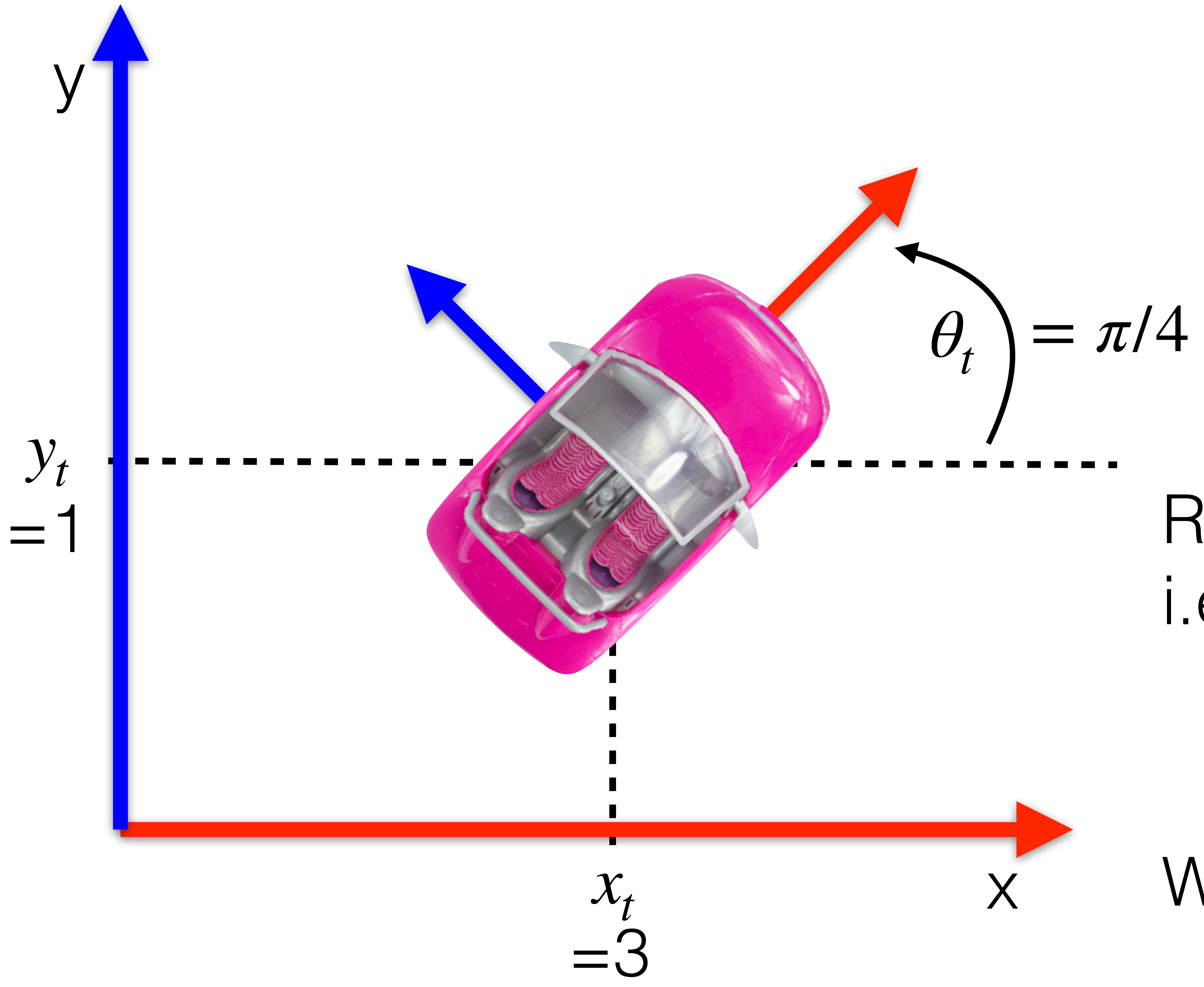


World coordinate frame (wcf)



Robot coordinate frame (rcf)
i.e. pose of robot in wcf

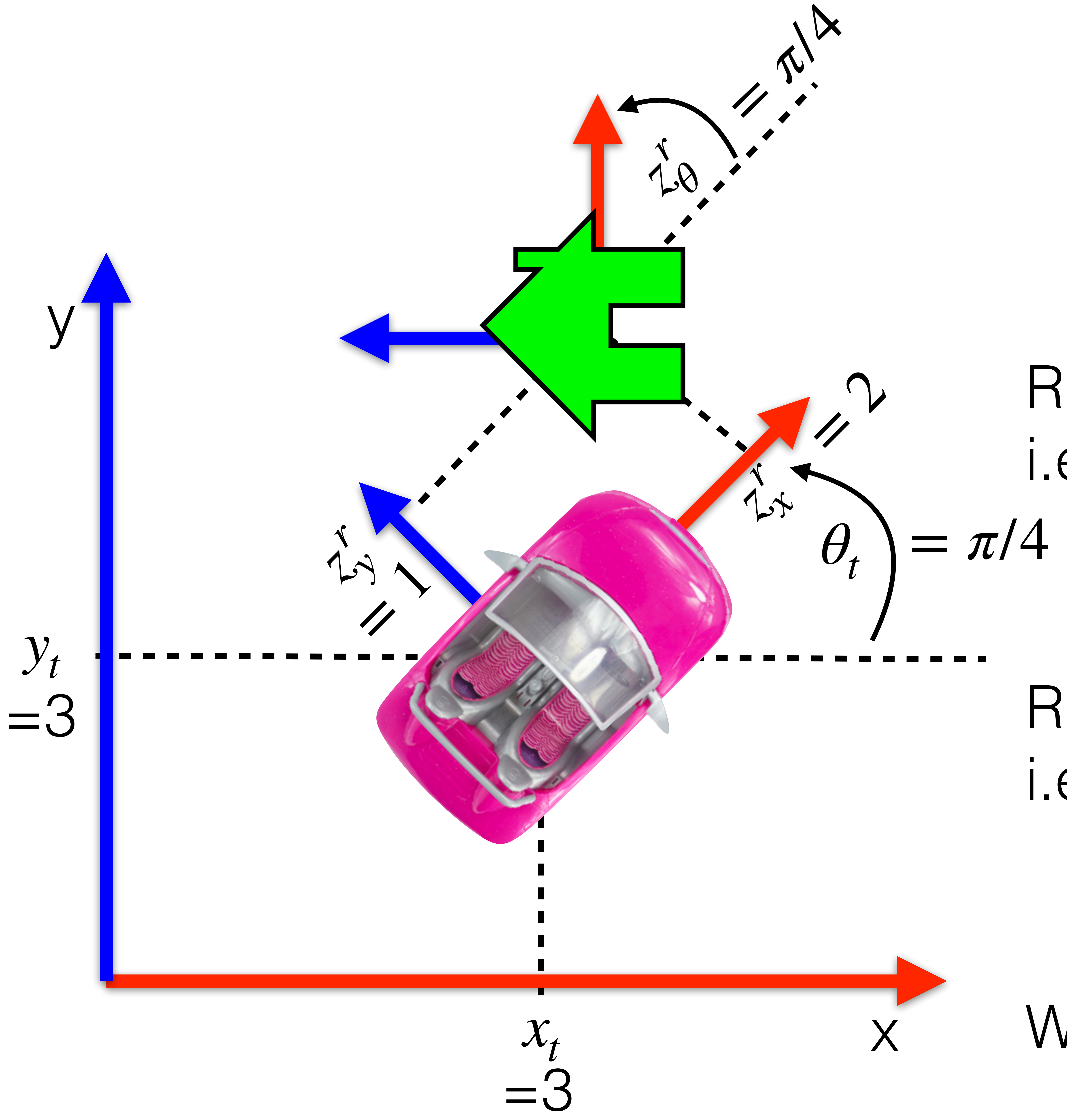
World coordinate frame (wcf)



Robot coordinate frame (rcf)
i.e. pose of robot in wcf

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

World coordinate frame (wcf)



Robot sees (measures) house in rcf
i.e. pose of house in rcf

$$\mathbf{z} = \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix}$$

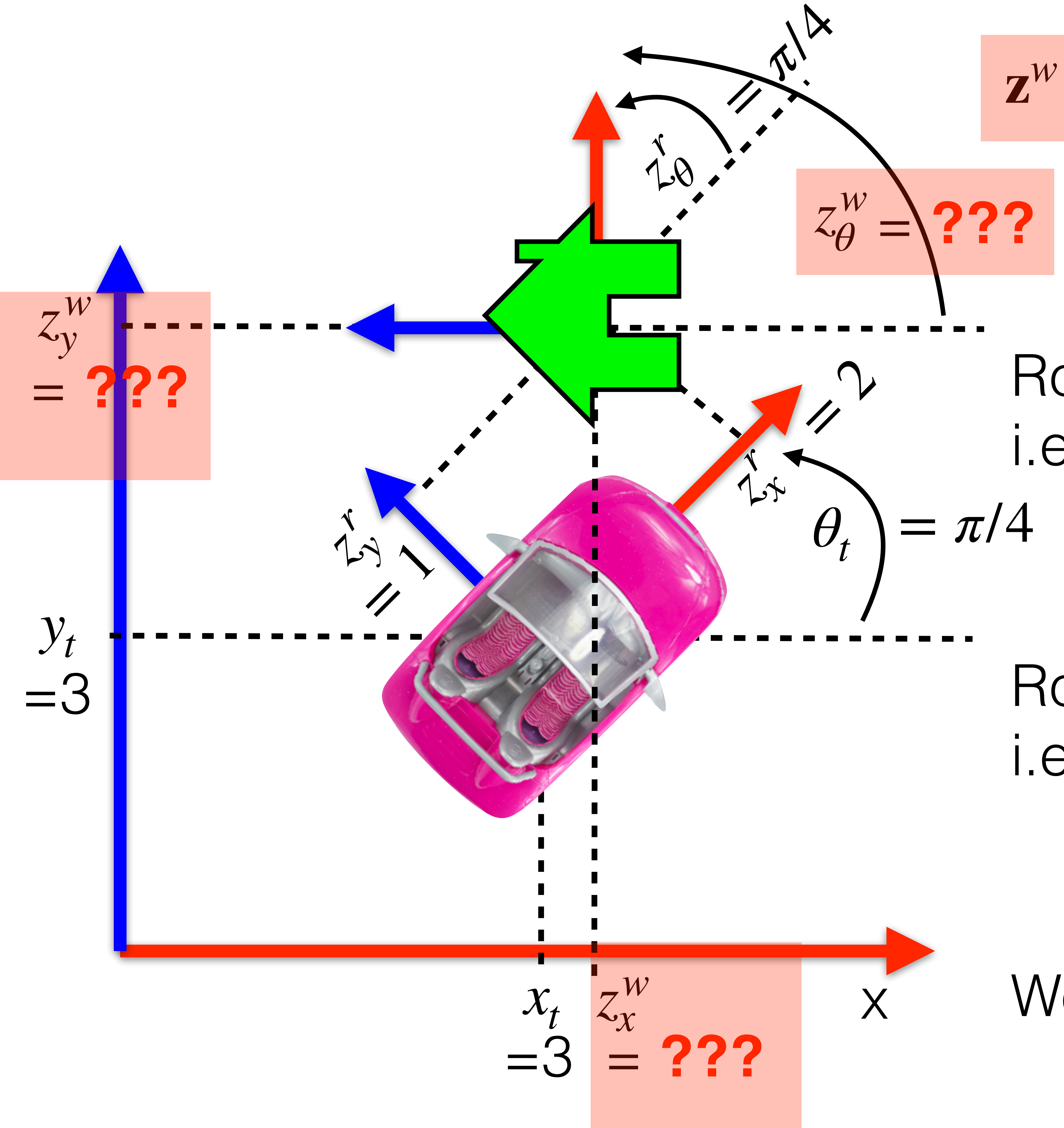
Robot coordinate frame (rcf)
i.e. pose of robot in wcf

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

World coordinate frame (wcf)

Pose of house in wcf:

$$\mathbf{z}^w = \begin{bmatrix} z_x^w \\ z_y^w \\ z_\theta^w \end{bmatrix} = \begin{bmatrix} \cos\theta_t & -\sin\theta_t & 0 \\ \sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$



Robot sees (measures) house in rcf
i.e. pose of house in rcf

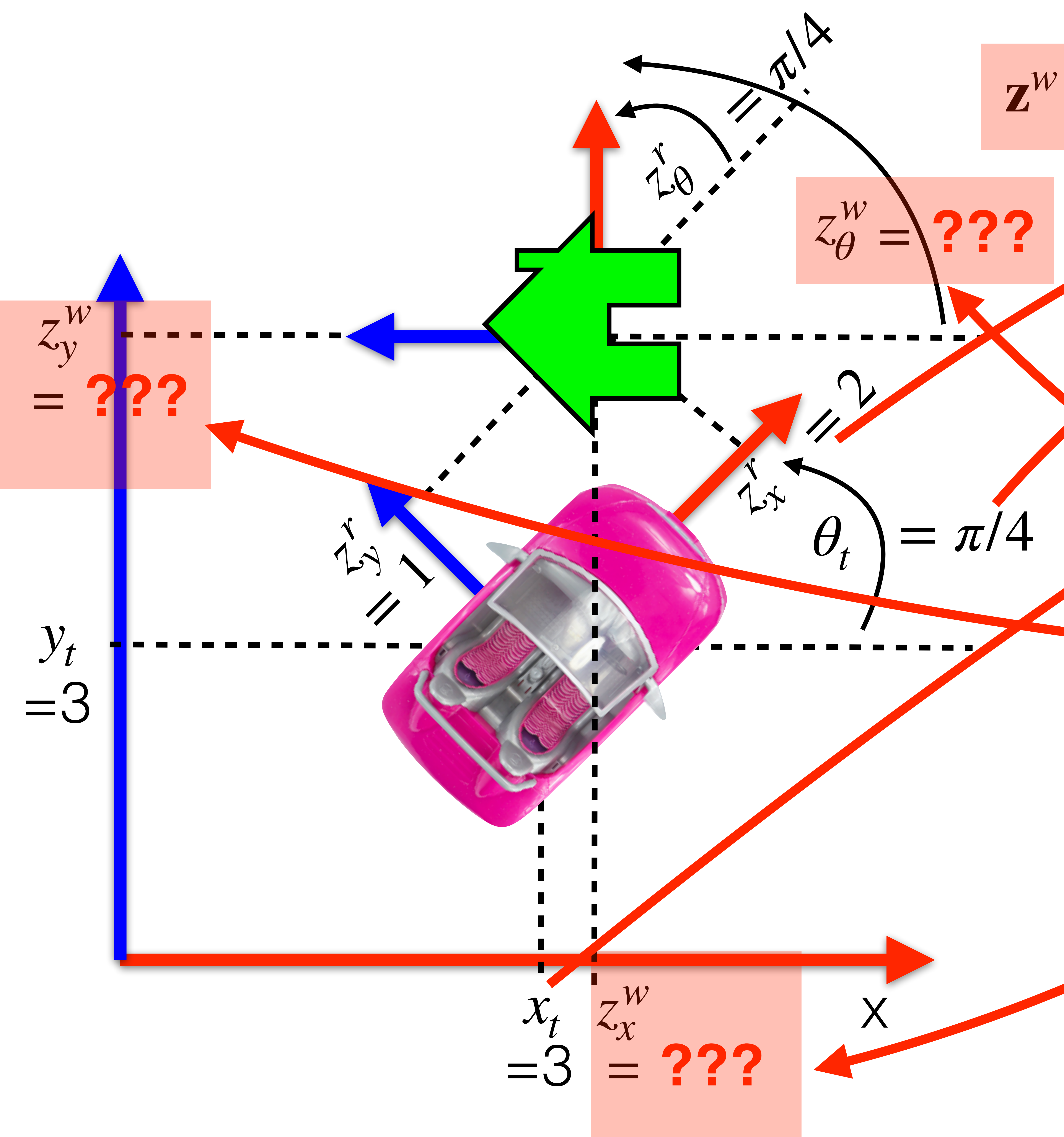
$$\mathbf{z} = \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix}$$

Robot coordinate frame (rcf)
i.e. pose of robot in wcf

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

World coordinate frame (wcf)

Pose of house in wcf:

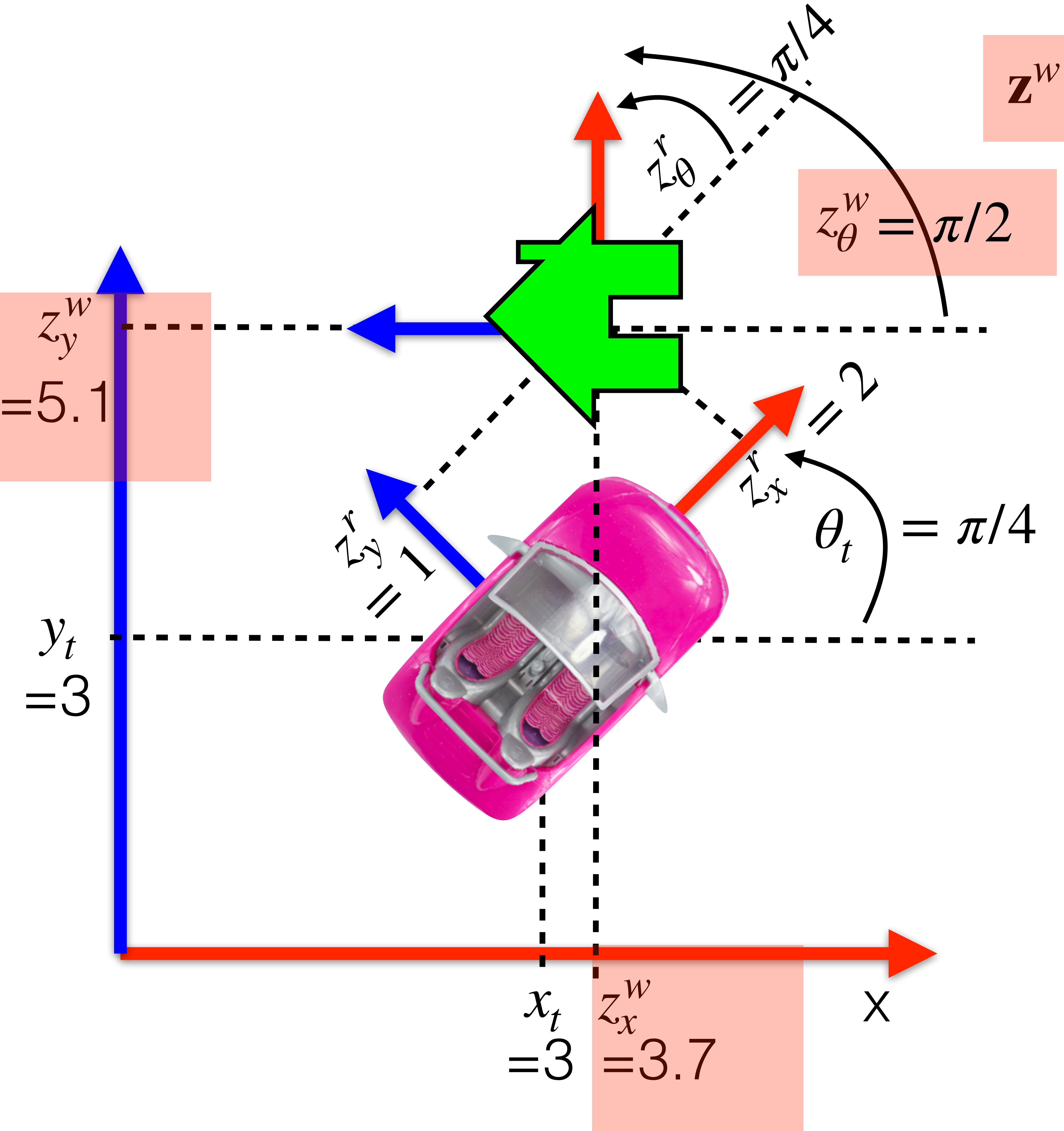


$$\mathbf{z}^w = \begin{bmatrix} z_x^w \\ z_y^w \\ z_\theta^w \end{bmatrix} = \begin{bmatrix} \cos\theta_t & -\sin\theta_t & 0 \\ \sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

$$= \begin{bmatrix} \cos\pi/4 & -\sin\pi/4 & 0 \\ \sin\pi/4 & \cos\pi/4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ \pi/4 \end{bmatrix} + \begin{bmatrix} 3 \\ 3 \\ \pi/4 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2}+3 \\ 3/\sqrt{2}+3 \\ \pi/2 \end{bmatrix} \approx \begin{bmatrix} 3.7 \\ 5.1 \\ \pi/2 \end{bmatrix}$$

Pose of house in wcf:



$$\mathbf{z}^w = \begin{bmatrix} z_x^w \\ z_y^w \\ z_\theta^w \end{bmatrix} = \begin{bmatrix} \cos\theta_t & -\sin\theta_t & 0 \\ \sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$$

$$= \begin{bmatrix} \cos\pi/4 & -\sin\pi/4 & 0 \\ \sin\pi/4 & \cos\pi/4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ \pi/4 \end{bmatrix} + \begin{bmatrix} 3 \\ 3 \\ \pi/4 \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} + 3 \\ 3/\sqrt{2} + 3 \\ \pi/2 \end{bmatrix} \approx \begin{bmatrix} 3.7 \\ 5.1 \\ \pi/2 \end{bmatrix}$$

Pose of the house transformed from rcf to wcf:

$$\mathbf{z}^w = \begin{bmatrix} z_x^w \\ z_y^w \\ z_\theta^w \end{bmatrix} = \begin{bmatrix} \cos\theta_t & -\sin\theta_t & 0 \\ \sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} R(\theta_t) & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{z}^r + \mathbf{x}_t \; = T(\mathbf{z}^r, \mathbf{x}_t) \; = \mathrm{r2w}(\mathbf{z}^r, \mathbf{x}_t)$$

Pose of the house transformed from wcf to rcf:

$$\mathbf{z}^r = \begin{bmatrix} R(\theta_t)^\top & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} (\mathbf{z}^w - \mathbf{x}_t) = T^{-1}(\mathbf{z}^w, \mathbf{x}_t) \; = \mathrm{w2r}(\mathbf{z}^w, \mathbf{x}_t)$$

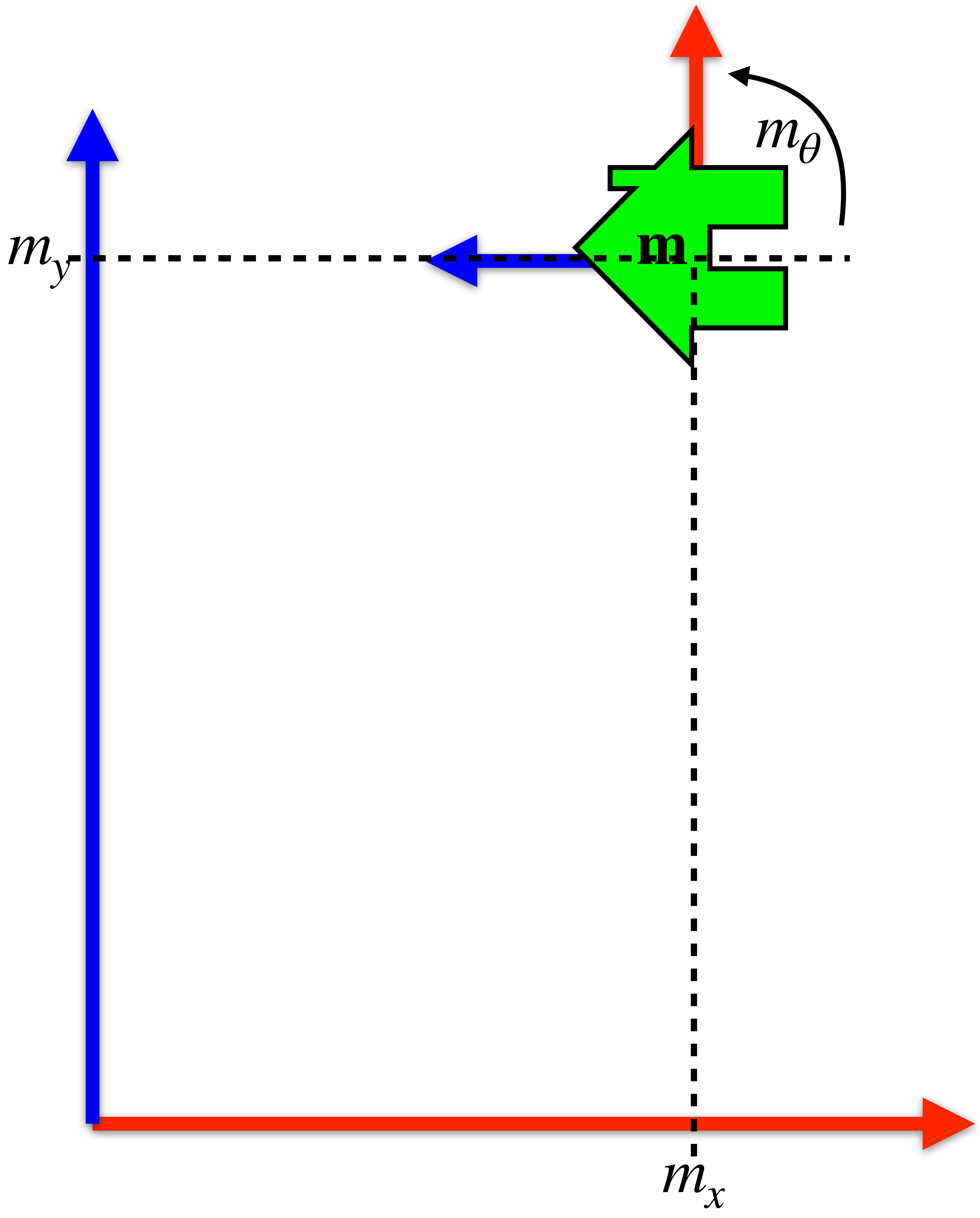
```
def r2w(z_r, x):
    z_w = torch.zeros(3)
    R = torch.vstack((torch.hstack((torch.cos(x[2]), -torch.sin(x[2]))), …
    z_w[0:2] = R @ z_r[0:2] + x[0:2]
    z_w[2] = z_r[2] + x[2]
    return z_w

def w2r(z_w, x):
    z_r = torch.zeros(3)
    R = torch.vstack((torch.hstack((torch.cos(x[2]), -torch.sin(x[2]))), …
    z_r[0:2] = R.t() @ (z_w[0:2] - x[0:2])
    z_r[2] = z_w[2] - x[2]
    return z_r
```

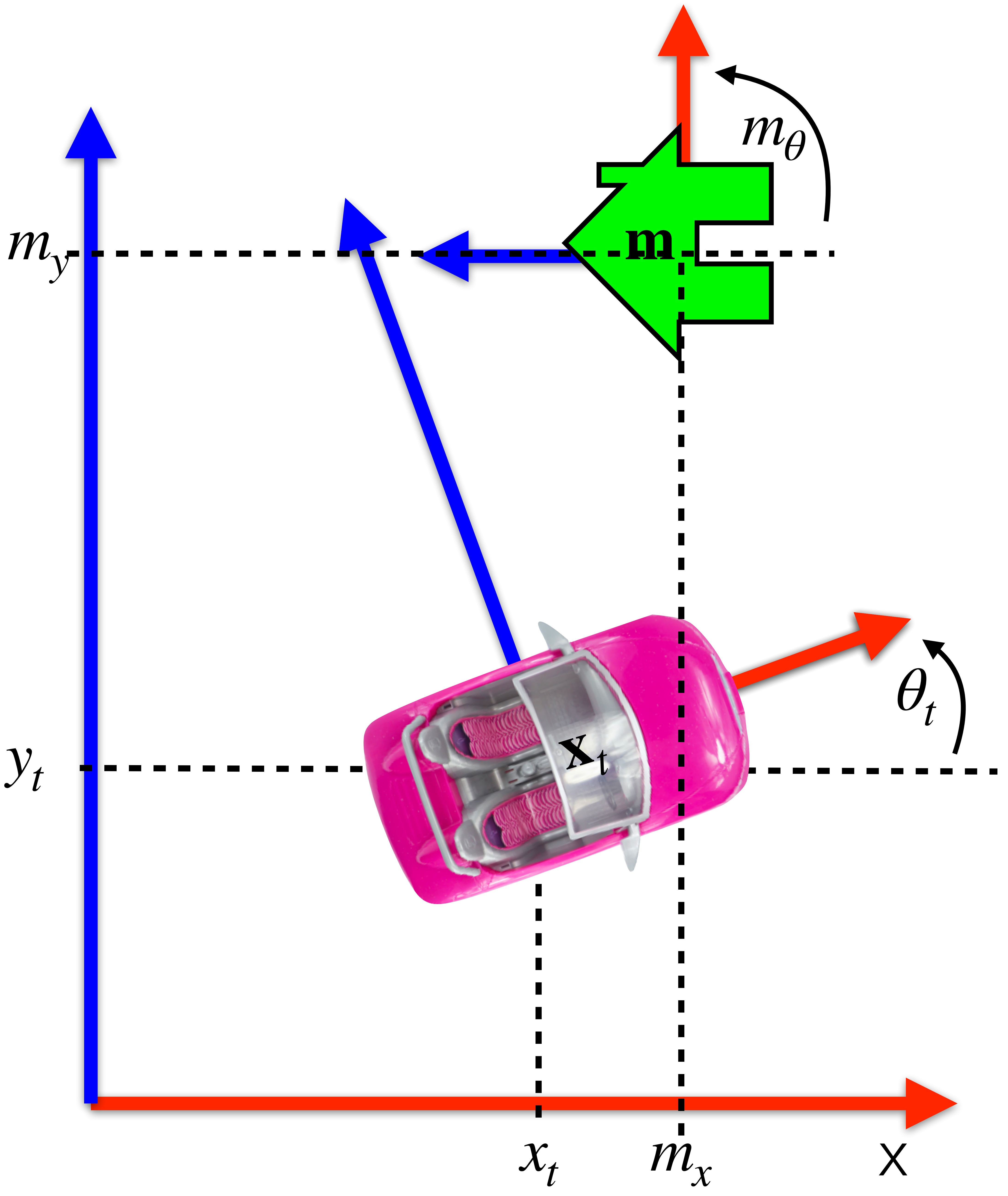
Localization of robot in wcf from known position of the house

Known marker pose in wcf

$$\mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_\theta \end{bmatrix}$$



Localization of robot in wcf from known position of the house



Known marker pose in wcf $\mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_\theta \end{bmatrix}$

Unknown robot pose in wcf $\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$

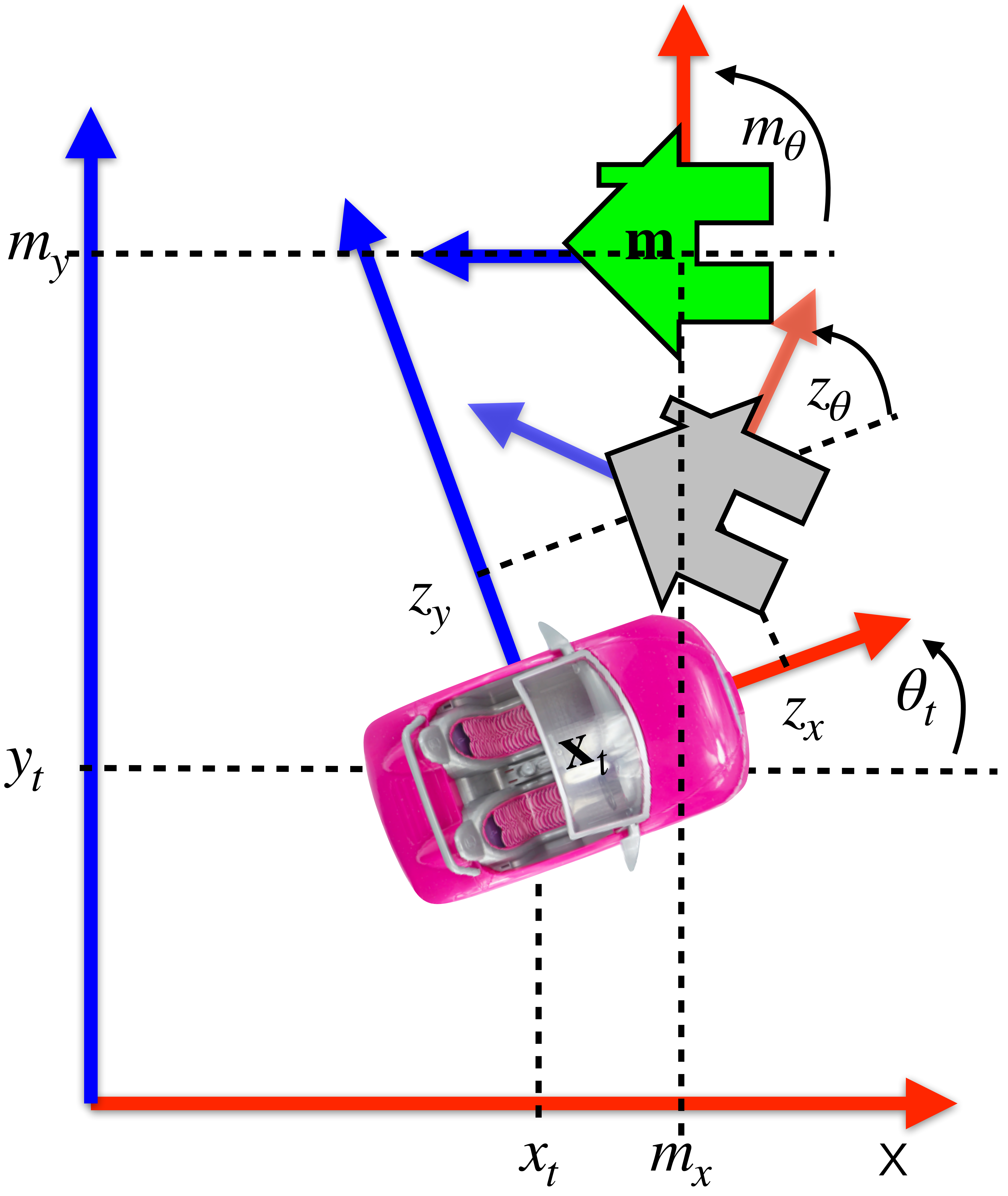
Robot is equipped by exteroceptive sensor:

RGBD camera

RGB camera

Lidar

Localization of robot in wcf from known position of the house



Known marker pose in wcf $\quad \mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_\theta \end{bmatrix}$

Unknown robot pose in wcf $\quad \mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$

Is this pose $\mathbf{x}_t$ correct?

Robot measures the house in rcf $\quad \mathbf{z} = \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix}$

What is the correct pose of robot?

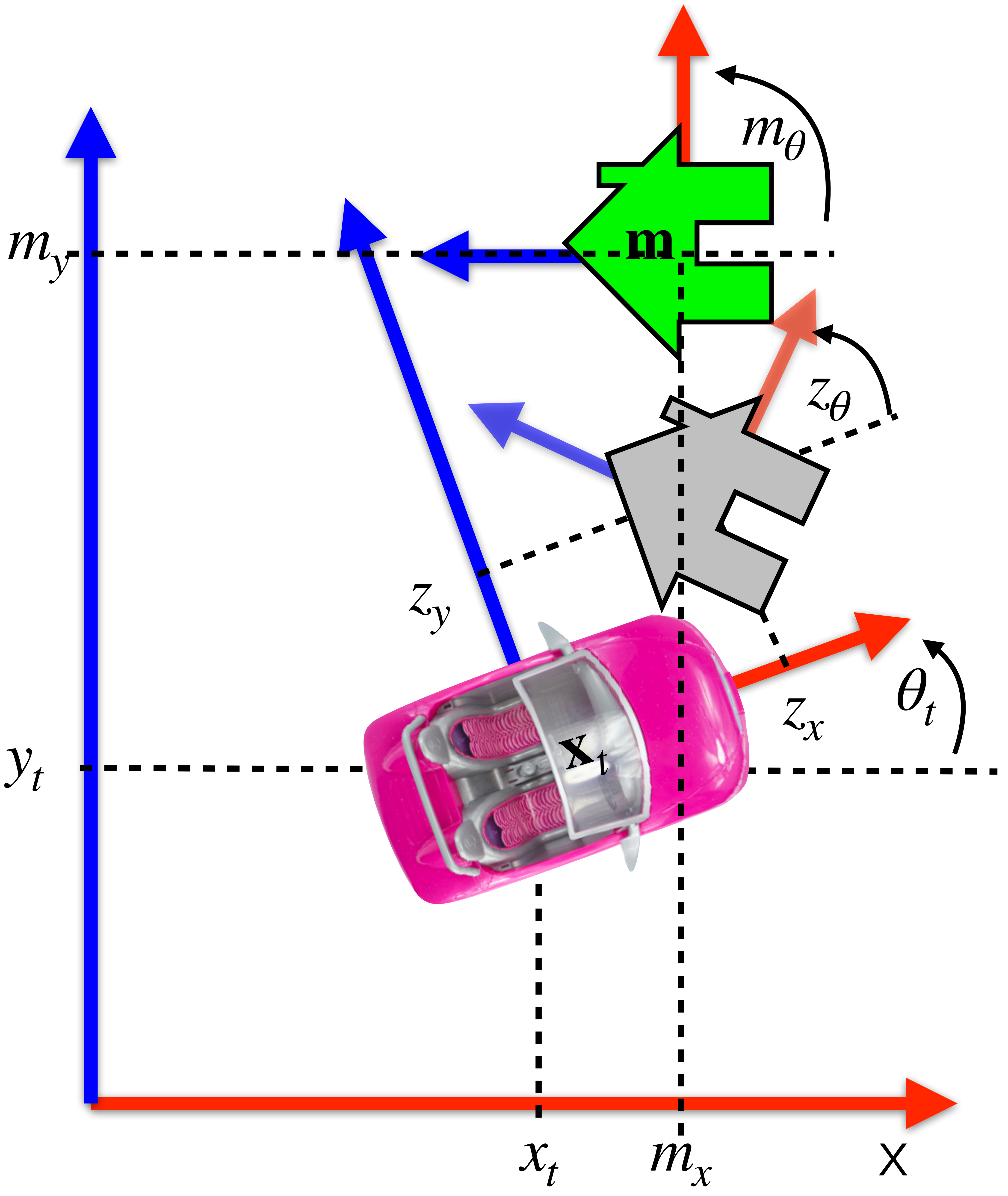
Robot is equipped by exteroceptive sensor:

RGBD camera

RGB camera

Lidar

Localization of robot in wcf from known position of the house



Known marker pose in wcf $\quad \mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_\theta \end{bmatrix}$

Unknown robot pose in wcf $\quad \mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$

Is this pose $\mathbf{x}_t$ correct?

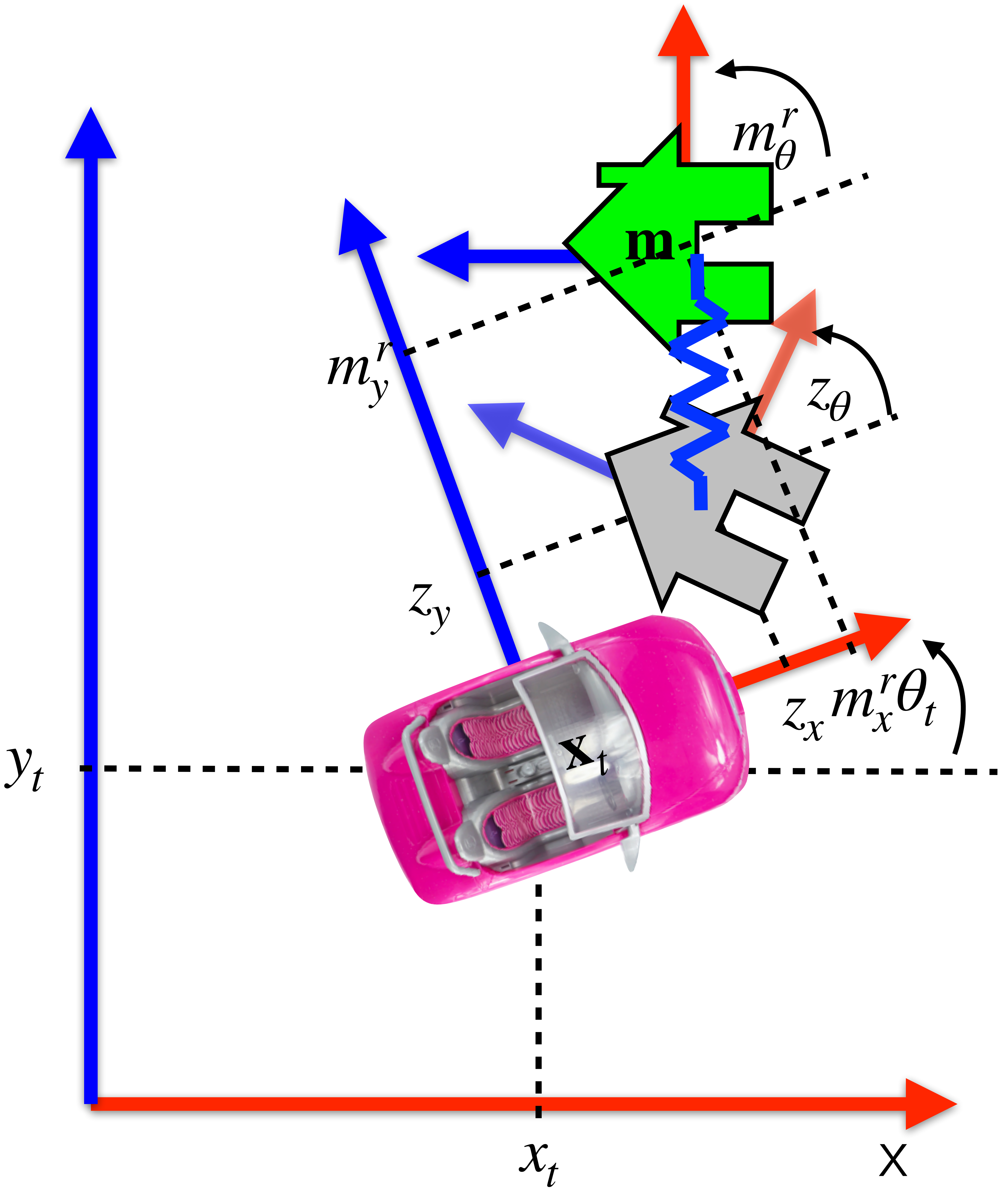
Robot measures the house in rcf $\quad \mathbf{z} = \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix}$

What is the correct pose of robot?

Correct pose is the one that aligns [grey house] with [green house] but they are in different coordinate frames!

Which coordinate frame should I use to measure their distance?

Localization of robot in wcf from known position of the house



Known marker pose in wcf $\quad \mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_\theta \end{bmatrix}$

Unknown robot pose in wcf $\quad \mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$

Is this pose $\mathbf{x}_t$ correct?

Robot measures the house in rcf $\quad \mathbf{z} = \begin{bmatrix} z_x^r \\ z_y^r \\ z_\theta^r \end{bmatrix}$

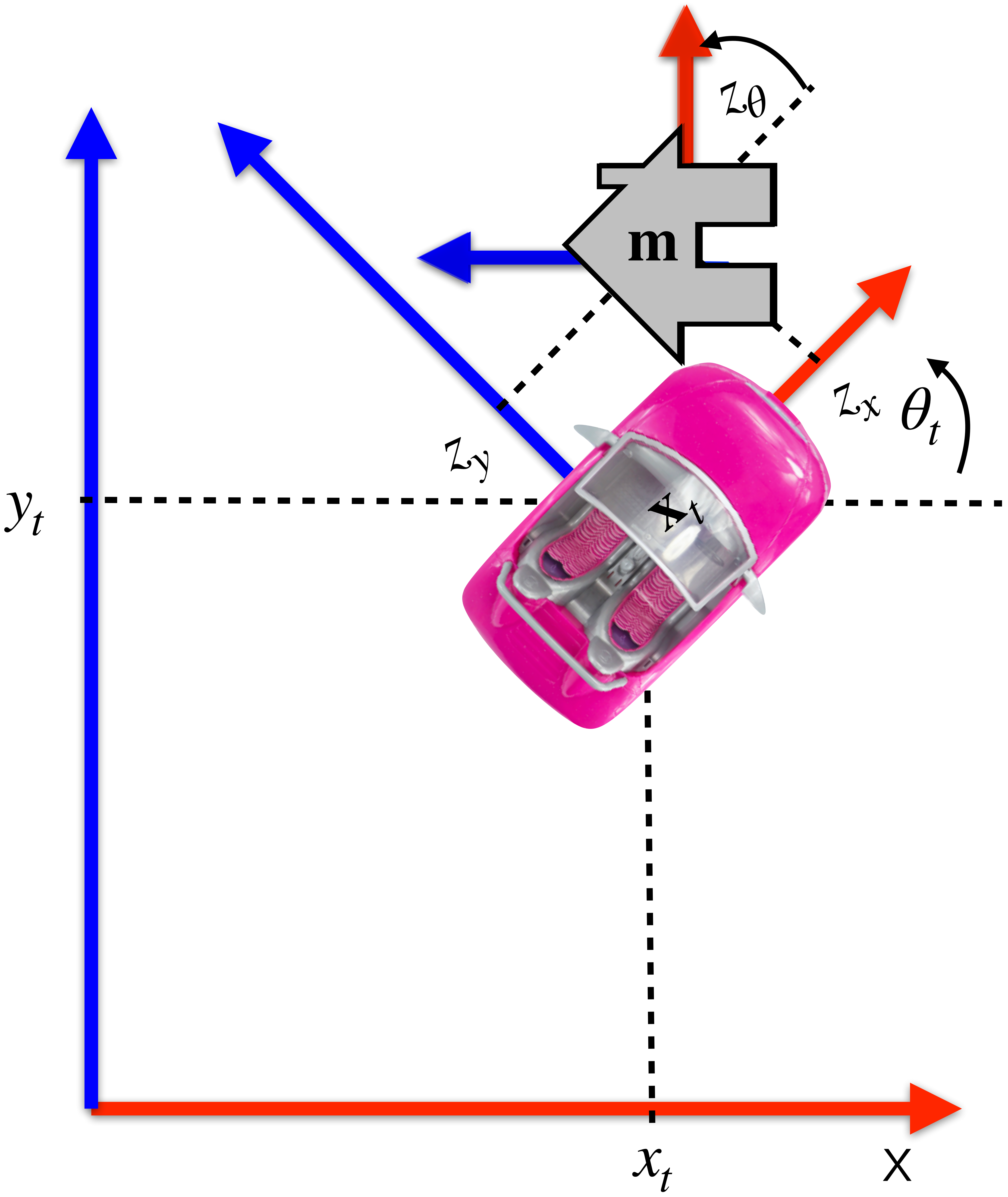
What is the correct pose of robot?

Correct pose is the one that aligns  with 
but they are in different coordinate frames!

Marker pose in rcf $\quad \mathbf{m}^r = \mathrm{w2r}(\mathbf{m}, \mathbf{x}_t)$

$$\mathbf{x}_t^\star = \arg\min_{\mathbf{x}_t} \|\mathbf{m}^r - \mathbf{z}\|^2 = \arg\min_{\mathbf{x}_t} \|\mathrm{w2r}(\mathbf{m}, \mathbf{x}_t) - \mathbf{z}\|^2$$

Localization of robot in wcf from known position of the house



Known marker pose in wcf $\quad \mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_\theta \end{bmatrix}$

Unknown robot pose in wcf $\quad \mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$

Is this pose $\mathbf{x}_t$ correct?

Robot measures the house in rcf $\quad \mathbf{z} = \begin{bmatrix} z^r_x \\ z^r_y \\ z^r_\theta \end{bmatrix}$

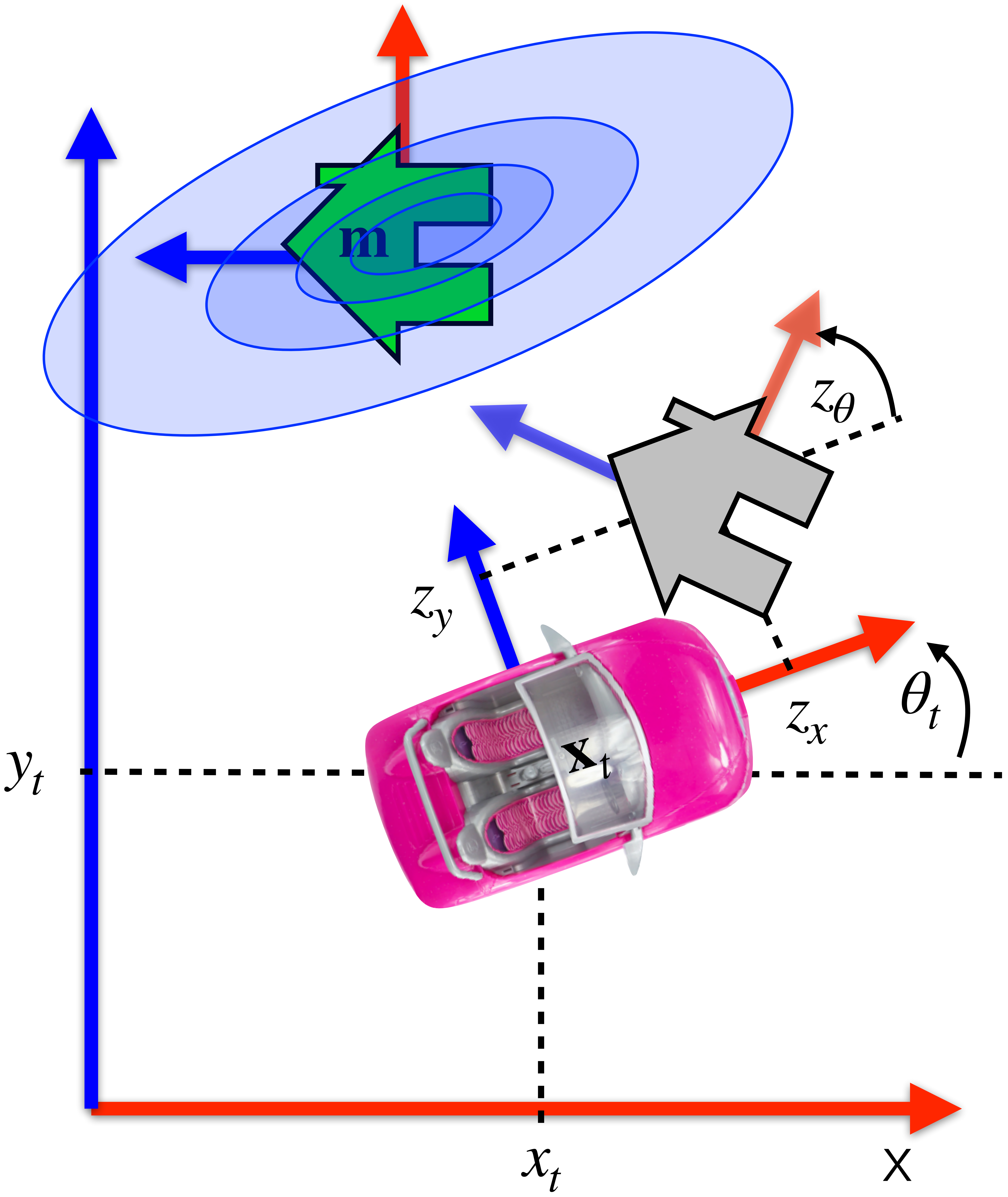
What is the correct pose of robot?

Correct pose is the one that aligns (grey house) with (green house) but they are in different coordinate frames!

Marker pose in rcf $\quad \mathbf{m}^r = \mathrm{w2r}(\mathbf{m}, \mathbf{x}_t)$

$$\mathbf{x}^\star_t = \arg\min_{\mathbf{x}_t} \|\mathbf{m}^r - \mathbf{z}\|^2 = \arg\min_{\mathbf{x}_t} \|\mathrm{w2r}(\mathbf{m}, \mathbf{x}_t) - \mathbf{z}\|^2$$
$$= \arg\min_{\mathbf{x}_t} \|\mathrm{r2w}(\mathbf{z}, \mathbf{x}_t) - \mathbf{m}\|^2$$

Localization of robot in wcf from known position of the house

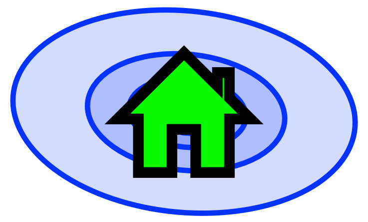


We completely ignored measurement inaccuracy

Given measurements $\mathbf{z}$ in rcf are normally distrib.

$$p(\mathbf{z}\,|\,\mathbf{x}_t) = \mathcal{N}(\mathbf{z};\ \underbrace{\mathrm{w2r}(\mathbf{m}, \mathbf{x}_t)}_{\mathbf{m}^r},\ \Sigma)$$

Dimensionality?

Correct pose maximize prob of  in 

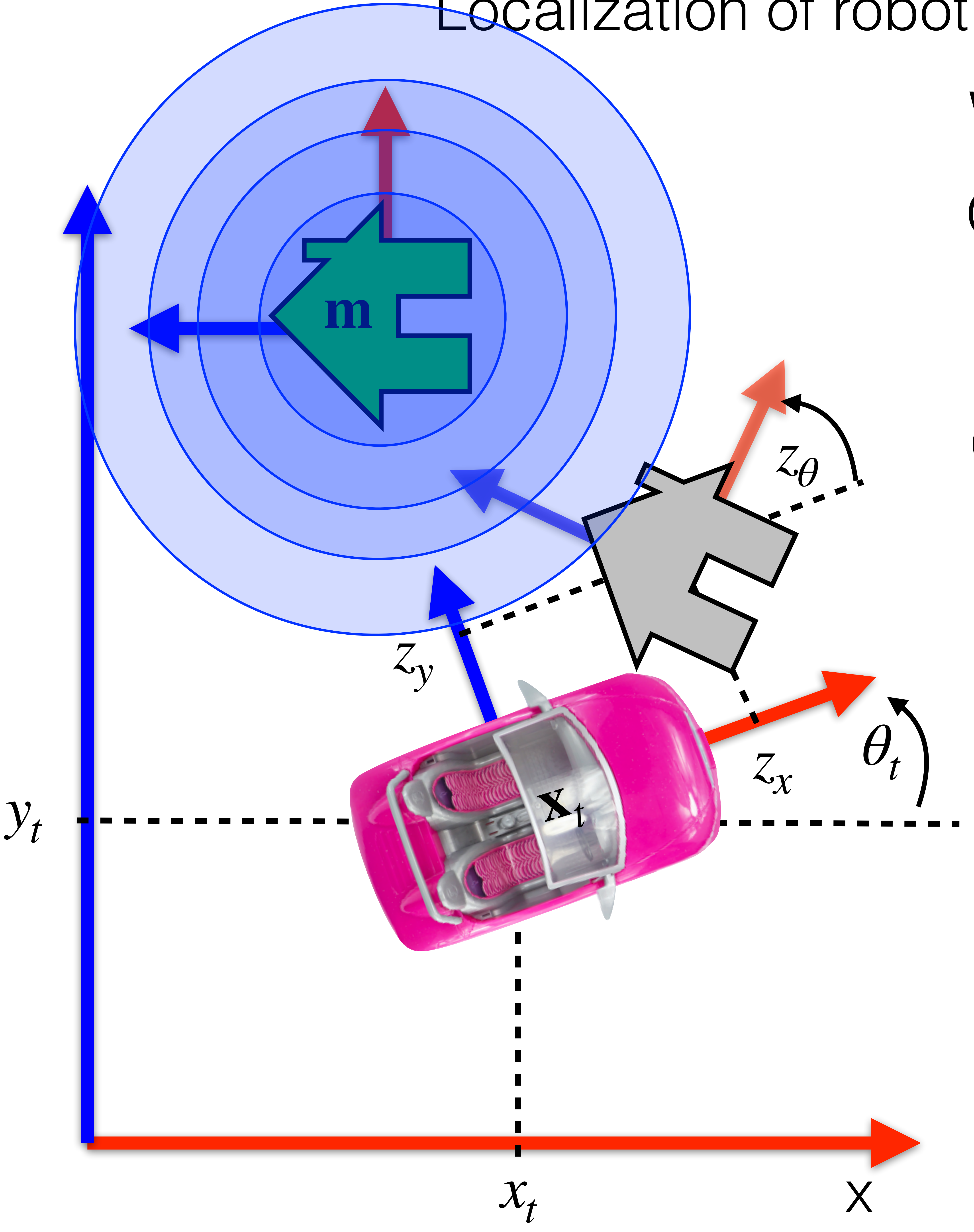
$$\mathbf{x}_t^\star = \arg\max_{\mathbf{x}_t} \mathcal{N}(\mathbf{z};\ \mathrm{w2r}(\mathbf{m}, \mathbf{x}_t),\ \Sigma)$$

$$= \arg\min_{\mathbf{x}_t} \|\mathrm{w2r}(\mathbf{m}, \mathbf{x}_t) - \mathbf{z}\|_\Sigma^2$$

$$\neq \arg\min_{\mathbf{x}_t} \|\mathrm{r2w}(\mathbf{z}, \mathbf{x}_t) - \mathbf{m}\|_\Sigma^2$$

… **why ???**

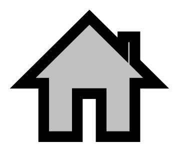
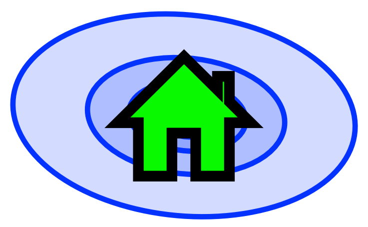
Localization of robot in wcf from known position of the house



We completely ignored measurement inaccuracy

Given $\mathbf{m}^r$ measurements will be normally distrib.

$$p(\mathbf{z}\,|\,\mathbf{x}_t) = \mathcal{N}(\mathbf{z};\ \underbrace{\mathrm{w2r}(\mathbf{m}, \mathbf{x}_t)}_{\mathbf{m}^r},\ \Sigma)$$

Correct pose maximize prob of  in 

$$\begin{aligned}\mathbf{x}_t^\star &= \arg\max_{\mathbf{x}_t} \mathcal{N}(\mathbf{z};\ \mathrm{w2r}(\mathbf{m}, \mathbf{x}_t),\ \Sigma)\\ &= \arg\min_{\mathbf{x}_t} \|\mathrm{w2r}(\mathbf{m}, \mathbf{x}_t) - \mathbf{z}\|_\Sigma^2\\ &\neq \arg\min_{\mathbf{x}_t} \|\mathrm{r2w}(\mathbf{z}, \mathbf{x}_t) - \mathbf{m}\|_\Sigma^2\end{aligned}$$

… **why ???**

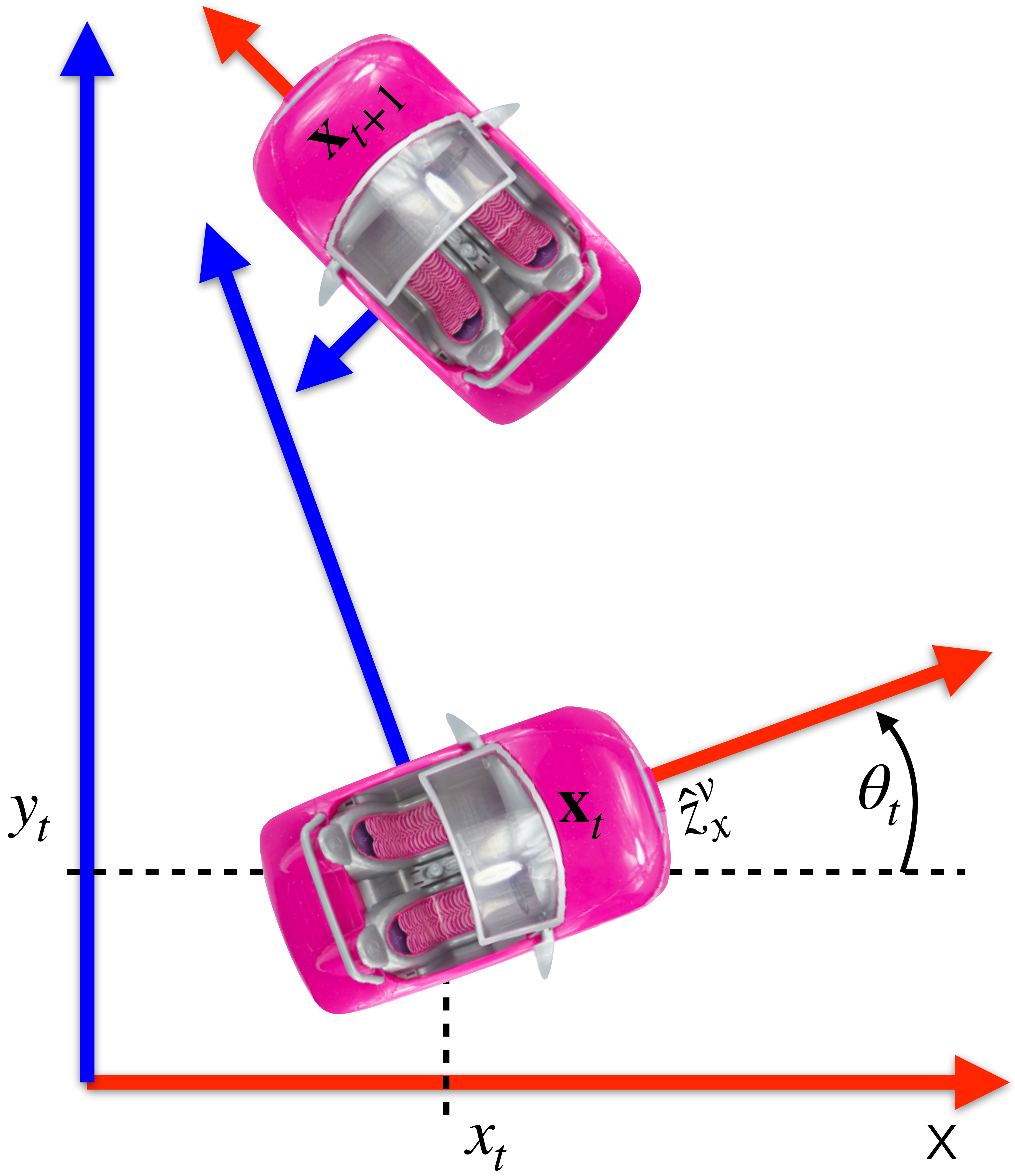
Especially if: $\Sigma = \frac{1}{c}\cdot \mathbf{I} = \begin{bmatrix} 1/c & 0 & 0\\ 0 & 1/c & 0 \\ 0 & 0 & 1/c\end{bmatrix}$

Then: $\mathbf{x}_t^\star = \arg\min_{\mathbf{x}_t} c\cdot \|\mathrm{w2r}(\mathbf{m}, \mathbf{x}_t) - \mathbf{z}\|^2$ rcf

$= \arg\min_{\mathbf{x}_t} c\cdot \|\mathrm{r2w}(\mathbf{z}, \mathbf{x}_t) - \mathbf{m}\|^2$ wcf

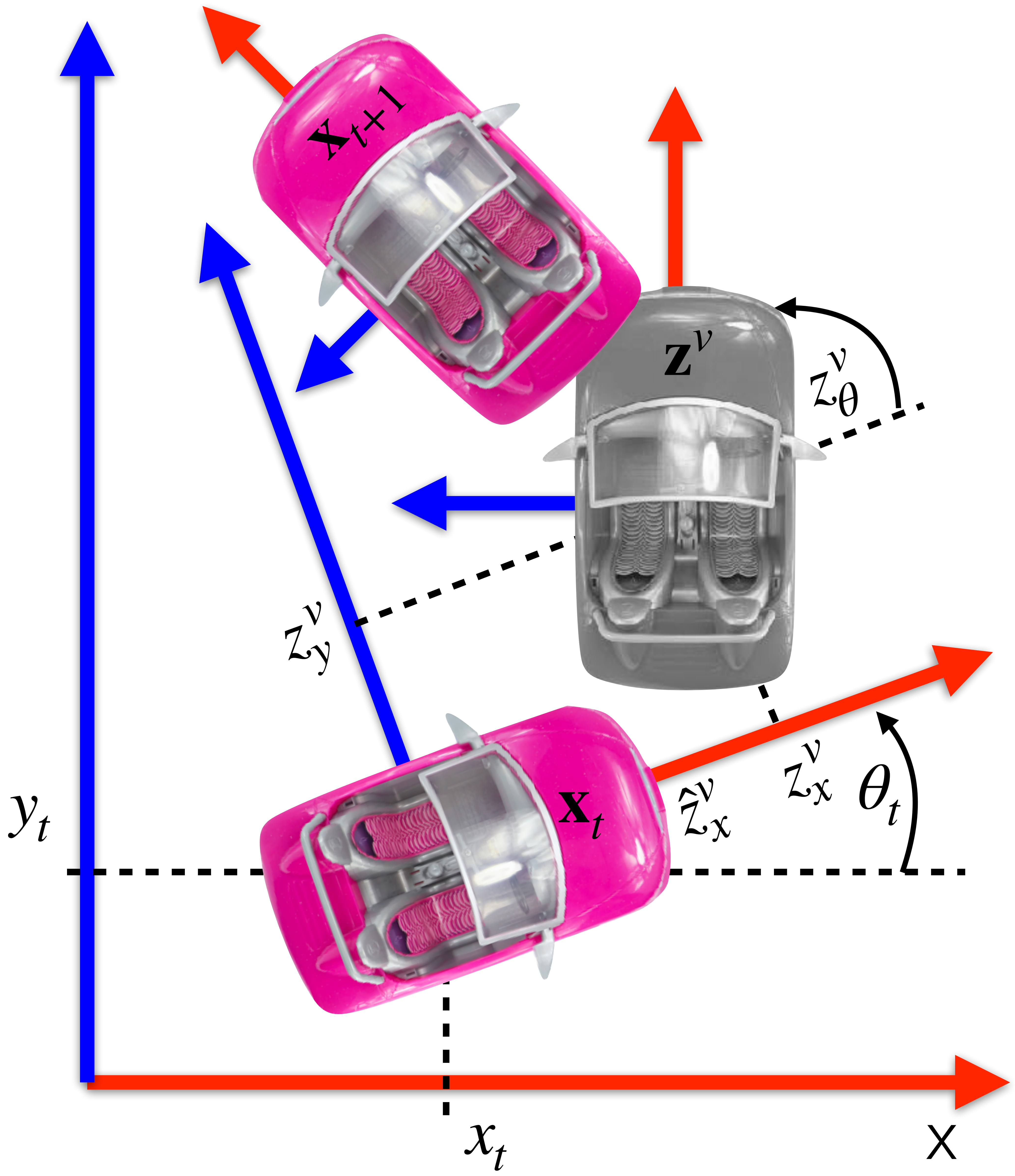
Localization of robot in wcf from known odometry

Odometry represented by linear+angular velocity



Robot poses in wcf: $\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \quad \mathbf{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix}$

Localization of robot in wcf from known odometry

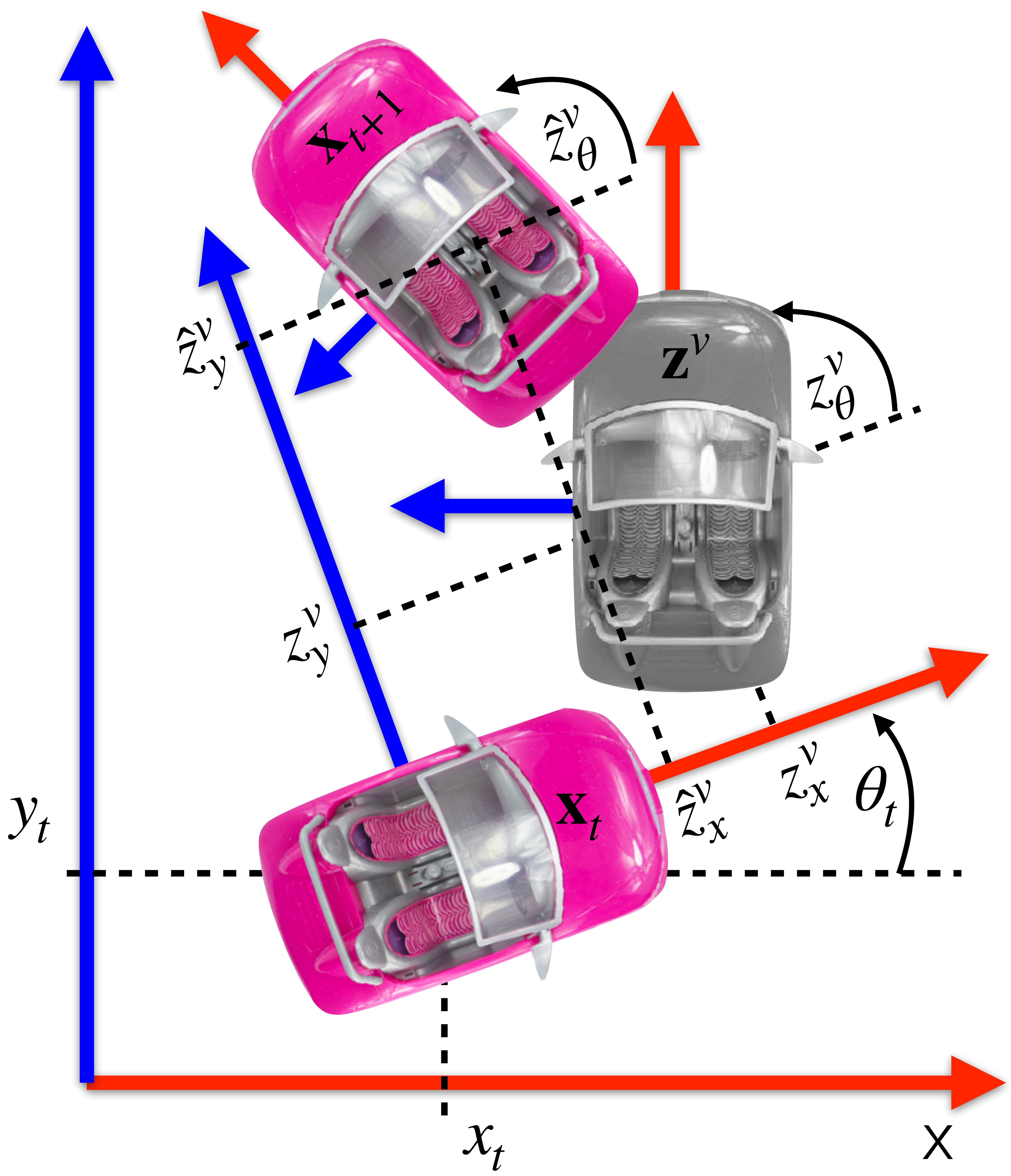


Odometry represented by linear+angular velocity

Robot poses in wcf: $\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \quad \mathbf{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix}$

Robot measures velocity in $\mathbf{x}_t$-rcf: $\mathbf{z}^v = \begin{bmatrix} z_x^v \\ z_y^v \\ z_\theta^v \end{bmatrix}$

Localization of robot in wcf from known odometry



Odometry represented by linear+angular velocity

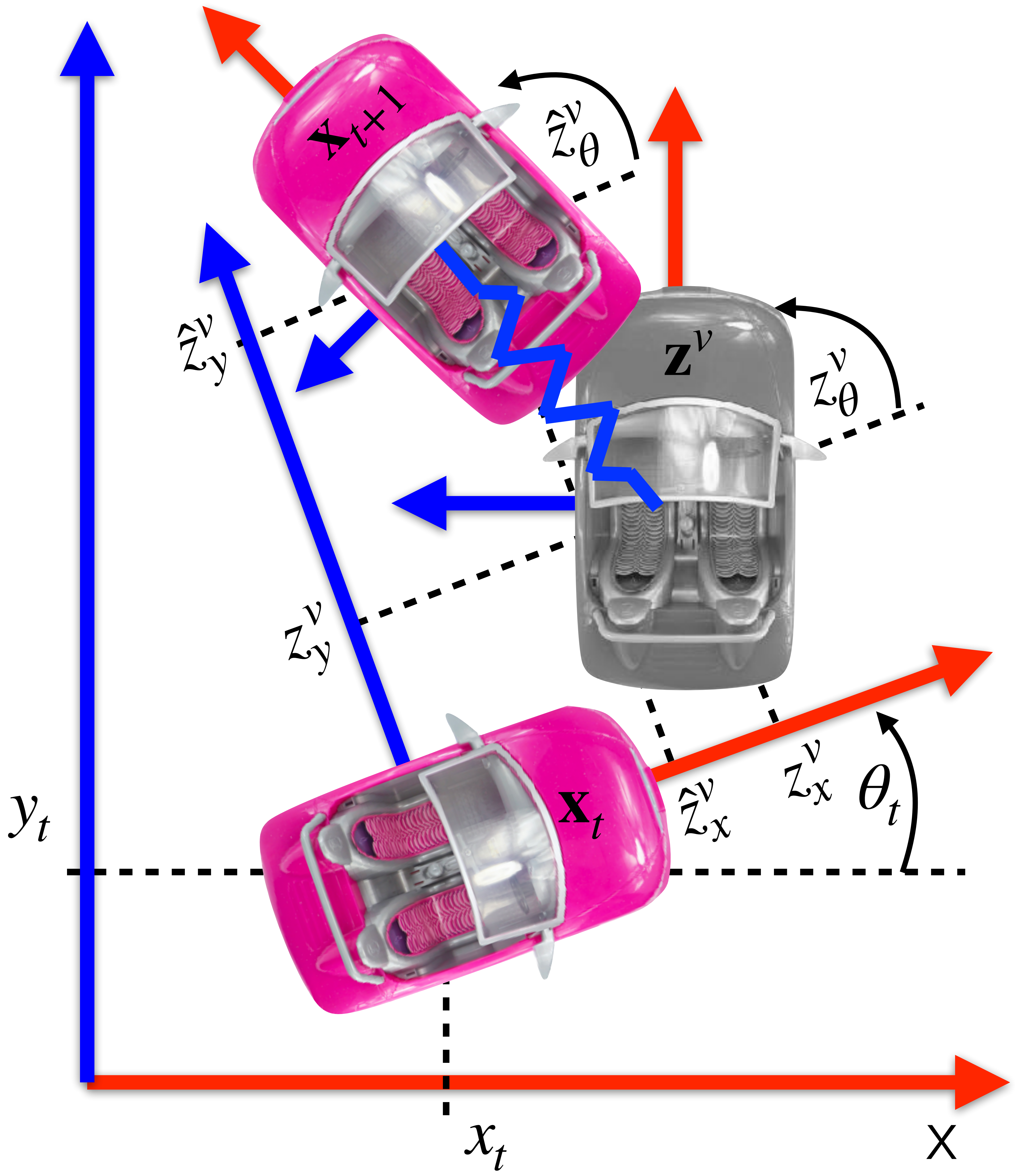
Robot poses in wcf: $\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \quad \mathbf{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix}$

Robot measures velocity in $\mathbf{x}_t$-rcf: $\mathbf{z}^v = \begin{bmatrix} z^v_x \\ z^v_y \\ z^v_\theta \end{bmatrix}$

Next pose $\mathbf{x}_{t+1}$ in $\mathbf{x}_t$-rcf: $\hat{\mathbf{z}}^v = \mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t)$

What is the correct pose of robot?

Localization of robot in wcf from known odometry



Odometry represented by linear+angular velocity

Robot poses in wcf: $\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$ $\mathbf{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix}$

Robot measures velocity in $\mathbf{x}_t$-rcf: $\mathbf{z}^v = \begin{bmatrix} z^v_x \\ z^v_y \\ z^v_\theta \end{bmatrix}$

Next pose $\mathbf{x}_{t+1}$ in $\mathbf{x}_t$-rcf: $\hat{\mathbf{z}}^v = \text{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t)$

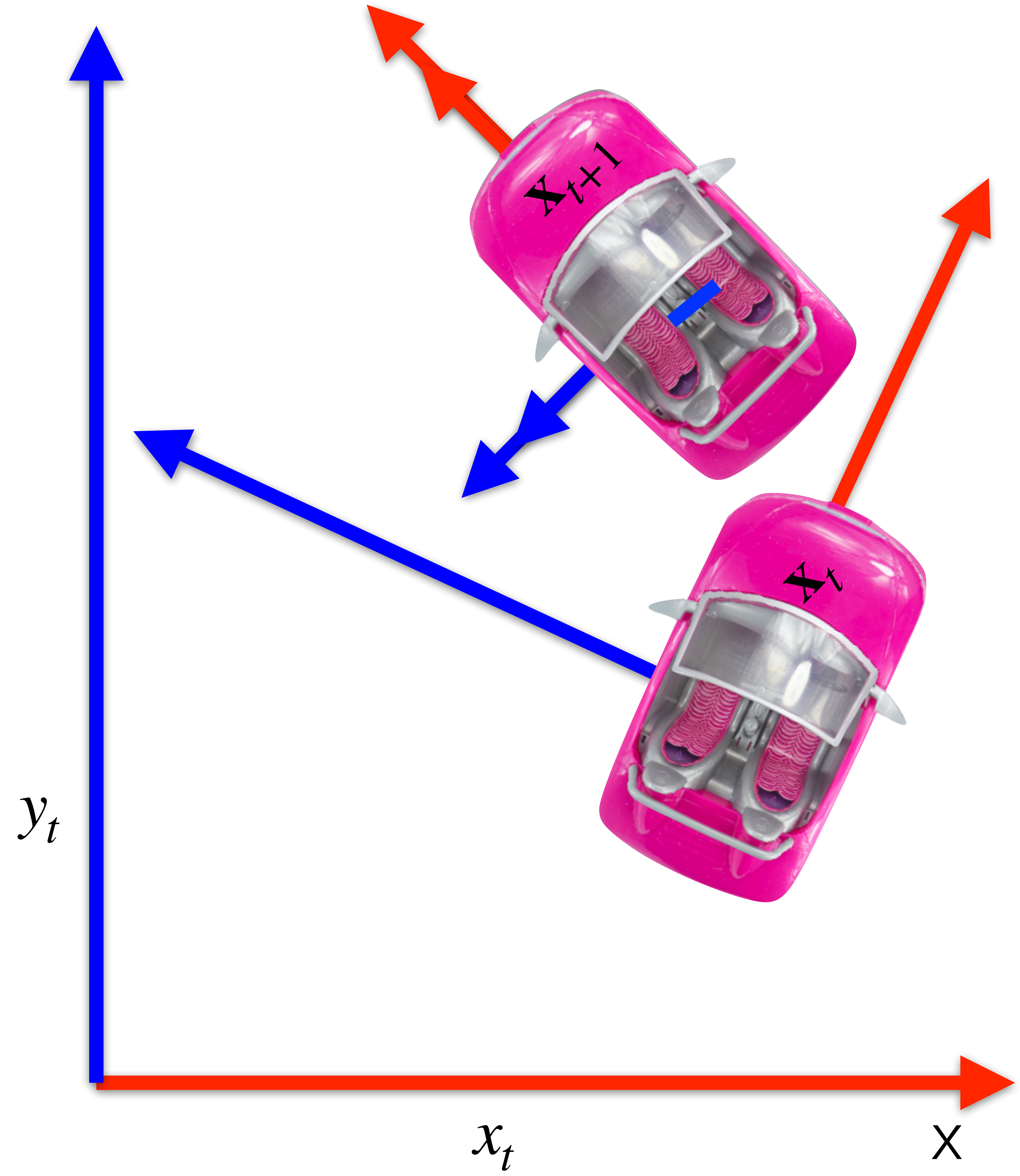
What is the correct pose of robot?

Find the correct poses

$$\mathbf{x}^\star_t, \mathbf{x}^\star_{t+1} = \arg\max_{\mathbf{x}_t, \mathbf{x}_{t+1}} \mathcal{N}(\mathbf{z}^v;\ \text{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t), \Sigma^v)$$

$$= \arg\min_{\mathbf{x}_t, \mathbf{x}_{t+1}} \|\text{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}^v\|^2_{\Sigma^v}$$

Localization of robot in wcf from known odometry



Odometry represented by linear+angular velocity

Robot poses in wcf: $\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \quad \mathbf{x}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix}$

Robot measures velocity in $\mathbf{x}_t$-rcf: $\mathbf{z}^v = \begin{bmatrix} z_x^v \\ z_y^v \\ z_\theta^v \end{bmatrix}$

Next pose $\mathbf{x}_{t+1}$ in $\mathbf{x}_t$-rcf: $\hat{\mathbf{z}}^v = \text{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t)$

What is the correct pose of robot?

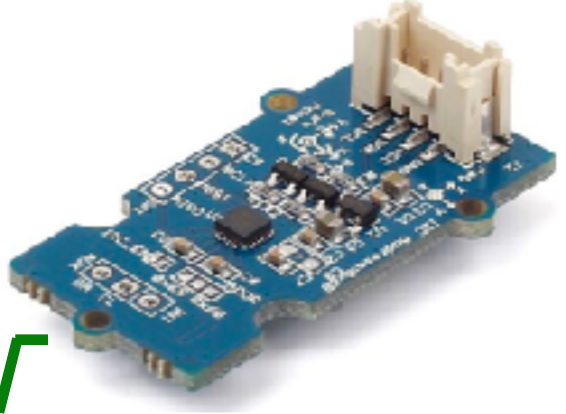
Find the correct poses

$$\mathbf{x}_t^\star, \mathbf{x}_{t+1}^\star = \arg\max_{\mathbf{x}_t, \mathbf{x}_{t+1}} \mathcal{N}(\mathbf{z}^v;\ \text{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t), \Sigma^v)$$

$$= \arg\min_{\mathbf{x}_t, \mathbf{x}_{t+1}} \|\text{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}^v\|_{\Sigma^v}^2$$

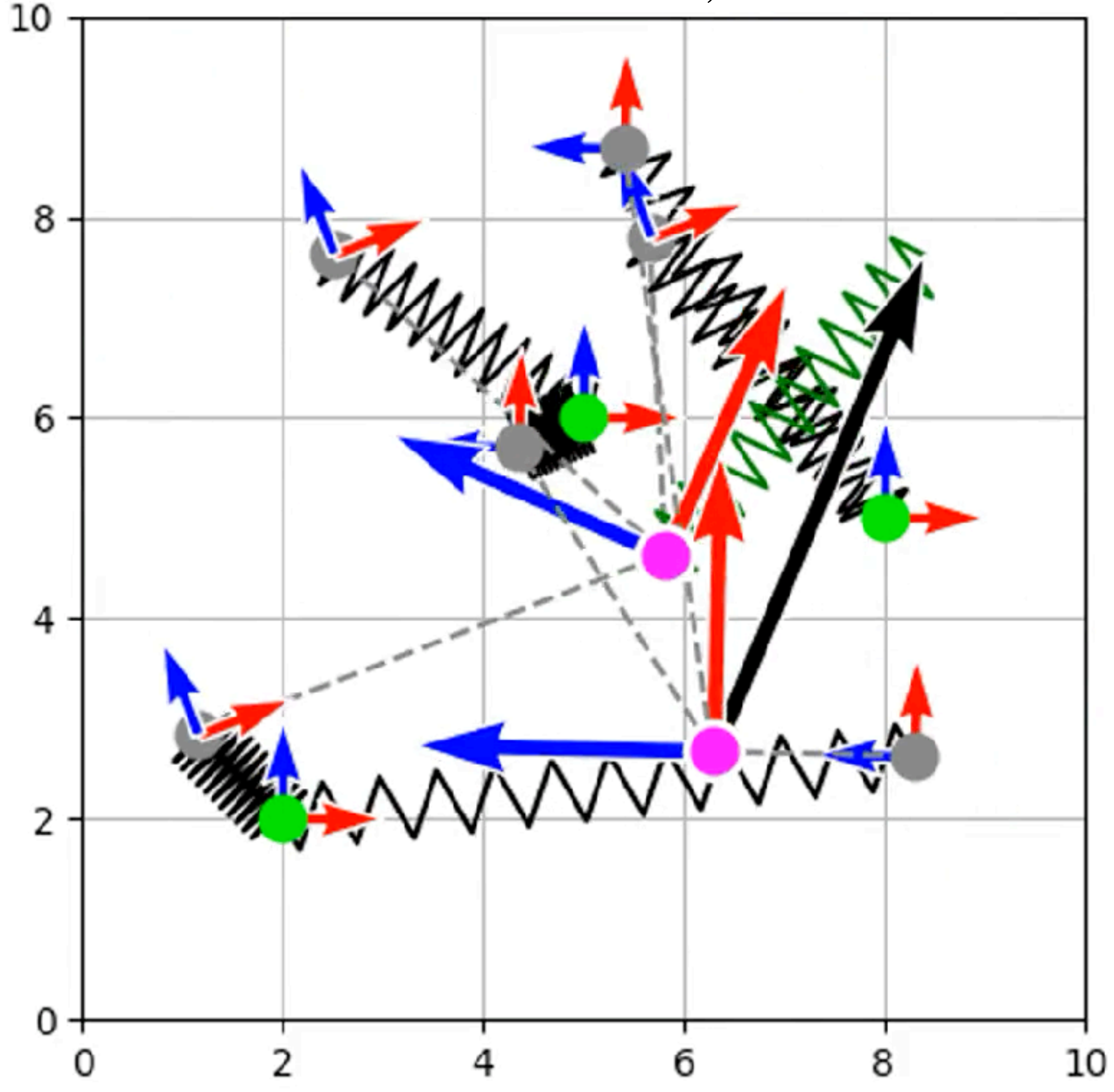
Localization

3D marker detector (RGBD camera) +



Odometry (IMU)

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}_t} \sum_{i,t} \|\mathrm{w2r}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m}_i}\|^2 + \|\mathrm{w2r}(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{z}_{12}^{odom}\|^2$$



$\mathbf{x}_t$ … robot poses

$\mathbf{m}_i$ … known marker positions

$\mathbf{z}_t^{\mathbf{m}_i}$ … marker measurements

…… local coordinate frame

…… odometry

$\sum_{i,t} \|\mathrm{w2r}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m}_i}\|^2$ … marker loss

$\|\mathrm{w2r}(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{z}_{12}^{odom}\|^2$ … odom loss

Localization of robot in wcf from known marker pose, odometry and GPS

GPS | odometry | marker

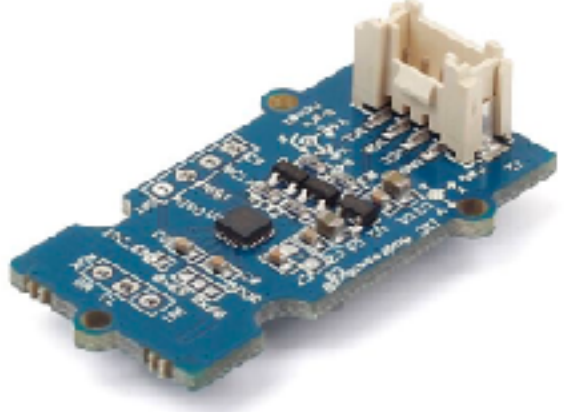
$$\mathbf{x}^\star = \arg\max_{\mathbf{x}_0,\dots\mathbf{x}_t} \prod_t p(\mathbf{z}_t^{gps} \,|\, \mathbf{x}_t) \cdot \prod_t p(\mathbf{z}_t^{v} \,|\, \mathbf{x}_t, \mathbf{x}_{t-1}) \cdot \prod_t p(\mathbf{z}_t^{m} \,|\, \mathbf{x}_t, \mathbf{m})$$

$$= \arg\max_{\mathbf{x}_0,\dots\mathbf{x}_t} \prod_t \mathcal{N}(\mathbf{z}^{gps};\; \mathbf{x}_t, \Sigma_t^{gps}) \cdot \prod_t \mathcal{N}(\mathbf{z}^{v};\; \mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t), \Sigma_t^{v}) \cdot \prod_t \mathcal{N}(\mathbf{z}^{m};\; \mathrm{w2r}(\mathbf{m}, \mathbf{x}_t), \Sigma_t^{m})$$

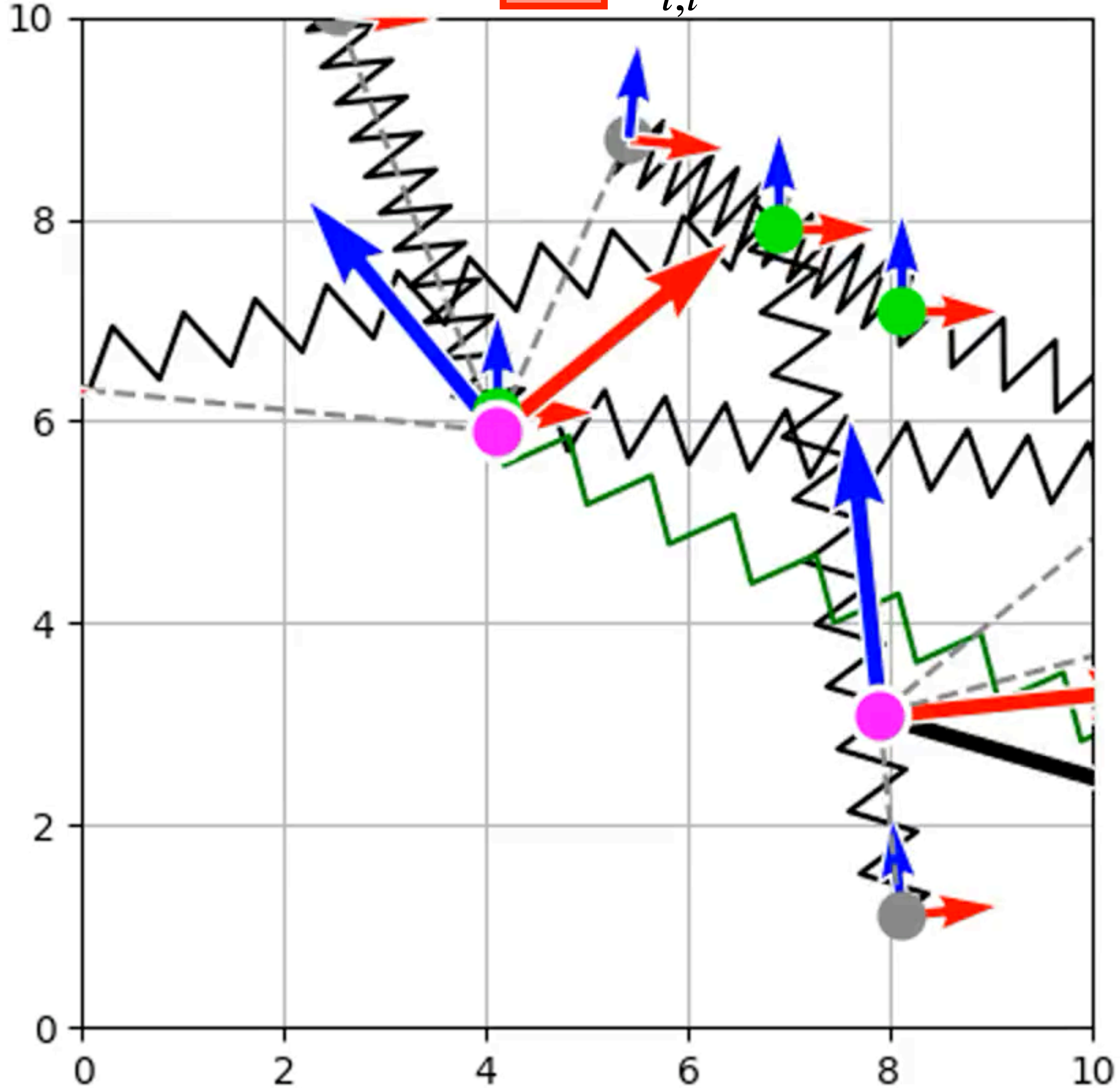
$$= \arg\min_{\substack{\mathbf{x}_0,\dots\mathbf{x}_T \\ \mathbf{m}^1\dots\mathbf{m}^J}} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}_t^{v}\|^2_{\Sigma_t^{v}} + \sum_t \|\mathrm{w2r}(\mathbf{m}, \mathbf{x}_t) - \mathbf{z}^{m}\|^2_{\Sigma_t^{m}}$$

Localization => SLAM

Localization

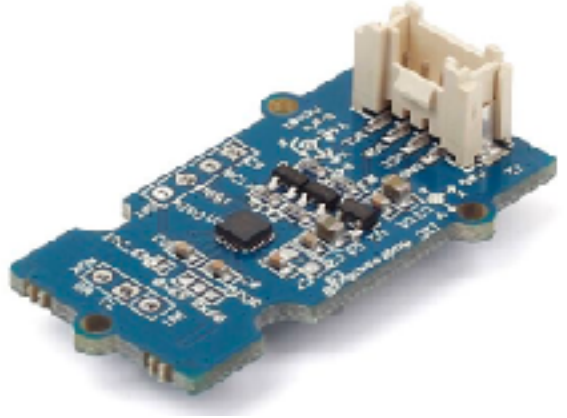
3D marker detector (RGBD camera) +  Odometry (IMU)

$$\mathbf{x}^\star = \arg\min_{\mathbf{m}_i, \mathbf{x}_t} \sum_{i,t} \|\mathrm{w2r}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m}_i}\|^2 + \|\mathrm{w2r}(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{z}_{12}^{odom}\|^2$$



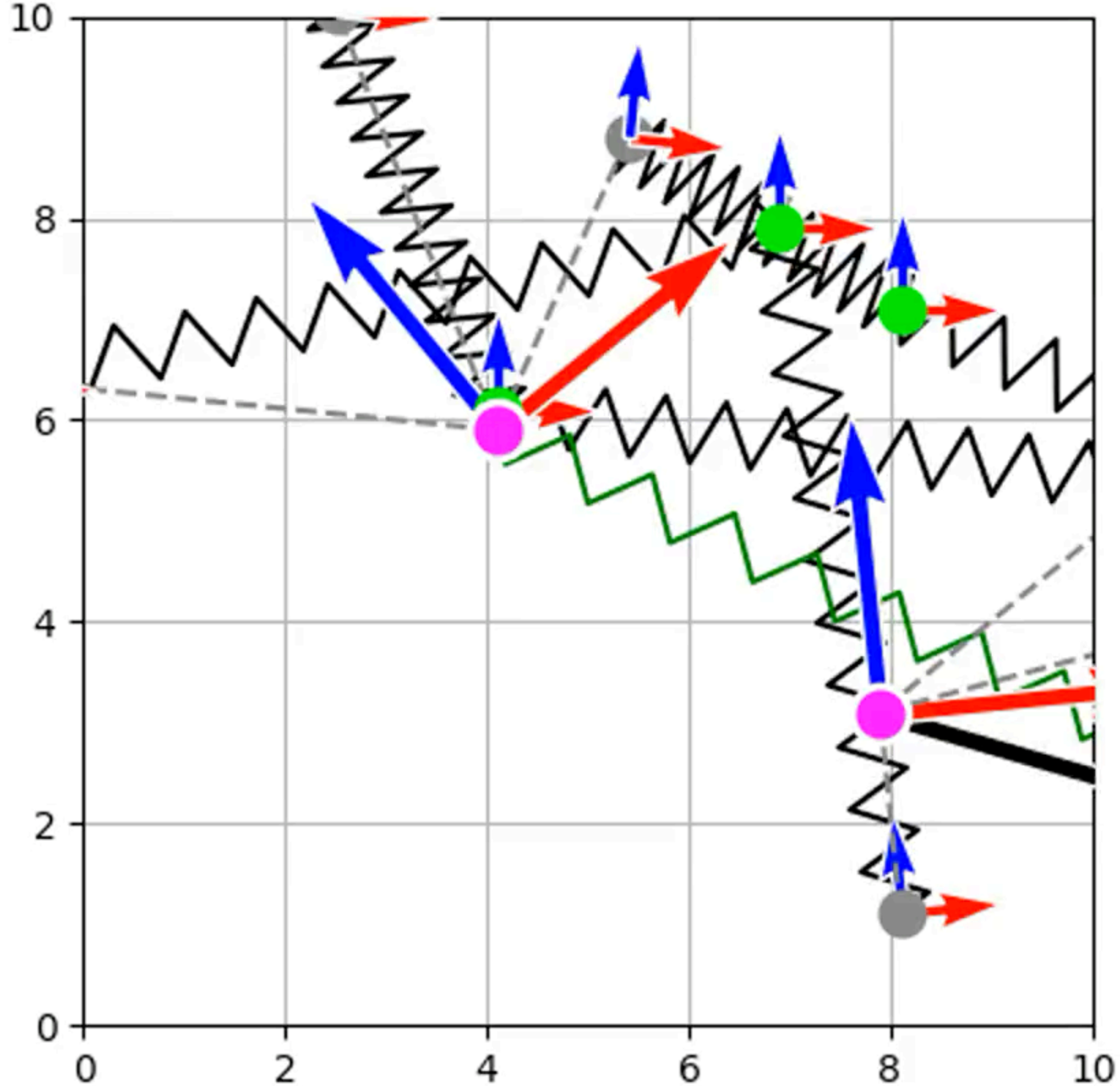
- $\mathbf{x}_t$ … robot poses
- $\mathbf{m}_i$ … known marker positions
- $\mathbf{z}_t^{\mathbf{m}_i}$ … marker measurements
- …… local coordinate frame
- …… odometry
- $\sum_{i,t} \|\mathrm{w2r}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m}_i}\|^2$ … marker loss
- $\|\mathrm{w2r}(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{z}_{12}^{odom}\|^2$ … odom loss

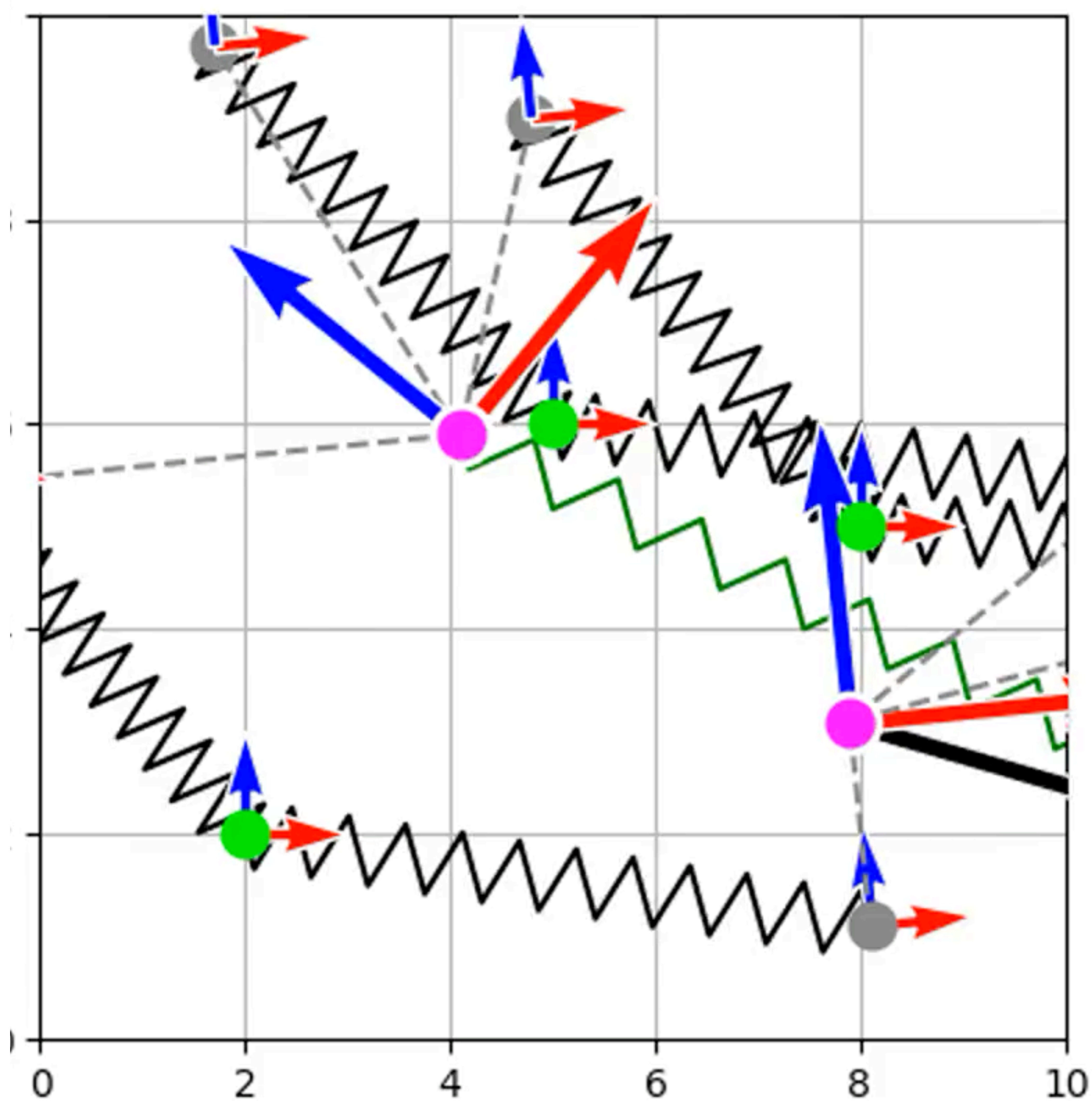
3D marker detector (RGBD camera) +



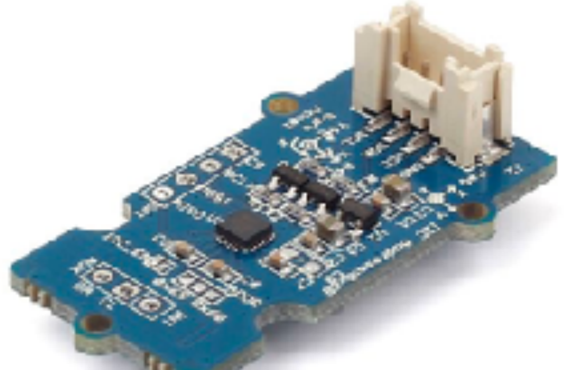
Odometry (IMU)

SLAM **Why is the result different?** Localization





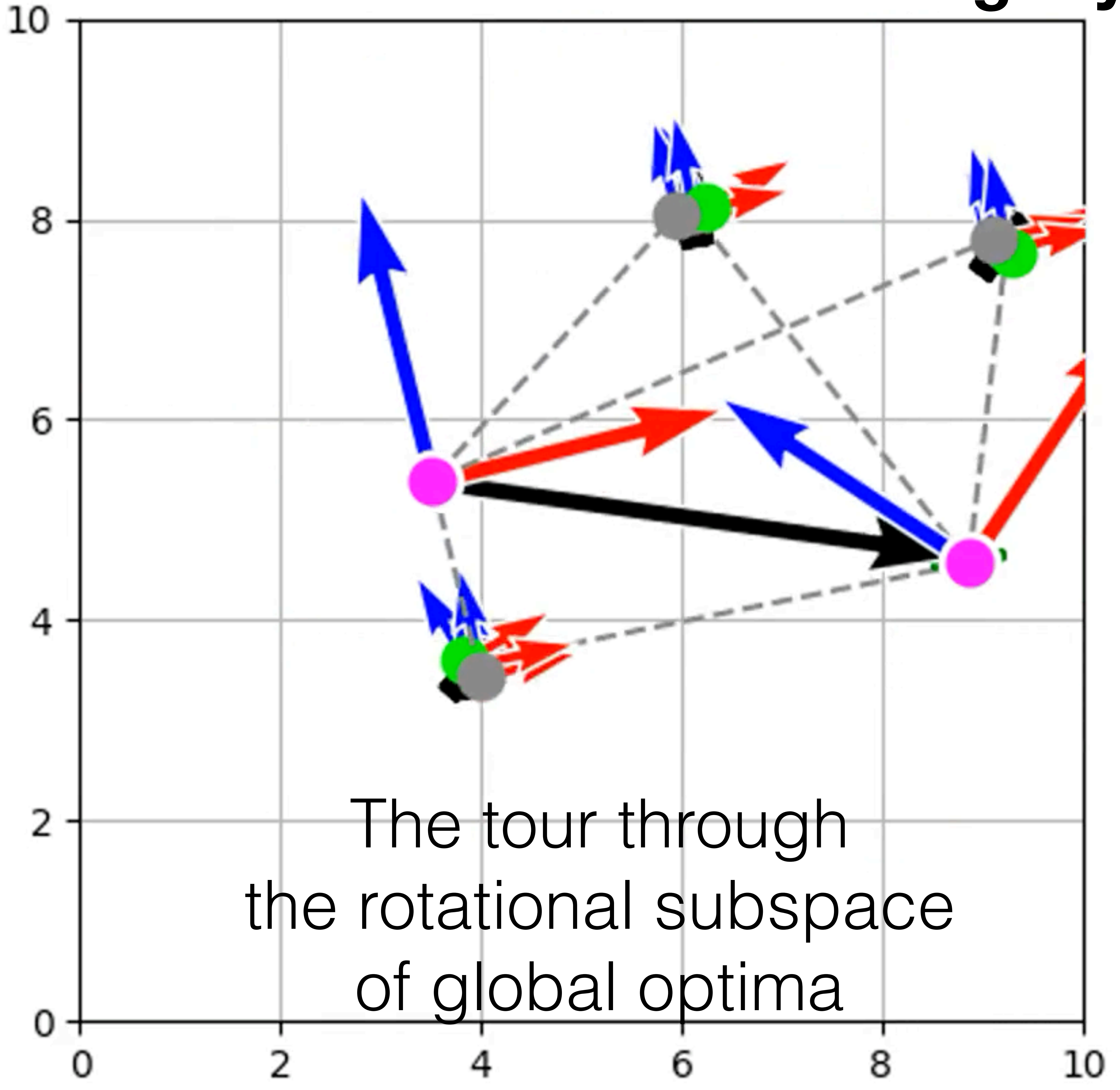
3D marker detector (RGBD camera) +



Odometry (IMU)

Nothing provides **absolute rotation+transl** => Global optimum is **3D subspace**

How can I resolve the ambiguity?

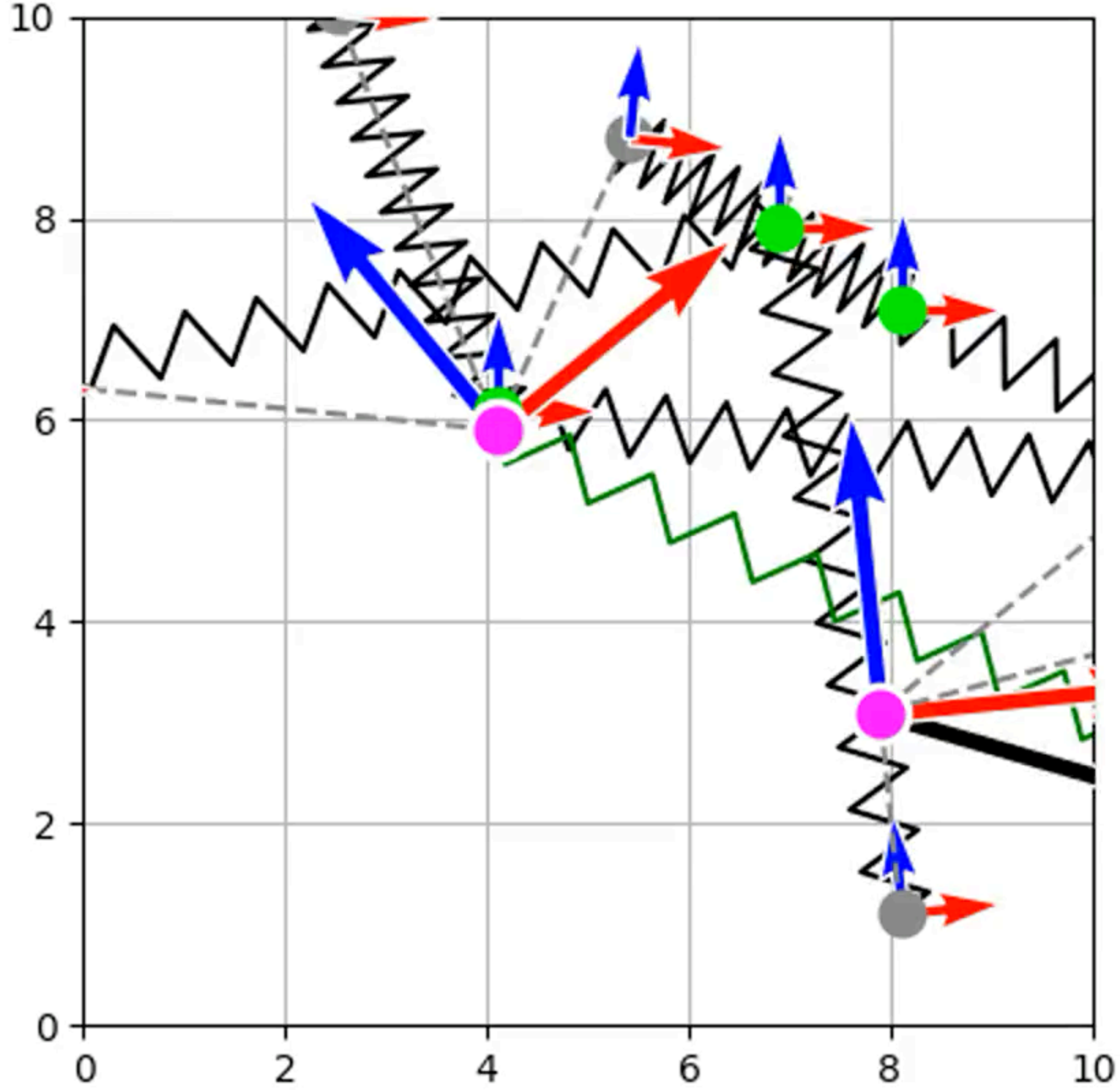


Removing the odometry does not matter

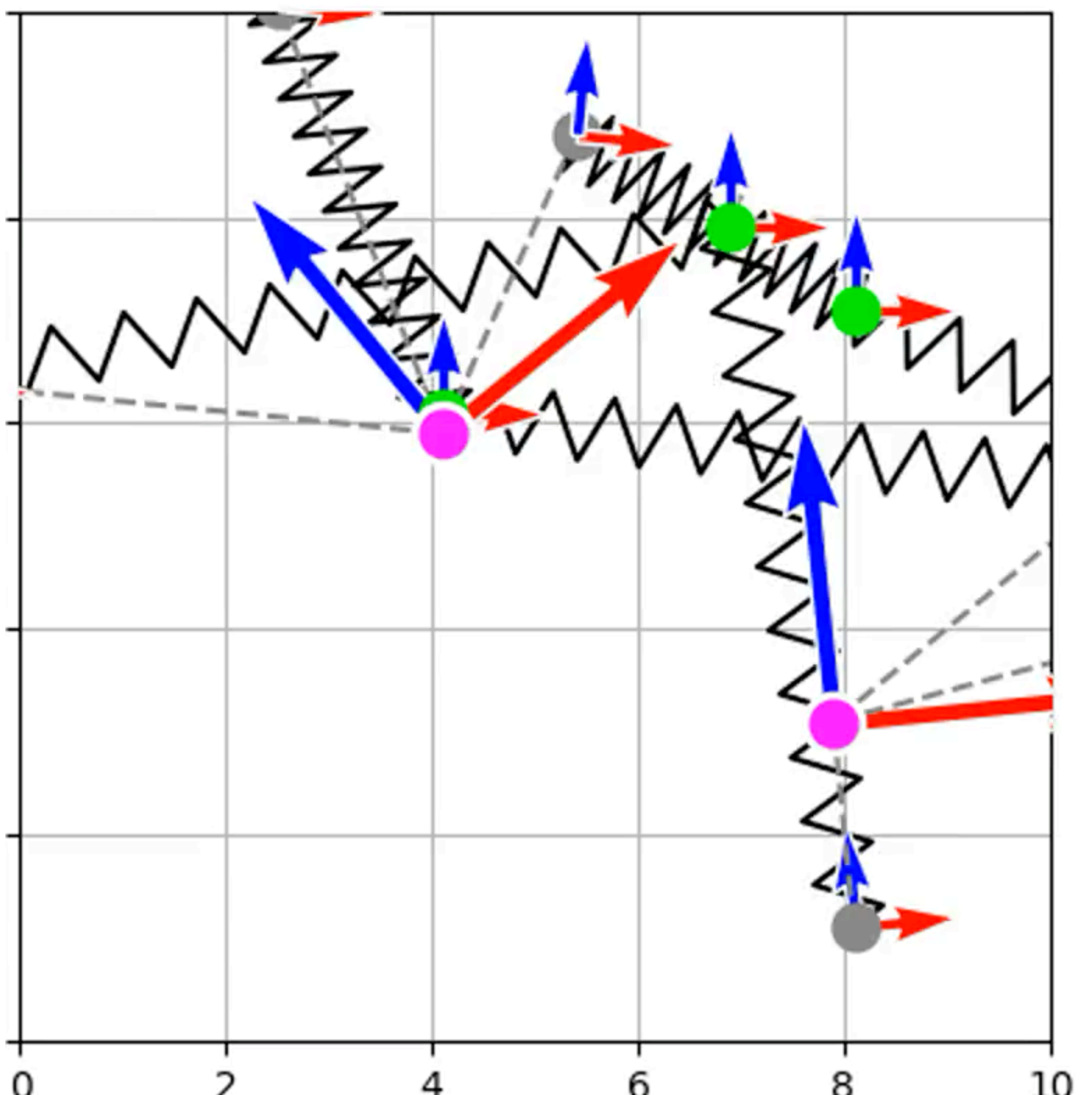
3D marker detector (RGBD camera) +

Odometry (IMU)

SLAM **with** odometry



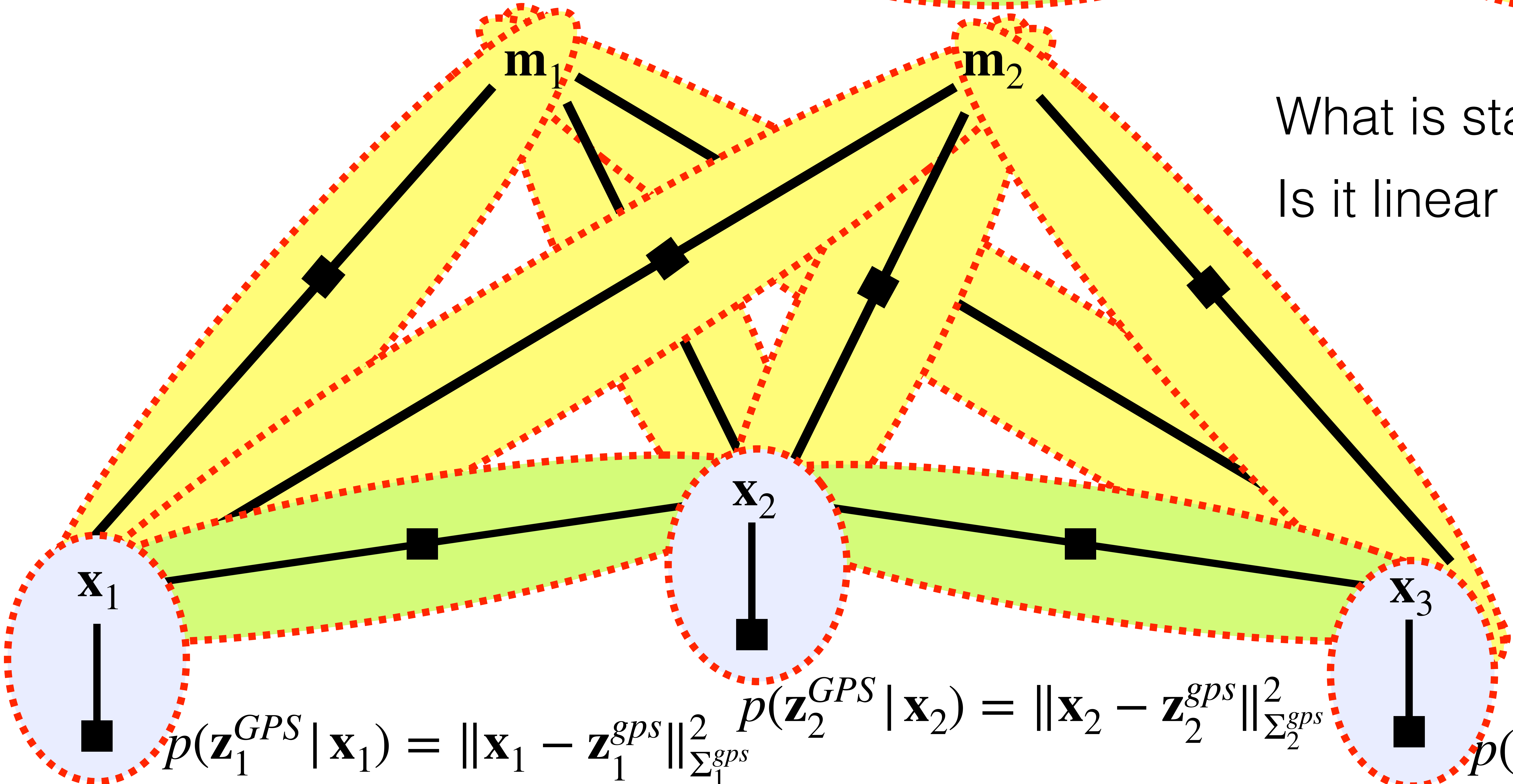
SLAM **without** odometry



Factorgraph

GPS — unary; odometry — pair-wise; 3D marker(s) — pair-wise

$$= \arg\min_{\substack{\mathbf{x}_0,\ldots\mathbf{x}_T \\ \mathbf{m}^1\ldots\mathbf{m}^J}} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}_t^{v}\|^2_{\Sigma_t^{v}} + \sum_{t,j} \|\mathrm{w2r}(\mathbf{m}^j, \mathbf{x}_t) - \mathbf{z}\|^2_{\Sigma_t^{mj}}$$



What is state dimensionality?

Is it linear LS problem?

$p(\mathbf{z}_1^{GPS} \mid \mathbf{x}_1) = \|\mathbf{x}_1 - \mathbf{z}_1^{gps}\|^2_{\Sigma_1^{gps}}$

$p(\mathbf{z}_2^{GPS} \mid \mathbf{x}_2) = \|\mathbf{x}_2 - \mathbf{z}_2^{gps}\|^2_{\Sigma_2^{gps}}$

$p(\mathbf{z}_3^{GPS} \mid \mathbf{x}_3) = \|\mathbf{x}_3 - \mathbf{z}_3^{gp}$

Straightforward extensions

GPS — odometry — 3D marker(s)

$$= \arg\min_{\substack{\mathbf{x}_0,\ldots\mathbf{x}_T \\ \mathbf{m}^1\ldots\mathbf{m}^J}} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}^v\|^2_{\Sigma_t^v} + \sum_{t,j} \|\mathrm{w2r}(\mathbf{m}^j, \mathbf{x}_t) - \mathbf{z}\|^2_{\Sigma_t^{m^j}}$$

priors — loop-closures

$$+ \sum_t \|\mathbf{x}_t - \mathbf{x}_t^{prior}\|^2_{\Sigma_t^{prior}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_0, \mathbf{x}_T)\|^2_{\Sigma_t^{lc}}$$

Straightforward extensions

GPS — odometry — 3D marker(s)

$$= \arg\min_{\substack{\mathbf{x}_0,\ldots\mathbf{x}_T \\ \mathbf{m}^1\ldots\mathbf{m}^J}} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}^v\|^2_{\Sigma_t^v} + \sum_{t,j} \|\mathrm{w2r}(\mathbf{m}^j, \mathbf{x}_t) - \mathbf{z}\|^2_{\Sigma_t^{mj}}$$

priors — loop-closures

$$+ \sum_t \|\mathbf{x}_t - \mathbf{x}_t^{prior}\|^2_{\Sigma_t^{prior}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_0, \mathbf{x}_T)\|^2_{\Sigma_t^{lc}}$$

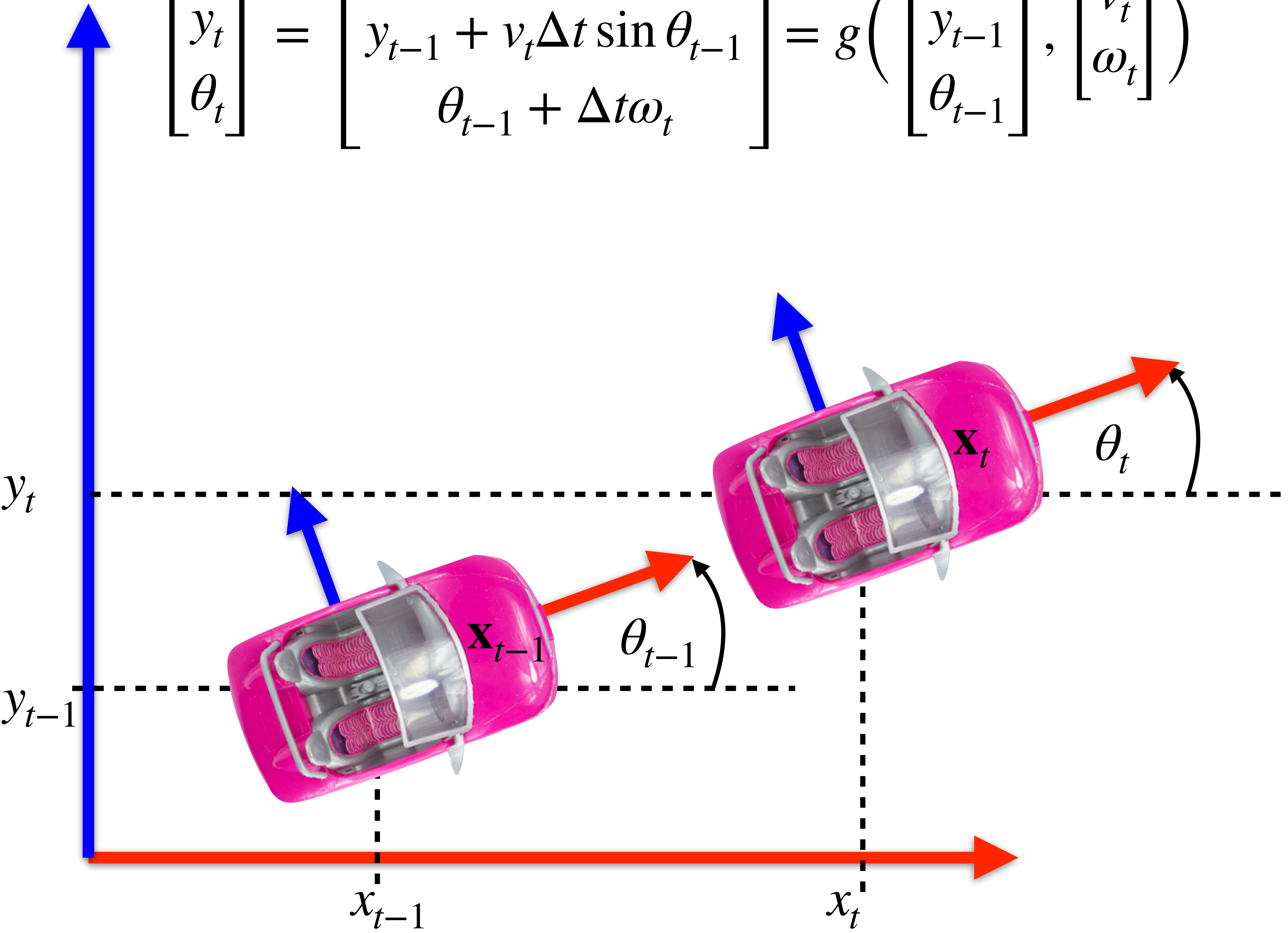
motion model — UWB — 2D marker(s)

\+ ??? + ??? + ???

e.g. camera detections

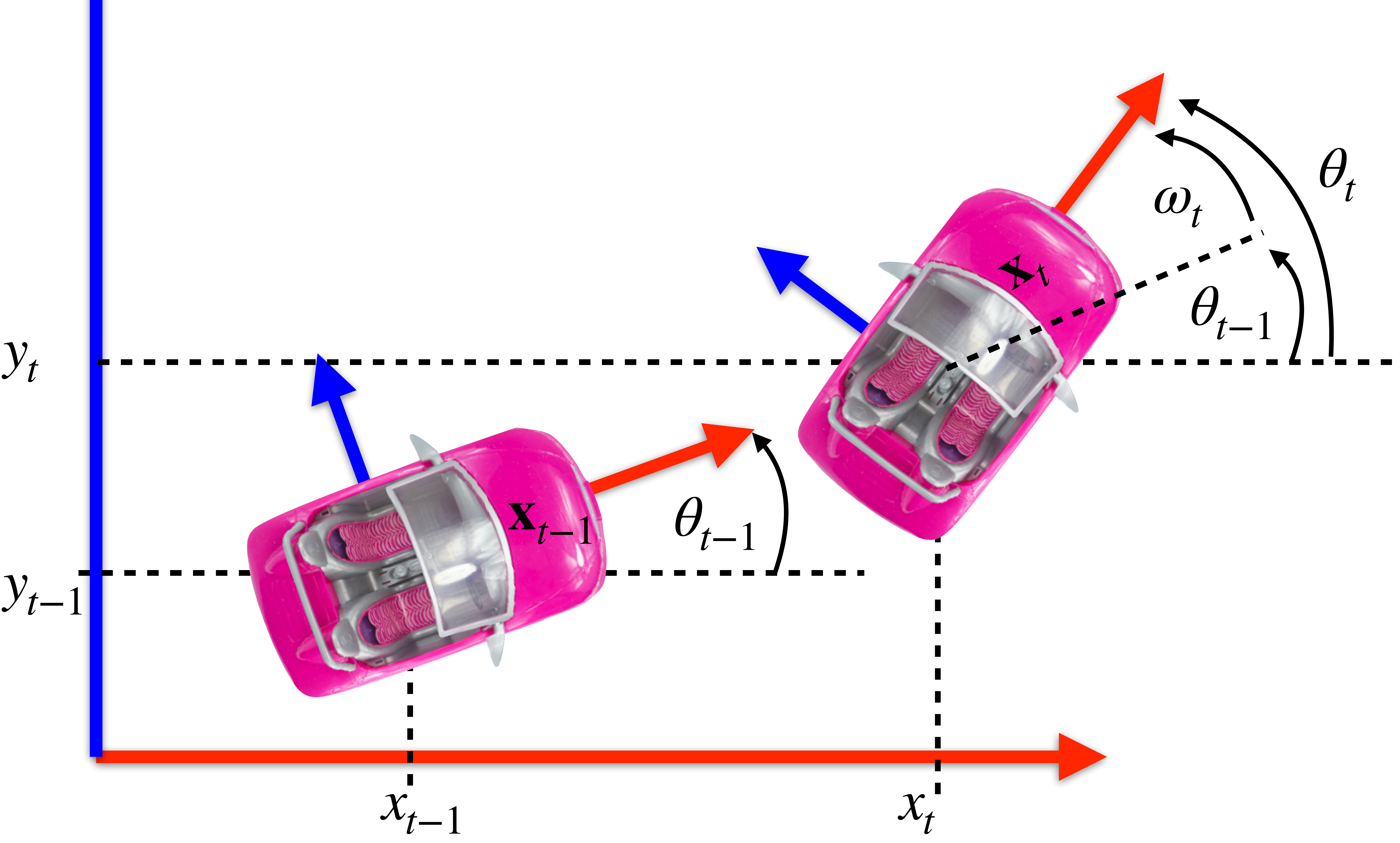
Differential drive model

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos \theta_{t-1} \\ y_{t-1} + v_t \Delta t \sin \theta_{t-1} \\ \theta_{t-1} + \Delta t \omega_t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$



Differential drive model

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos \theta_{t-1} \\ y_{t-1} + v_t \Delta t \sin \theta_{t-1} \\ \theta_{t-1} + \Delta t \omega_t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$



Example: GPS + motion model

Turtlebot transition probability:

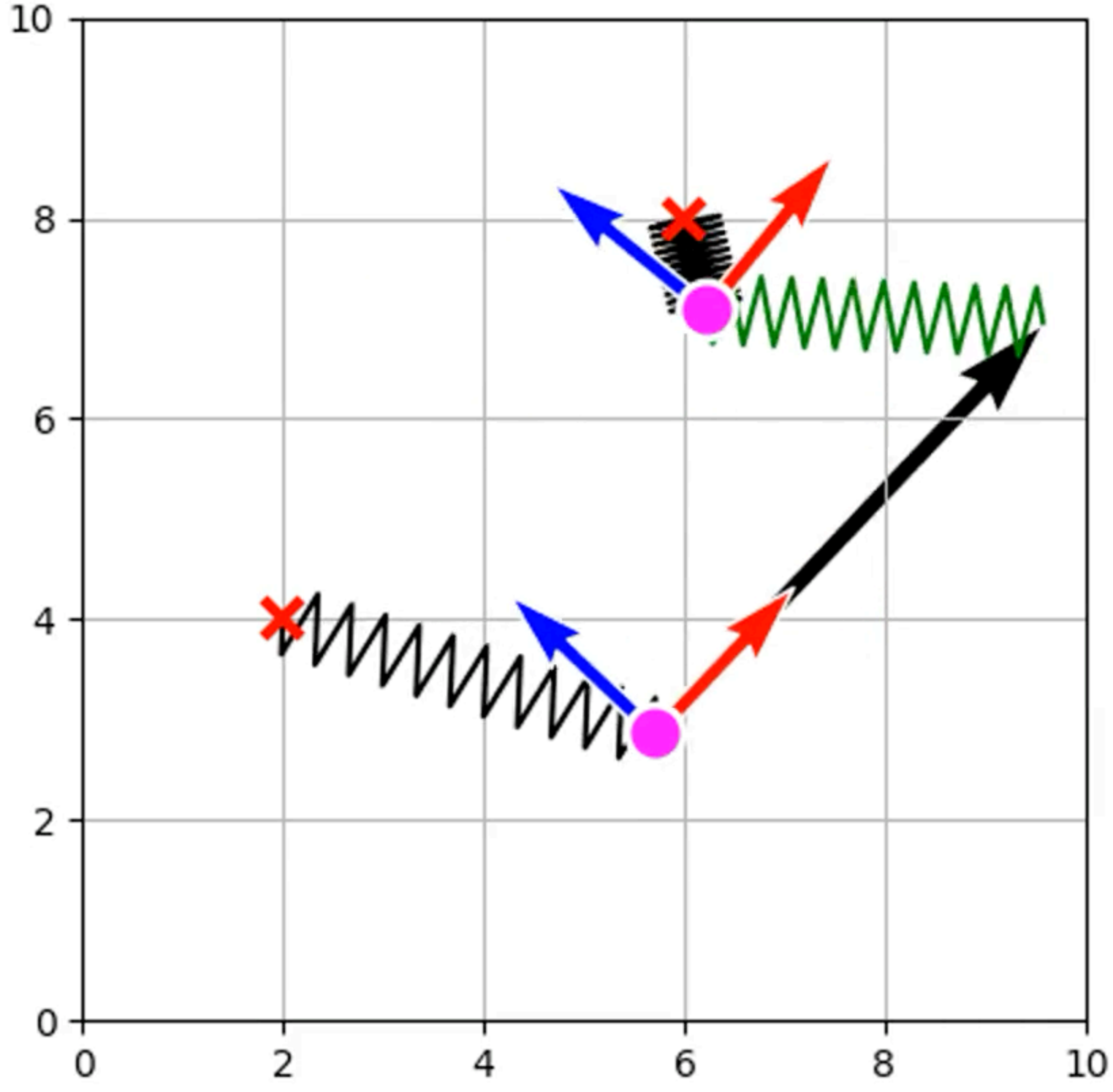
$$p\Big(\underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t} \Big| \underbrace{\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}}_{\mathbf{x}_{t-1}}, \underbrace{\begin{bmatrix} v_t \\ \omega_t \end{bmatrix}}_{\mathbf{u}_t}\Big) = \mathcal{N}\Big(\mathbf{x}_t; \underbrace{\begin{bmatrix} x_{t-1} + v_t \Delta t \cos\theta_{t-1} \\ y_{t-1} + v_t \Delta t \sin\theta_{t-1} \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix}}_{g(\mathbf{u}_t, \mathbf{x}_{t-1})}, \quad \mathbf{R}_t\Big)$$

GPS measurement probability:

$$p\Big(\underbrace{\begin{bmatrix} z_t^x \\ z_t^y \end{bmatrix}}_{\mathbf{z}_t^{\text{GPS}}} \Big| \underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t}\Big) = \mathcal{N}\Big(\mathbf{z}_t^{\text{GPS}}; \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{h(\mathbf{x}_t)}, \quad Q_t^{\text{GPS}}\Big)$$

Example: GPS + motion model

$$= \arg\min_{\mathbf{x}_0,\ldots\mathbf{x}_T} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} \quad + \quad \sum_t \|g(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t^g}$$



$\mathbf{x}_t$ … robot poses

$\mathbf{z}_t^{gps}$ … gps measurements

…… local coordinate frame

$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t)$ …… motion model

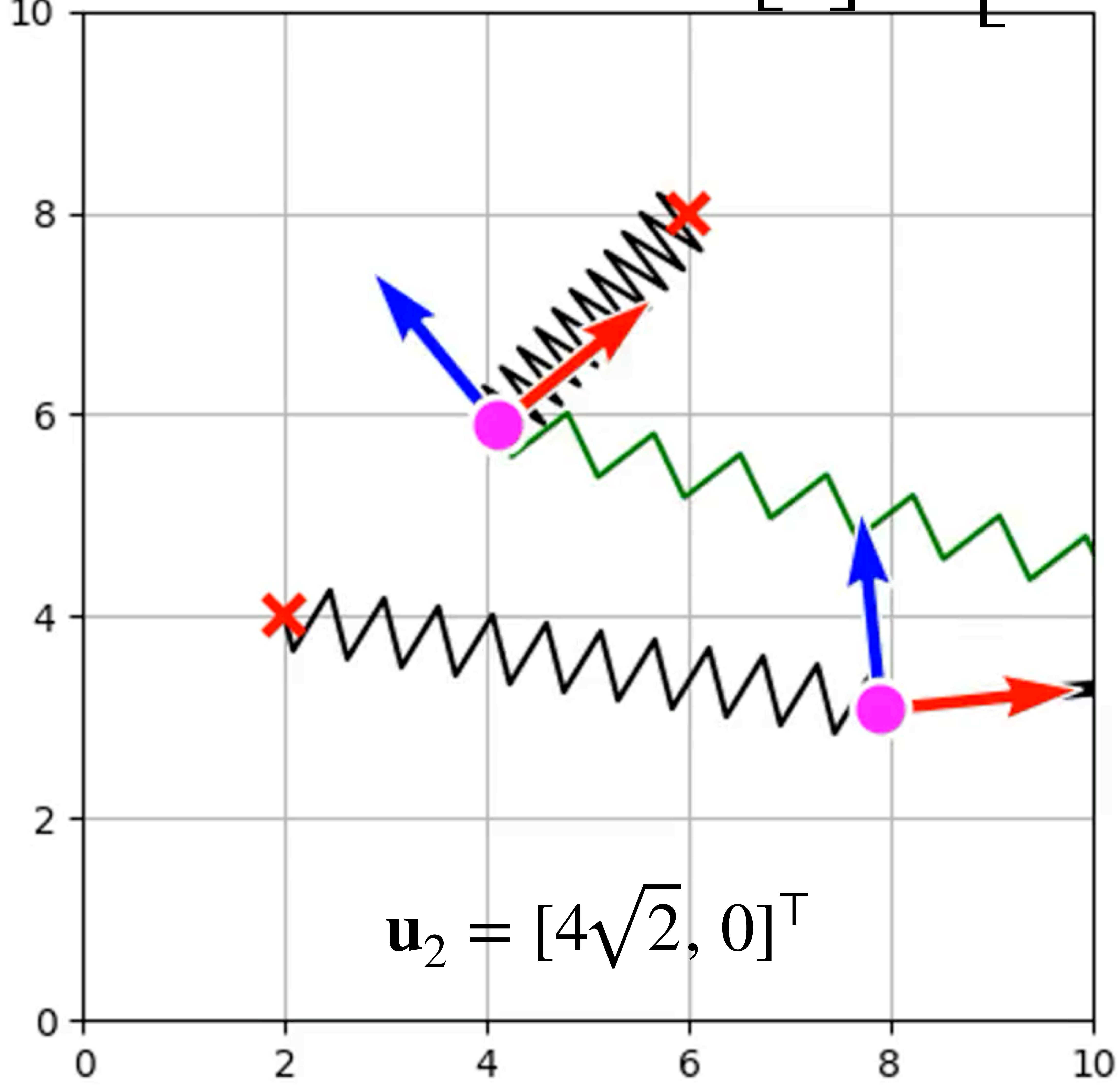
$\mathbf{u}_2 = [4\sqrt{2}, 0]^\top$

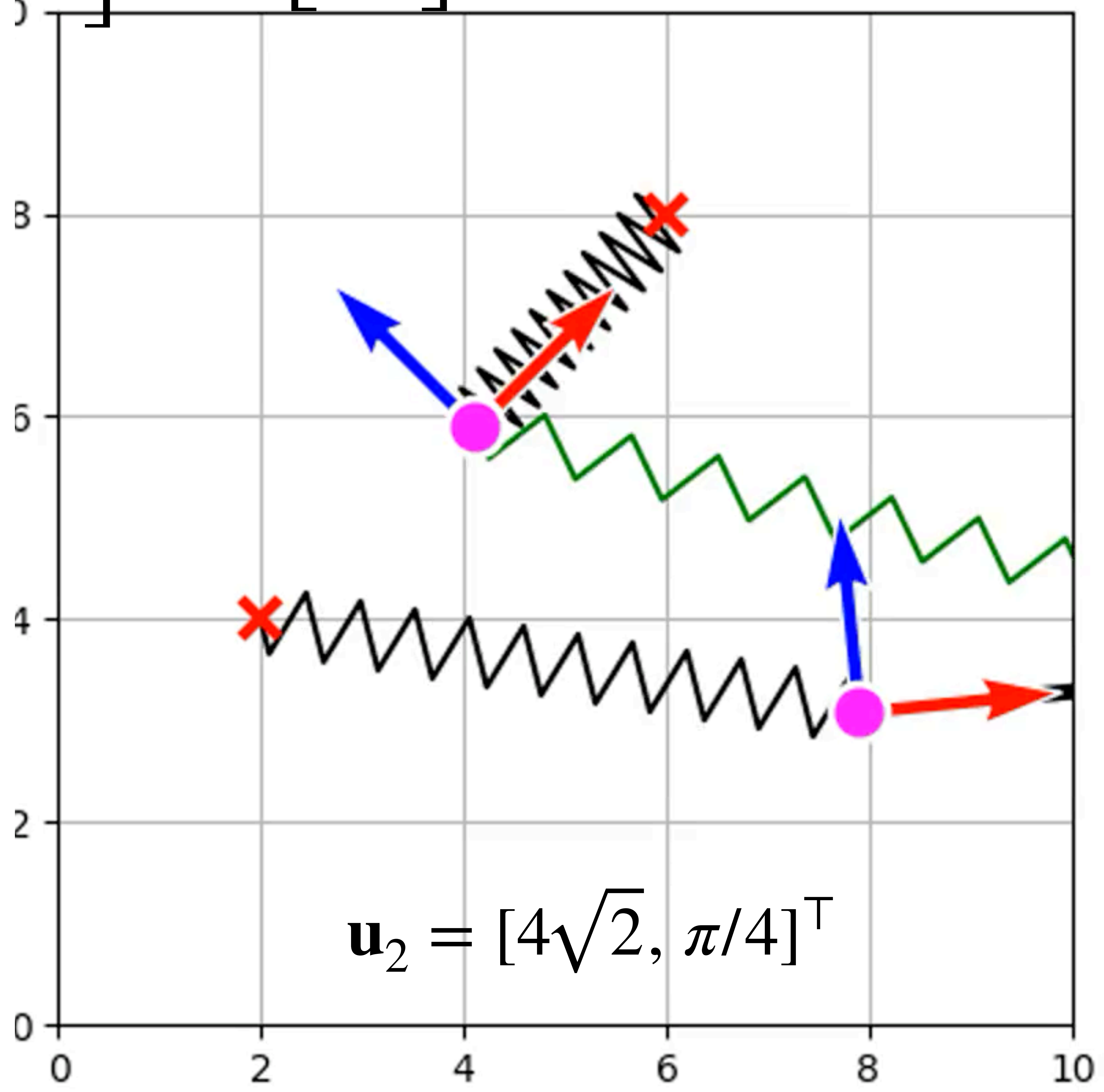
$\|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}}$ ………………. gps loss

$\|g(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t^g}$ …motion model loss

Example: GPS + motion model

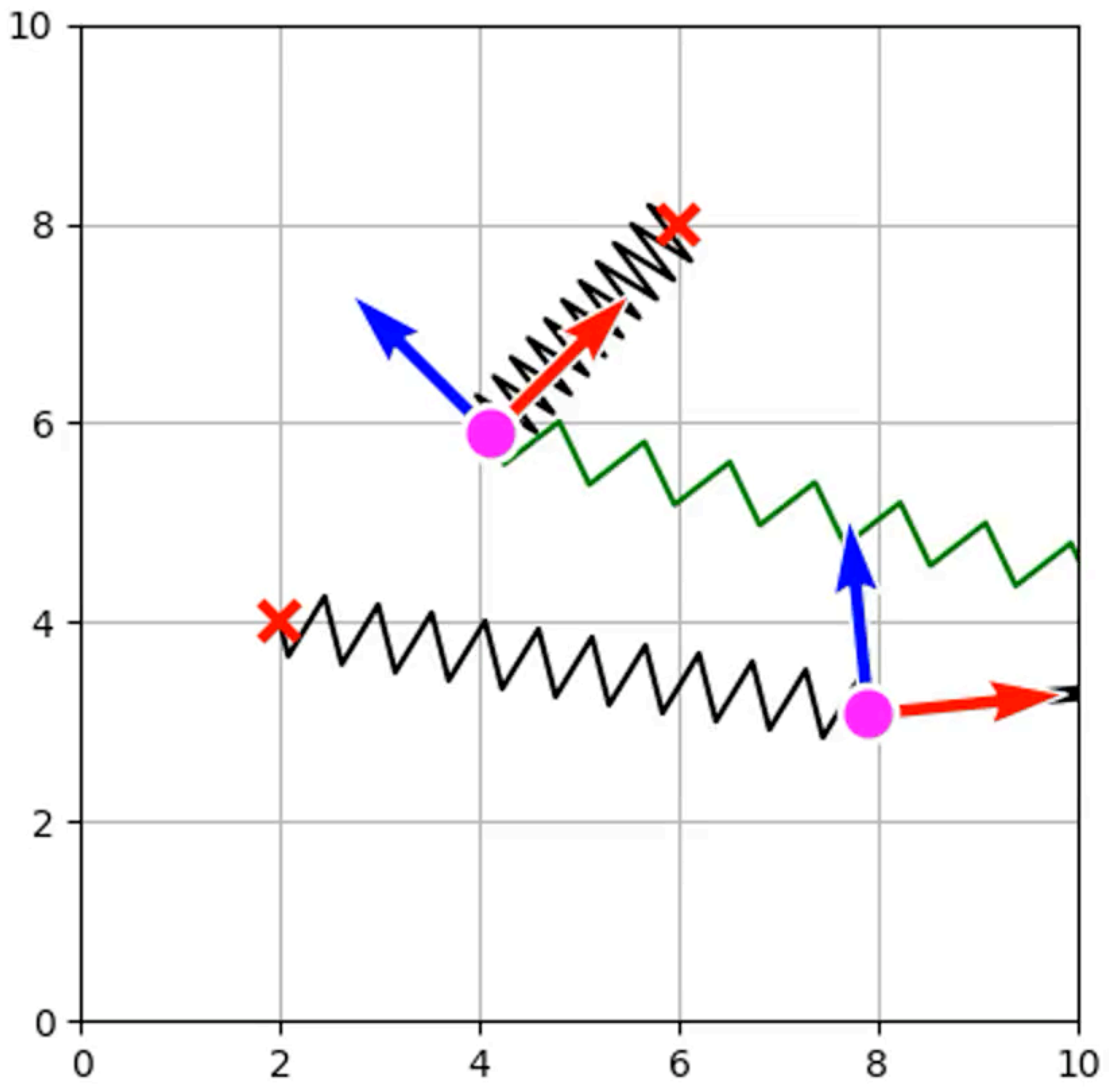
$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos \theta_{t-1} \\ y_{t-1} + v_t \Delta t \sin \theta_{t-1} \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$





$\mathbf{u}_2 = [4\sqrt{2}, \pi/4]^\top$

Example: GPS + motion model



Move then turn

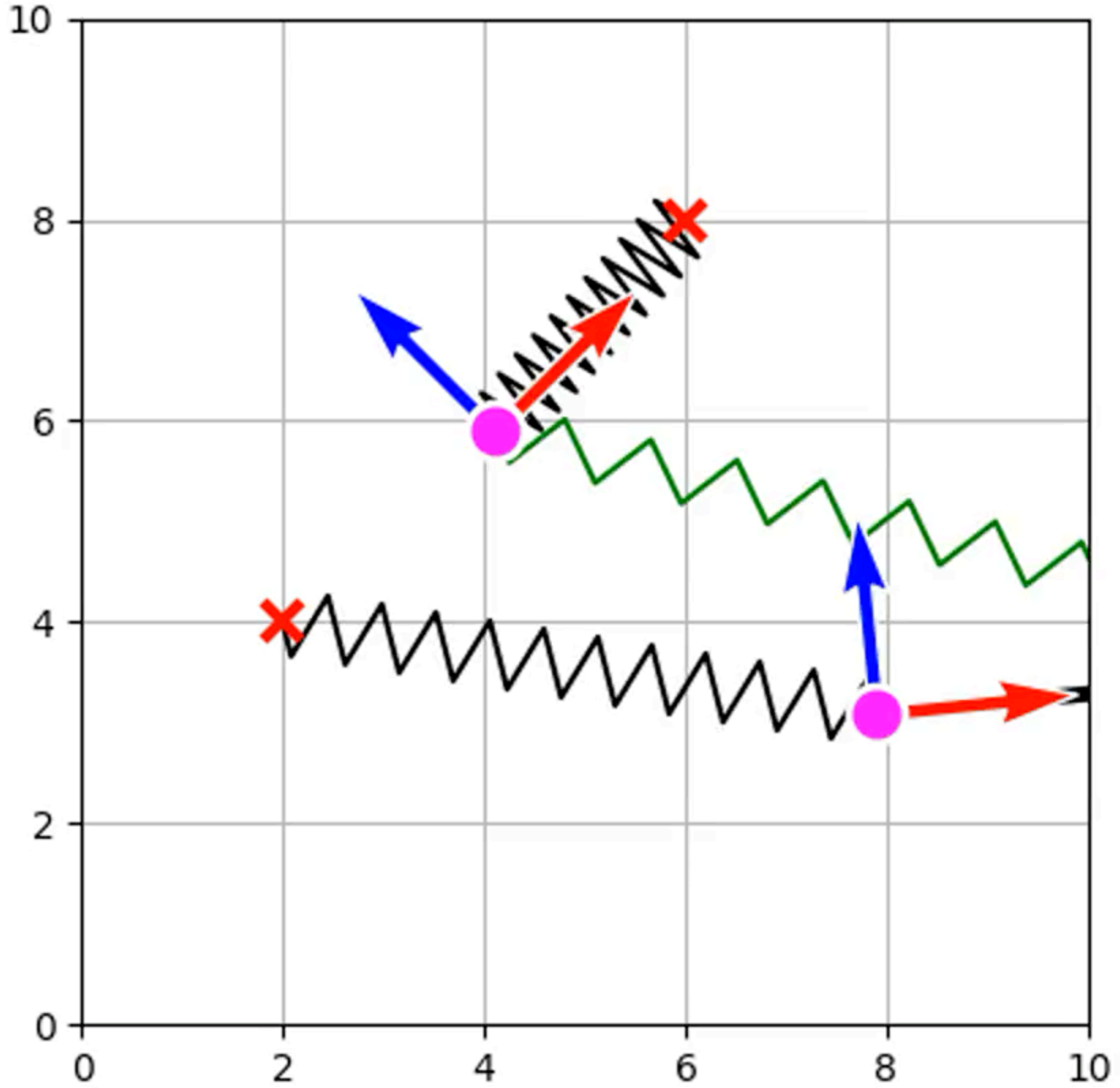
$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos \theta_{t-1} \\ y_{t-1} + v_t \Delta t \sin \theta_{t-1} \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$

Turn then move

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos(\theta_{t-1} + \omega_t \Delta t) \\ y_{t-1} + v_t \Delta t \sin(\theta_{t-1} + \omega_t \Delta t) \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$

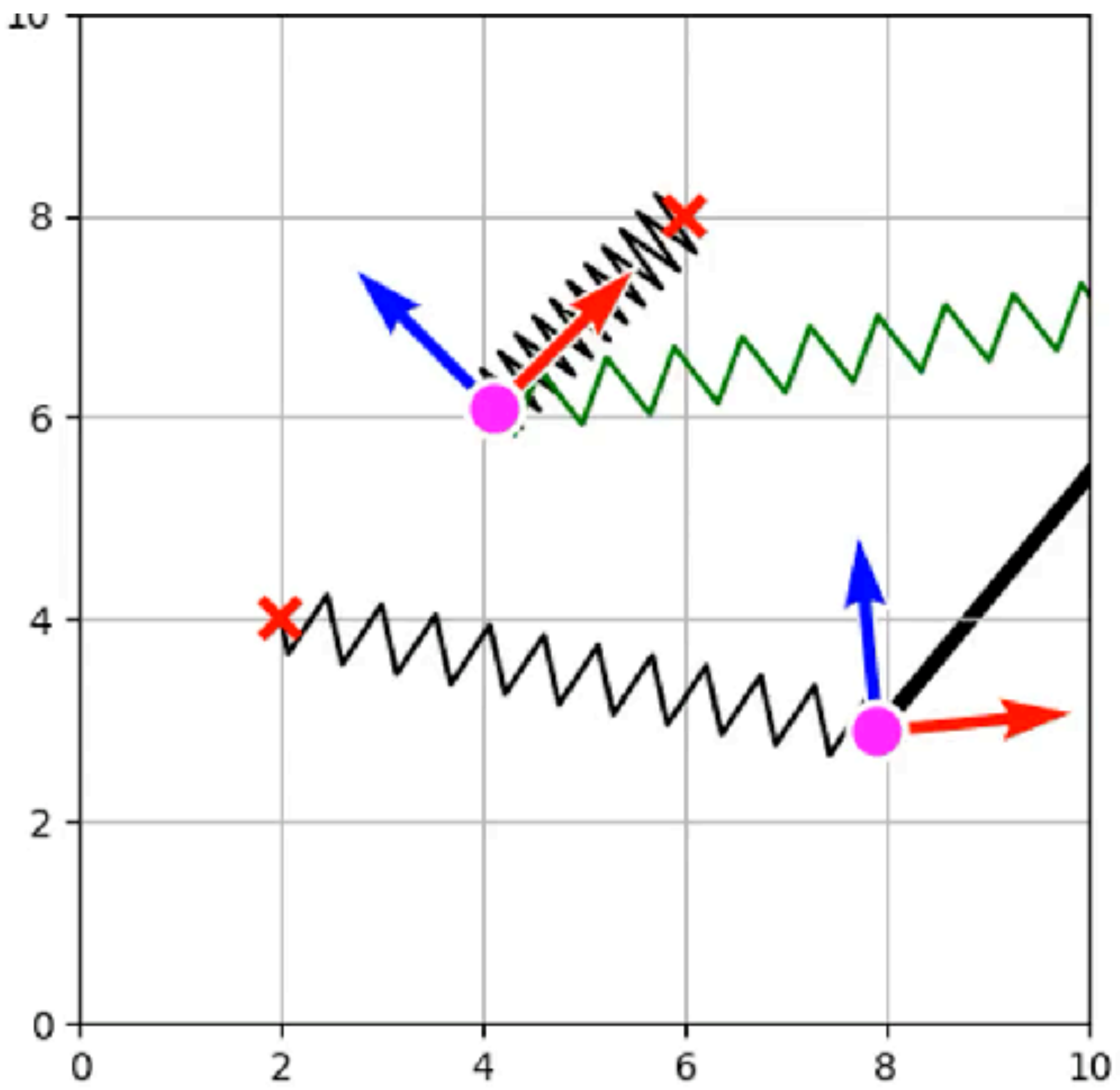
$\mathbf{u}_2 = [4\sqrt{2}, \pi/4]^\top$

Example: GPS + motion model



Move then turn

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos \theta_{t-1} \\ y_{t-1} + v_t \Delta t \sin \theta_{t-1} \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$

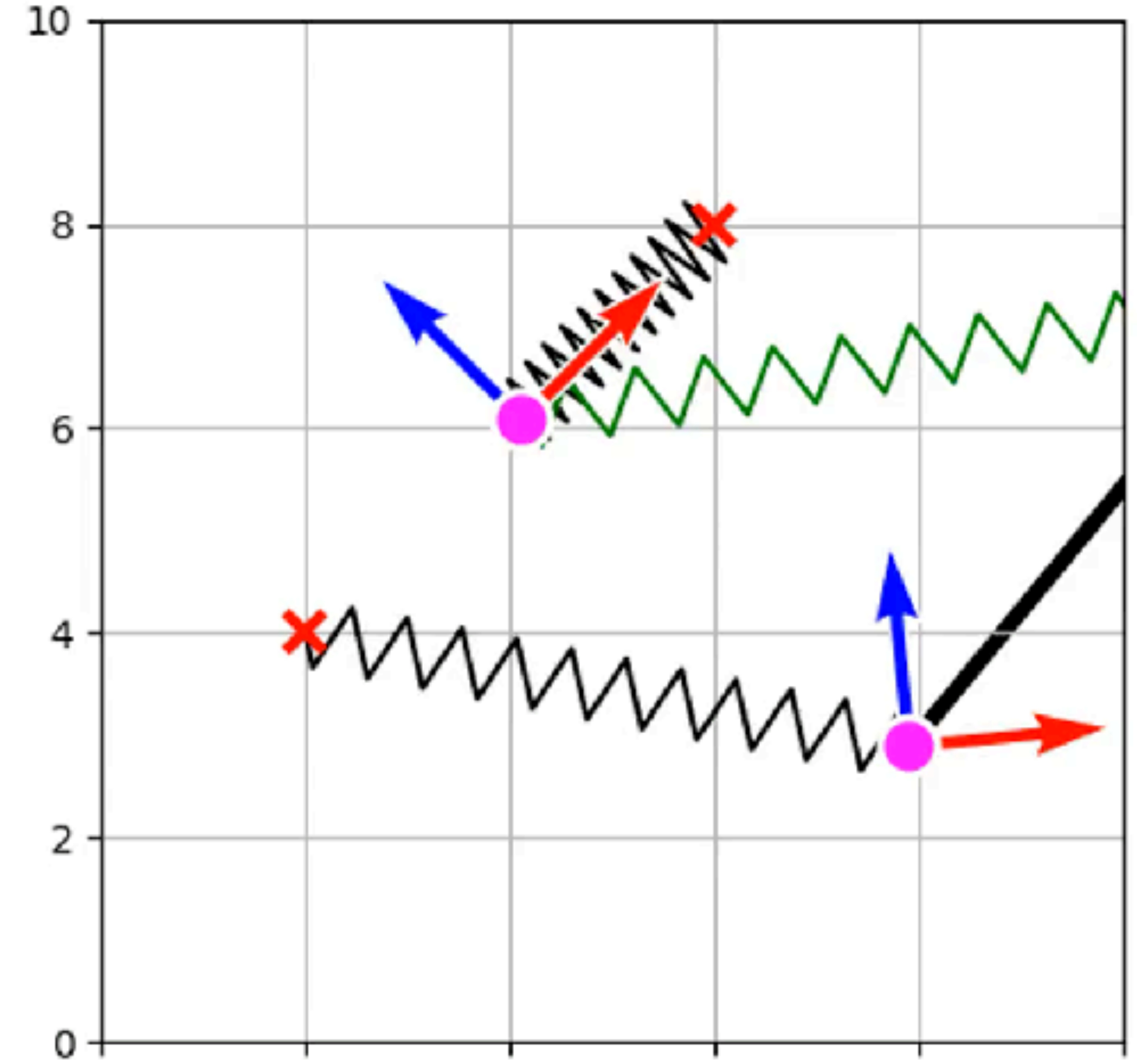


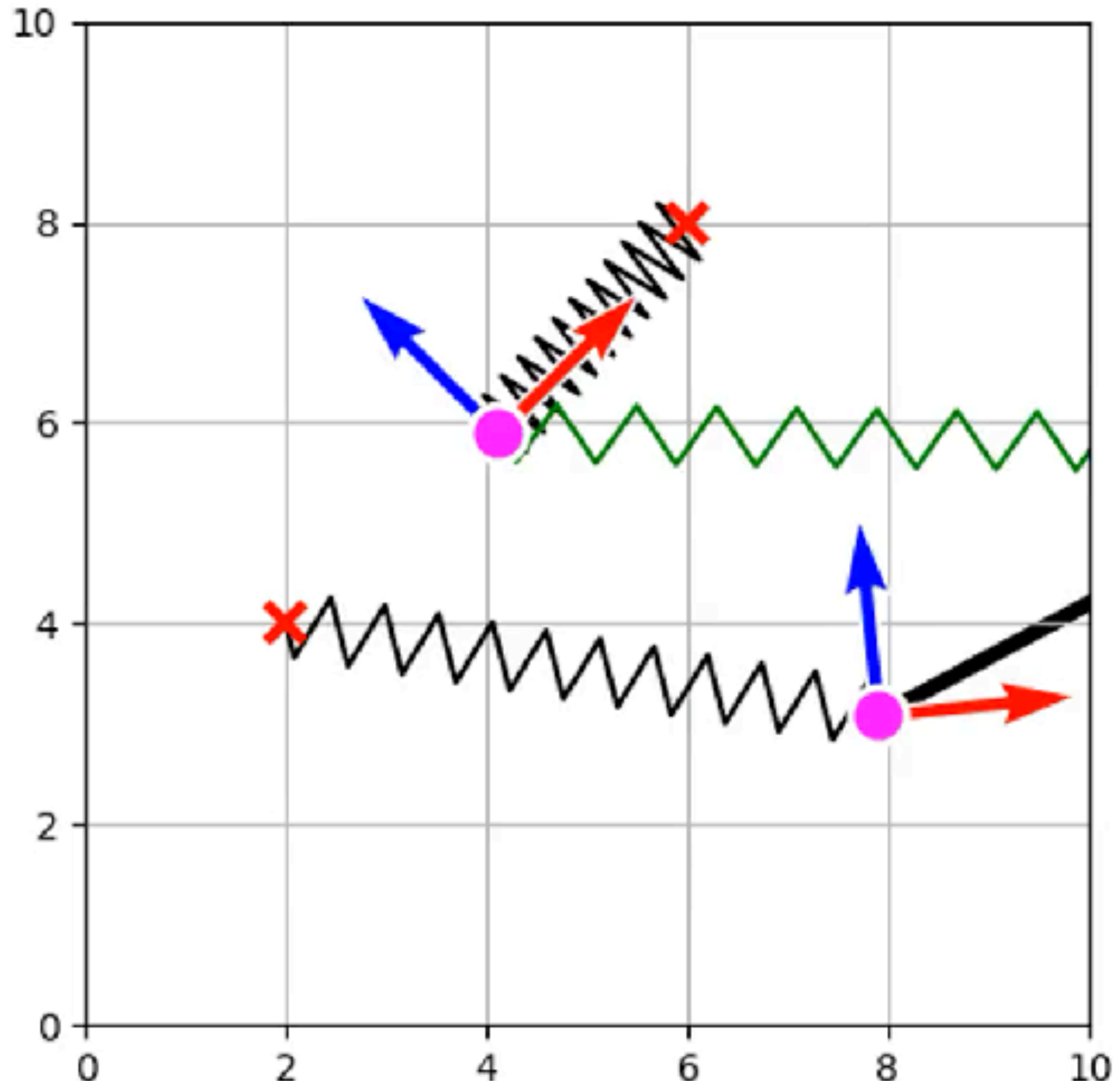
How do I get the best of both worlds?

Turn then move

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos(\theta_{t-1} + \omega_t \Delta t) \\ y_{t-1} + v_t \Delta t \sin(\theta_{t-1} + \omega_t \Delta t) \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$

$$\mathbf{u}_2 = [4\sqrt{2}, \pi/4]^\top$$





Example: GPS + motion model

Turn then move

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos(\theta_{t-1} + \omega_t \Delta t) \\ y_{t-1} + v_t \Delta t \sin(\theta_{t-1} + \omega_t \Delta t) \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$

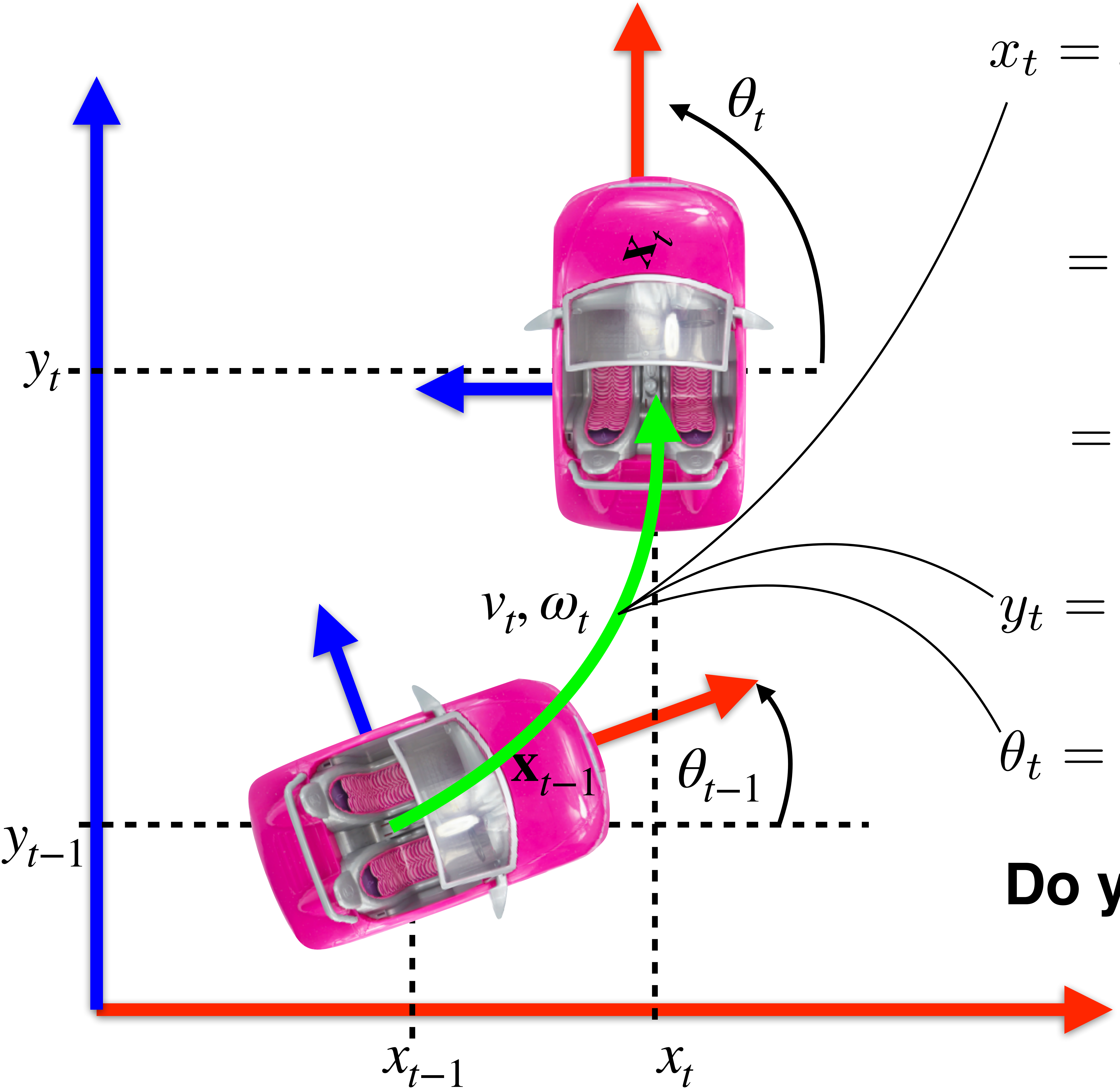
How do I get the best of both worlds?

Half-turn, then move, then half-turn

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos(\theta_{t-1} + \omega_t \Delta t/2) \\ y_{t-1} + v_t \Delta t \sin(\theta_{t-1} + \omega_t \Delta t/2) \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$

Is there even better way to do it?

Differential drive model



$$x_t = x_{t-1} + \int_0^{\Delta t} \overbrace{v_t \cos(\underbrace{\theta_{t-1} + \omega_t t}_{\theta(t)})dt}^{\dot{x}_t}$$

$$= x_{t-1} + v_t \left[\frac{\sin(\theta_{t-1} + \omega_t t)}{\omega_t} \right]_0^{\Delta t}$$

$$= x_{t-1} + \frac{v_t}{\omega_t}\left(+ \sin(\theta_{t-1} + \omega_t \Delta t) - \sin(\theta_{t-1})\right)$$

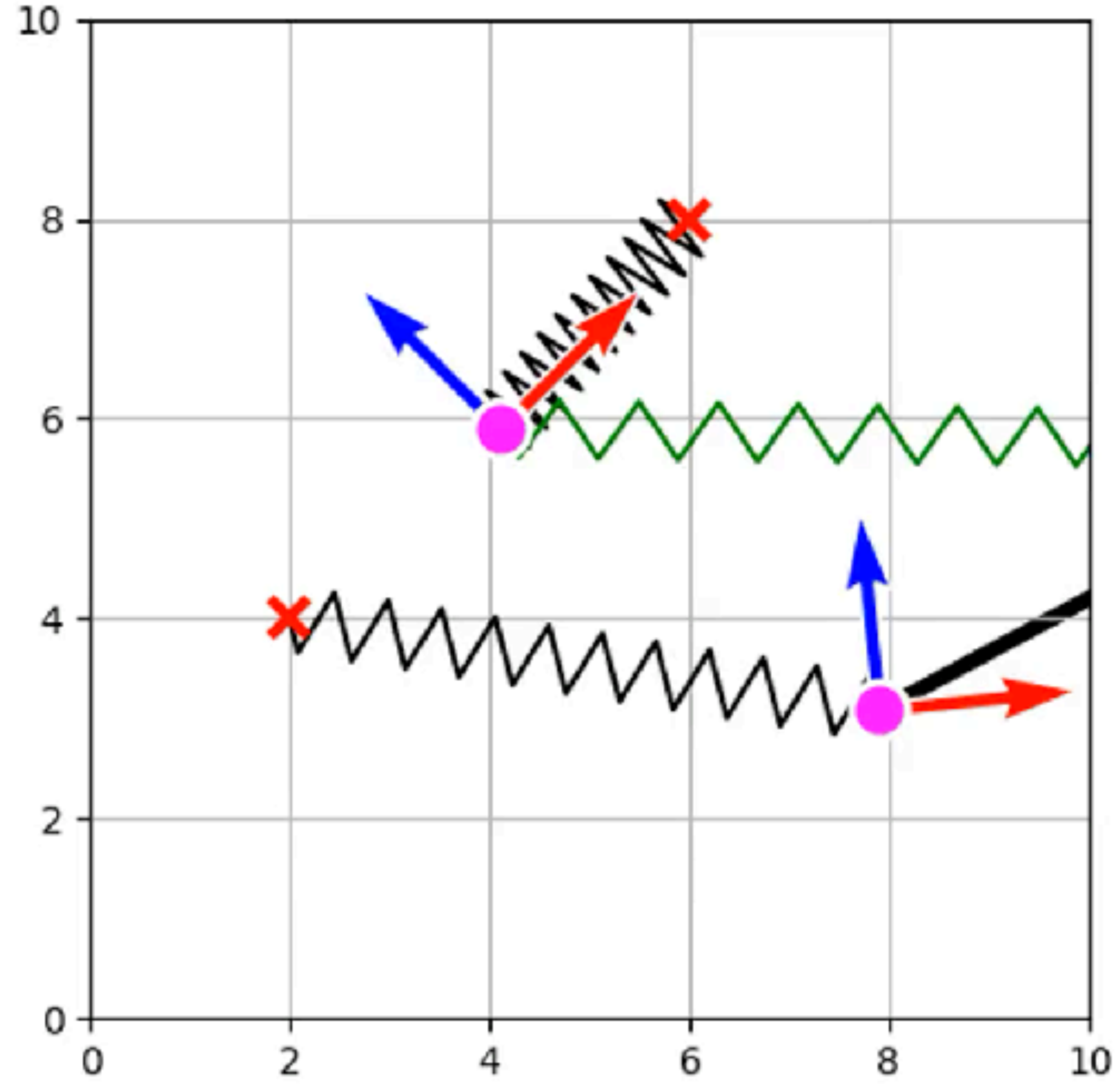
$$y_t = y_{t-1} + \frac{v_t}{\omega_t}\left(- \cos(\theta_{t-1} + \omega_t \Delta t) + \cos(\theta_{t-1})\right)$$

$$\theta_t = \theta_{t-1} + \omega \Delta t$$

Do you see any disadvantage of the model?

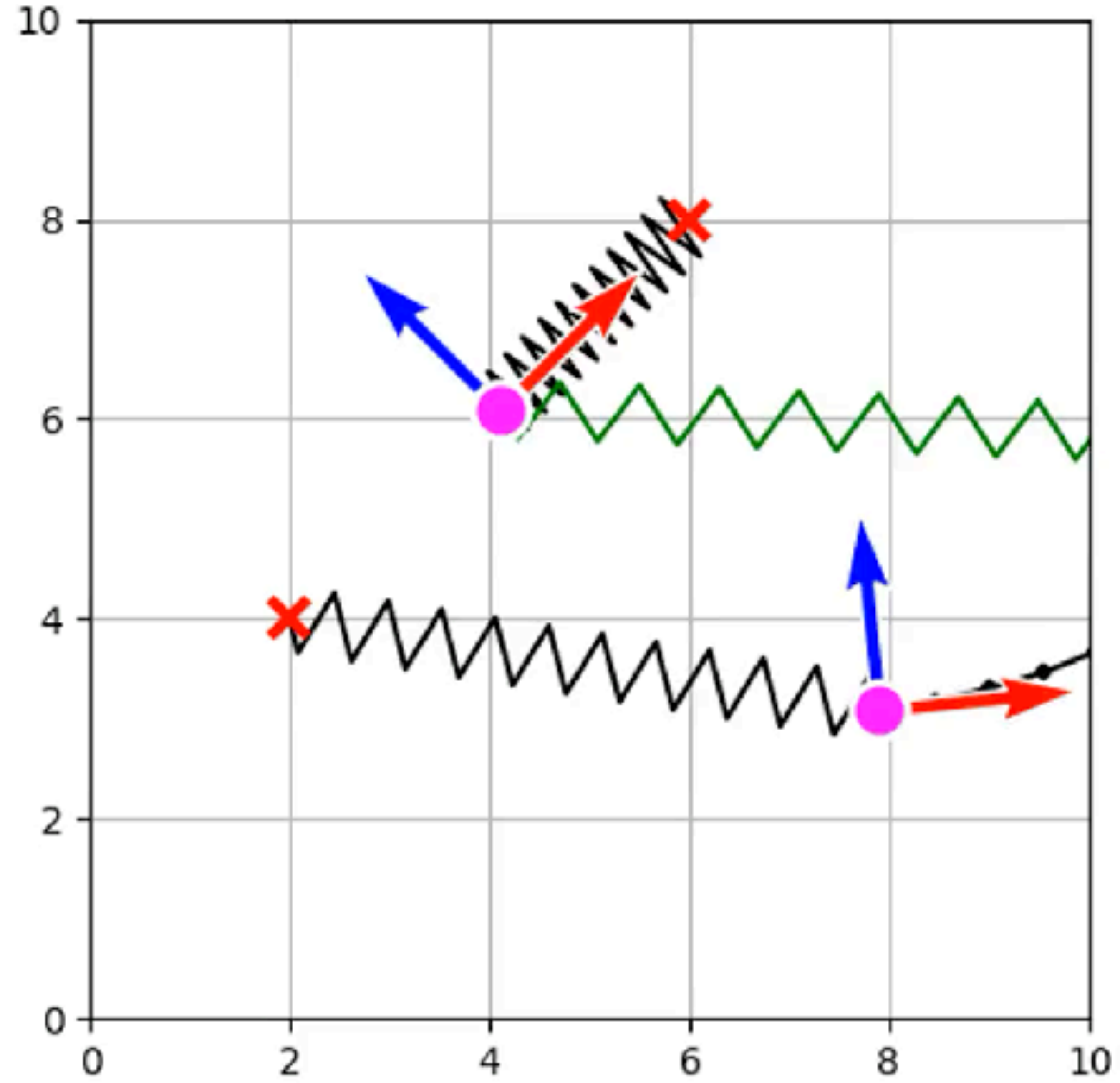
$\mathbf{u}_2 = [4\sqrt{2}, \pi/4]^\top$

Example: GPS + motion model



Half-turn, then move, then half-turn

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \Delta t \cos(\theta_{t-1} + \omega_t \Delta t/2) \\ y_{t-1} + v_t \Delta t \sin(\theta_{t-1} + \omega_t \Delta t/2) \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix} = g\left(\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}, \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \right)$$

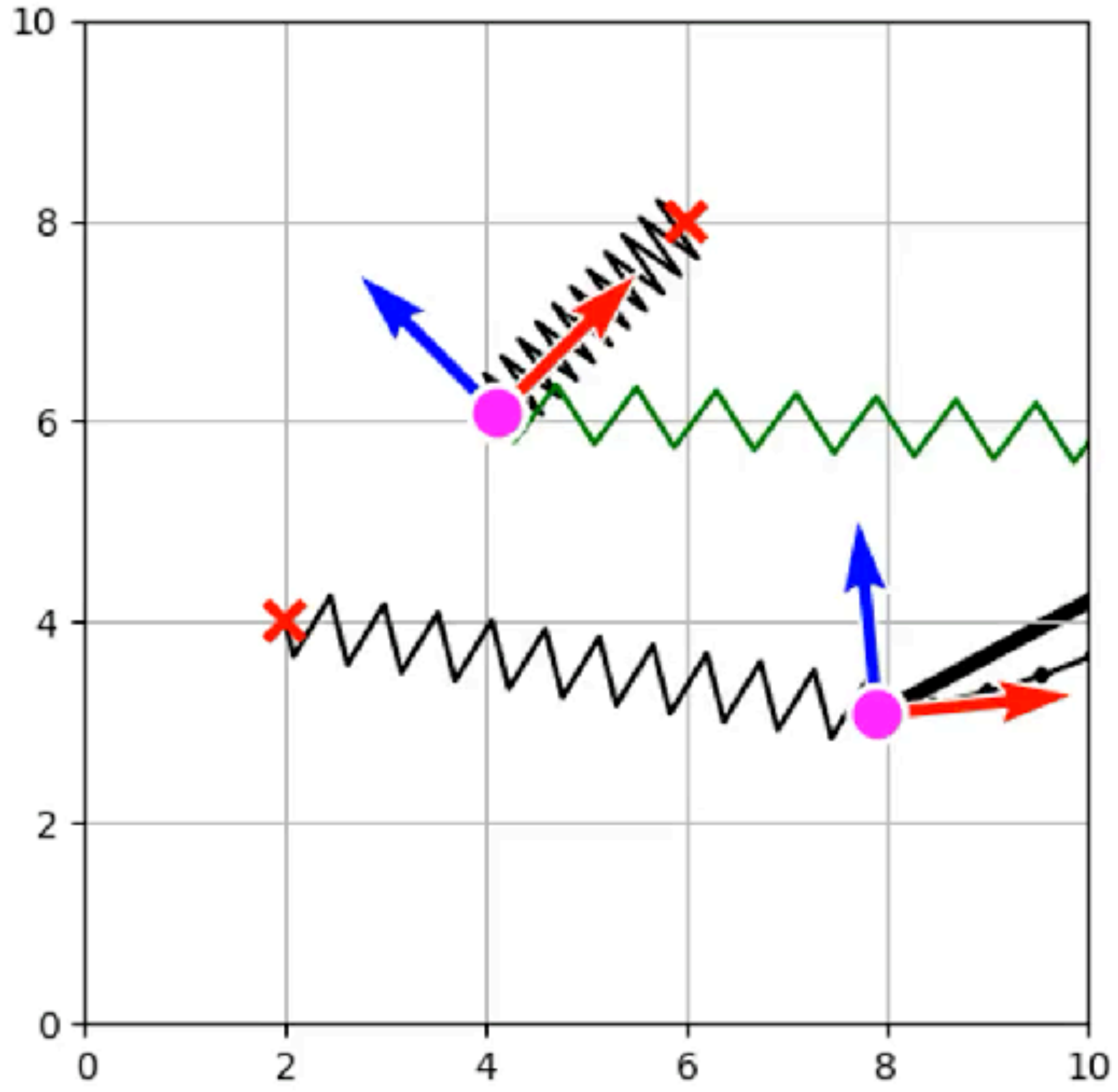


Analytical integration

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t/\omega_t(\sin(\theta_{t-1} + \omega_t \Delta t) - \sin\theta_{t-1}) \\ y_{t-1} + v_t/\omega_t(-\cos(\theta_{t-1} + \omega_t \Delta t) + \cos\theta_{t-1}) \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix}$$

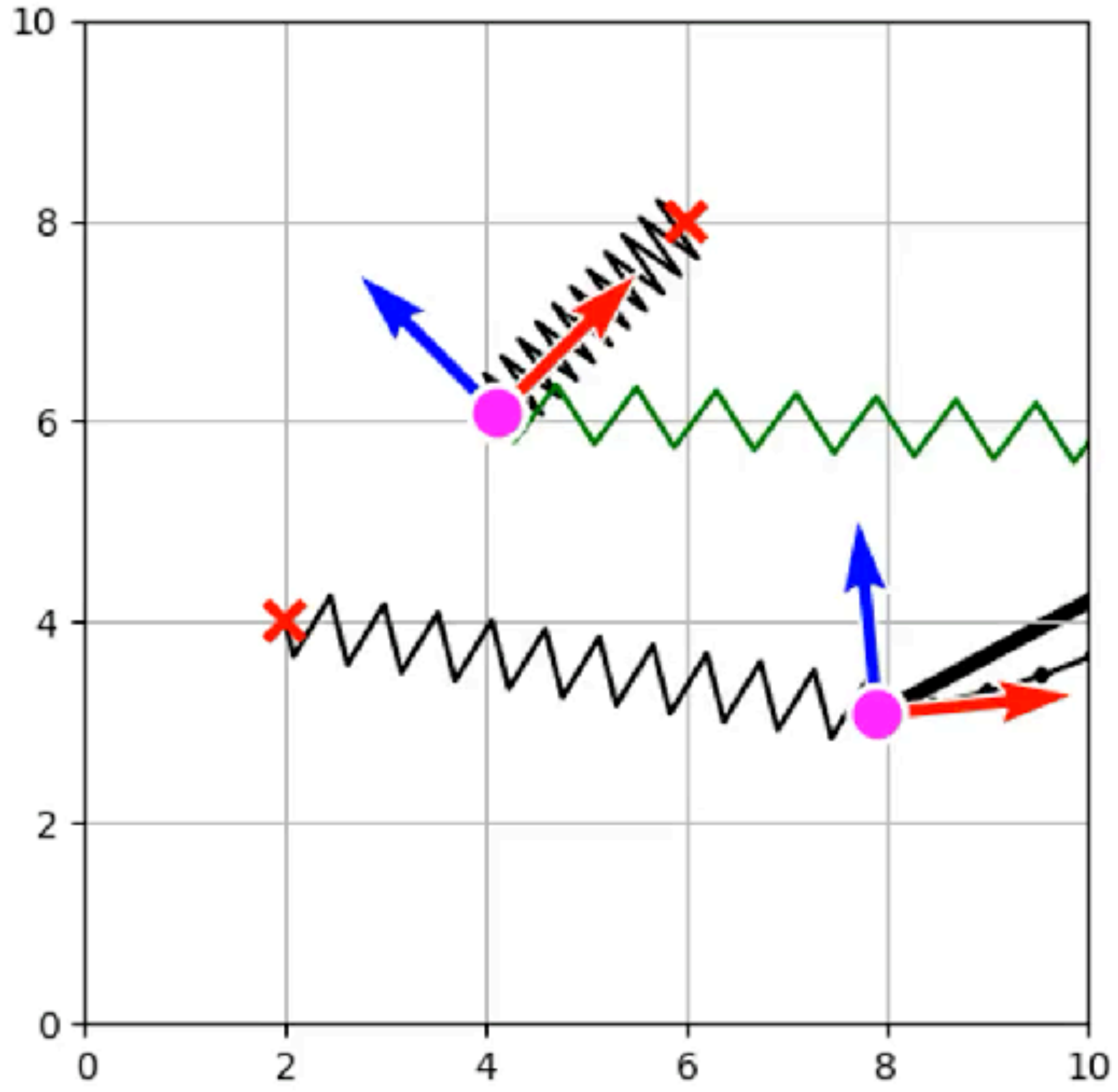
The difference is quite small for small angular velocity

$$\mathbf{u}_2 = [4\sqrt{2}, \pi/4]^\top$$

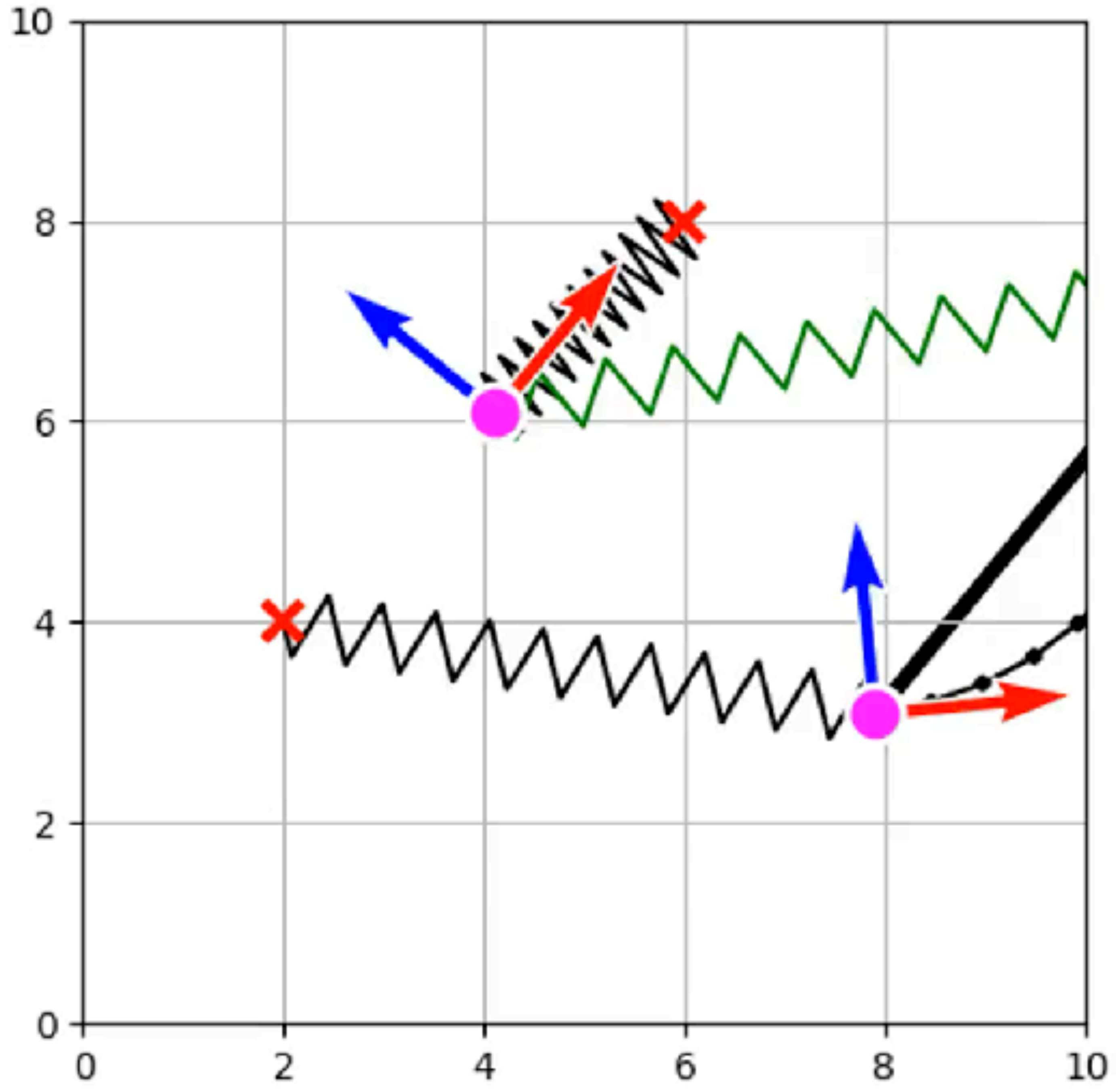


The difference is quite small for small angular velocity

$$\mathbf{u}_2 = [4\sqrt{2}, \pi/4]^\top$$

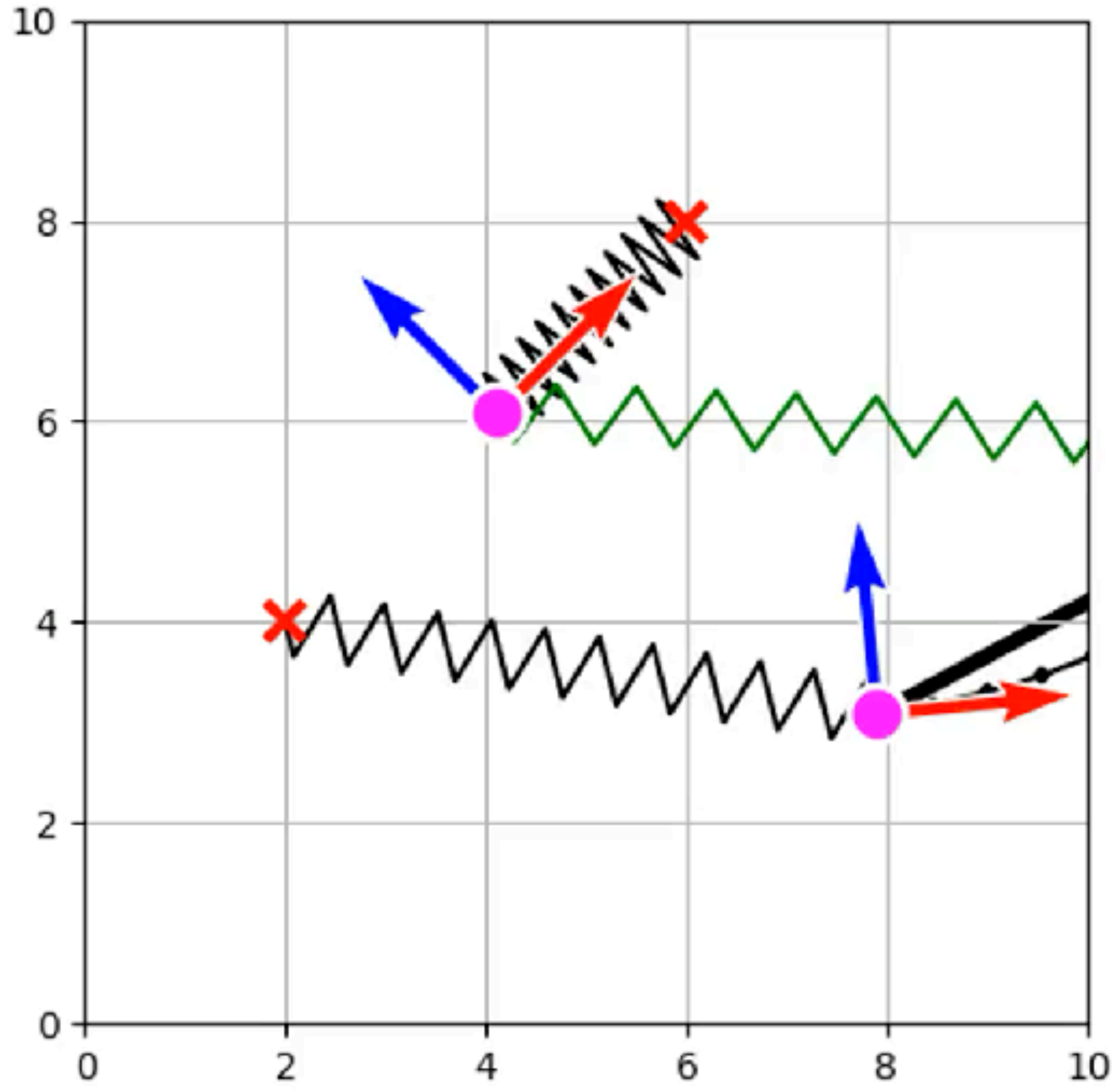


$$\mathbf{u}_2 = [4\sqrt{2}, \pi/2]^\top$$

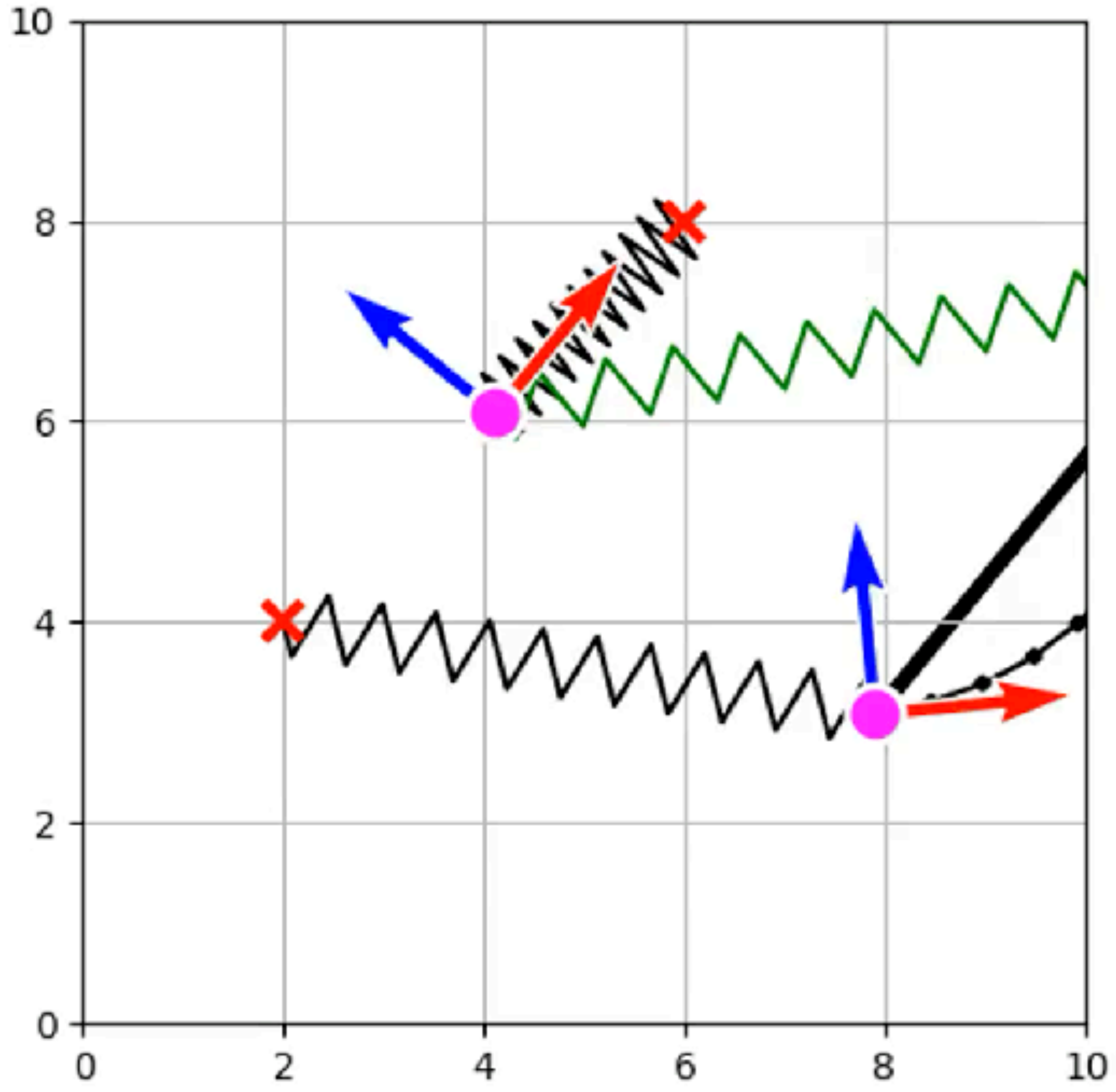


The difference is quite small for small angular velocity

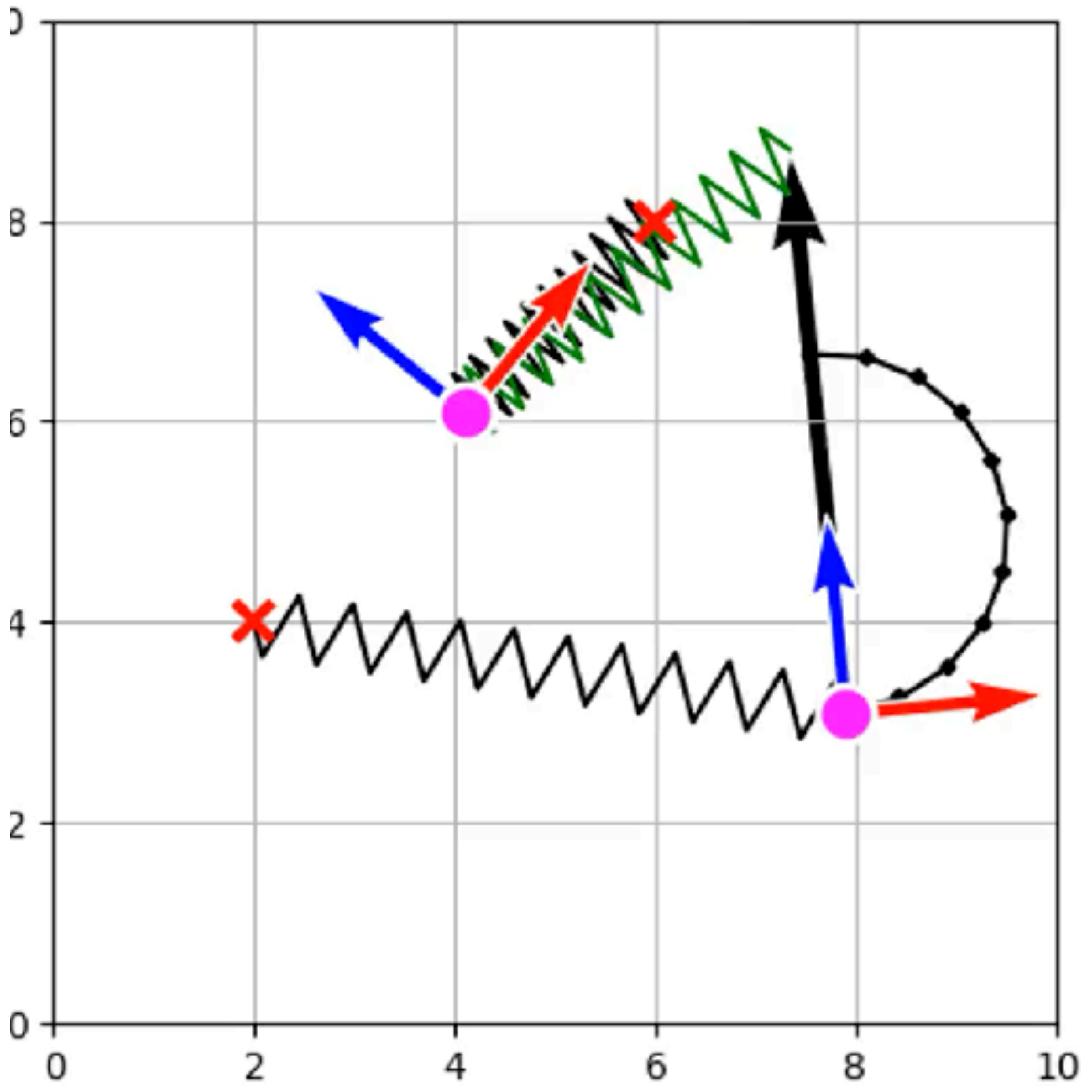
$\mathbf{u}_2 = [4\sqrt{2}, \pi/4]^\top$



$\mathbf{u}_2 = [4\sqrt{2}, \pi/2]^\top$



$\mathbf{u}_2 = [4\sqrt{2}, \pi]^\top$



Where do I get control command?

Standard control ROS topic `/cmd_vel`

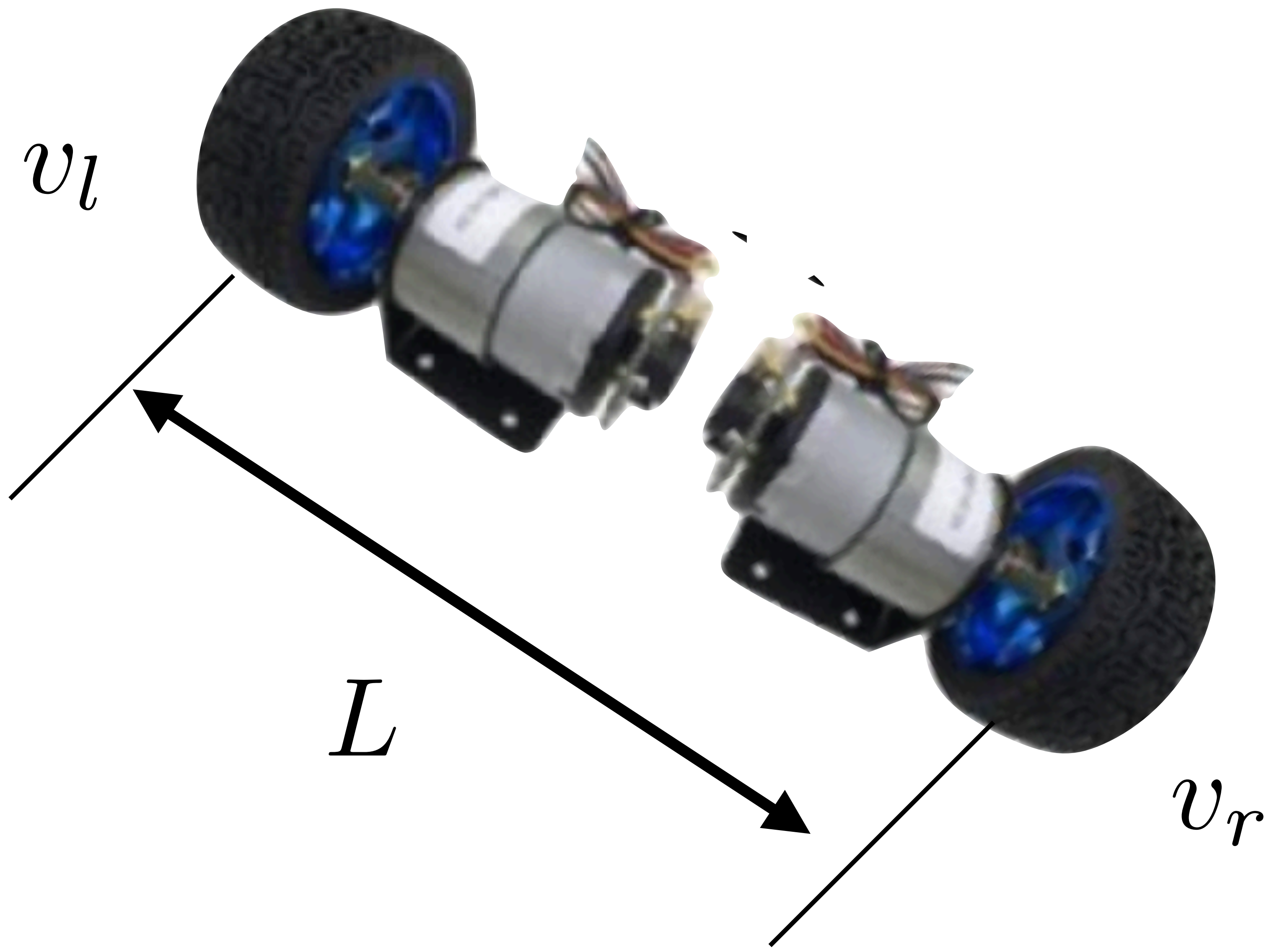
```
/cmd_vel= '{linear:  {x: 1.0, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 1.0}}'
```

Differential drive platforms:

$$\mathbf{u}_t = [\ \text{Linear velocity } v,\quad \text{Angular velocity } \omega\ ]$$

Control commands often replaced by wheel velocities

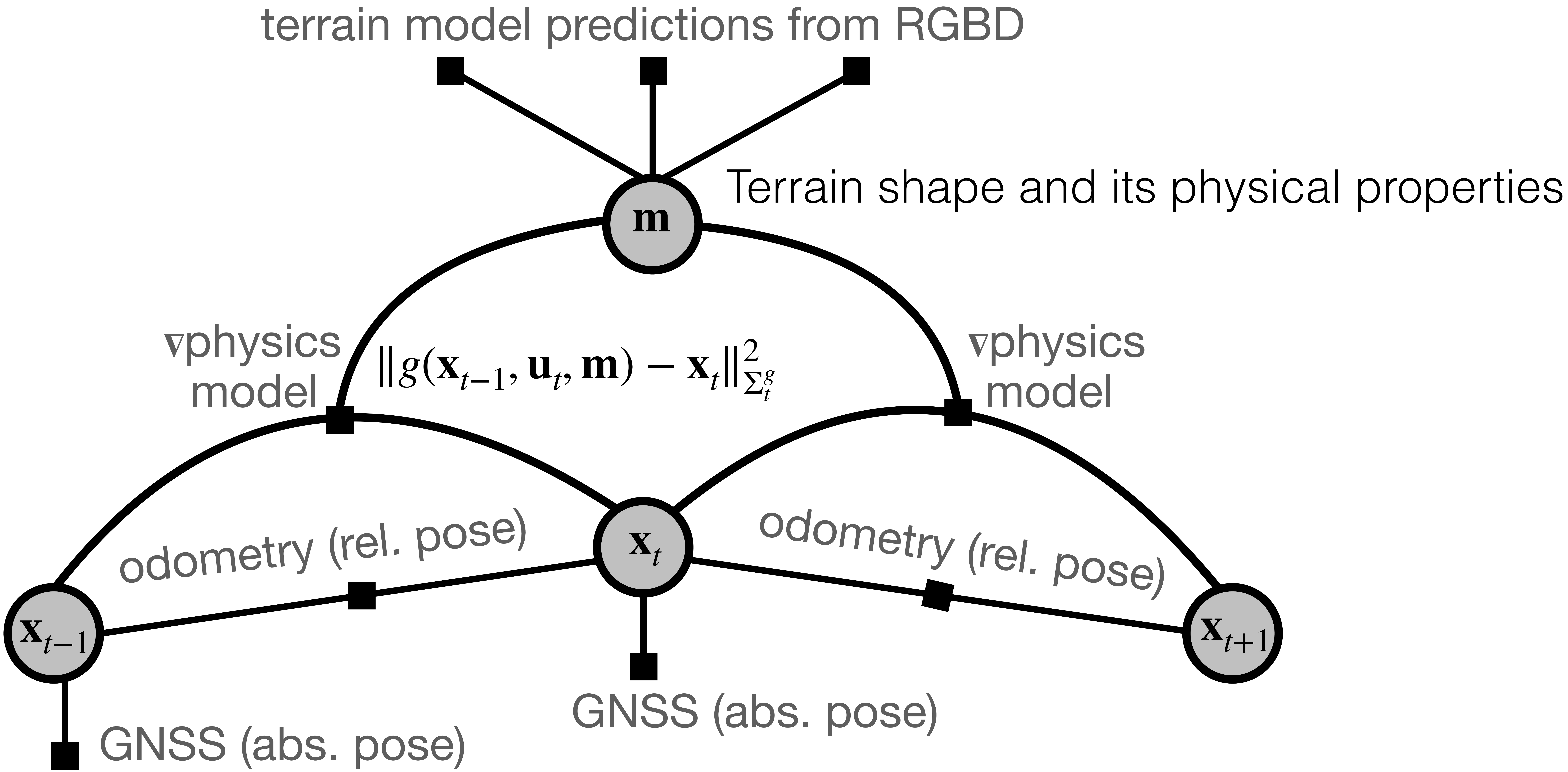
$$v = \frac{v_l + v_r}{2} \qquad \omega = \frac{v_l - v_r}{L}$$



What if I bounce into obstacle?

Are there any more sophisticated models that take the terrain in account?

SLAM - our approach

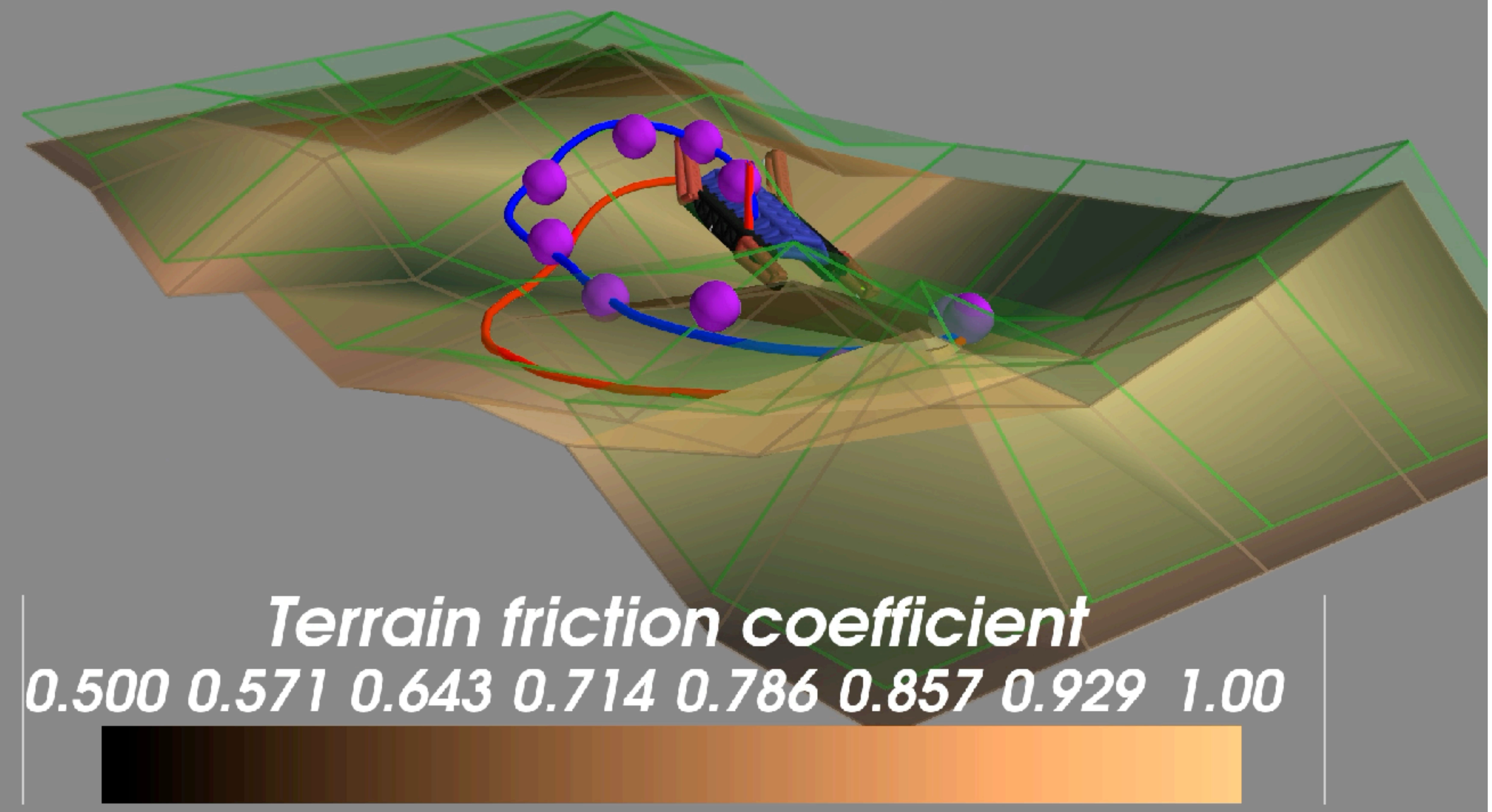


Input: noisy gps + imu + control

Output: trajectory + map (field of robot-terrain inter. forces)

loss = 26.205

GraphSLAM requires differentiable ODE solver



Google's BRAX - differentiable physics engine

https://github.com/google/brax

Brax simulates these environments at millions of physics steps per second on TPU

NVIDIA WARP - differentiable physics engine

https://developer.nvidia.com/warp-python

Cloth simulation

Particle-based simulation

Straightforward extensions

GPS, odometry, 3D marker(s)

$$= \arg \min_{\substack{\mathbf{x}_0,\ldots\mathbf{x}_T \\ \mathbf{m}^1\ldots\mathbf{m}^J}} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}^v\|^2_{\Sigma_t^v} + \sum_{t,j} \|\mathrm{w2r}(\mathbf{m}^j, \mathbf{x}_t) - \mathbf{z}\|^2_{\Sigma_t^{m^j}}$$

priors, loop-closures

$$+ \sum_t \|\mathbf{x}_t - \mathbf{x}_t^{prior}\|^2_{\Sigma_t^{prior}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_0, \mathbf{x}_T)\|^2_{\Sigma_t^{lc}}$$

motion model, UWB, 2D marker(s)

$$+ \sum_t \|g(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t^g} + ??? + ???$$

e.g. camera detections

Straightforward extensions

GPS | odometry | 3D marker(s)

$$= \arg\min_{\substack{\mathbf{x}_0,\ldots\mathbf{x}_T \\ \mathbf{m}^1\ldots\mathbf{m}^J}} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}^v\|^2_{\Sigma_t^v} + \sum_{t,j} \|\mathrm{w2r}(\mathbf{m}^j, \mathbf{x}_t) - \mathbf{z}\|^2_{\Sigma_t^{m^j}}$$

priors | loop-closures

$$+ \sum_t \|\mathbf{x}_t - \mathbf{x}_t^{prior}\|^2_{\Sigma_t^{prior}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_0, \mathbf{x}_T)\|^2_{\Sigma_t^{lc}}$$

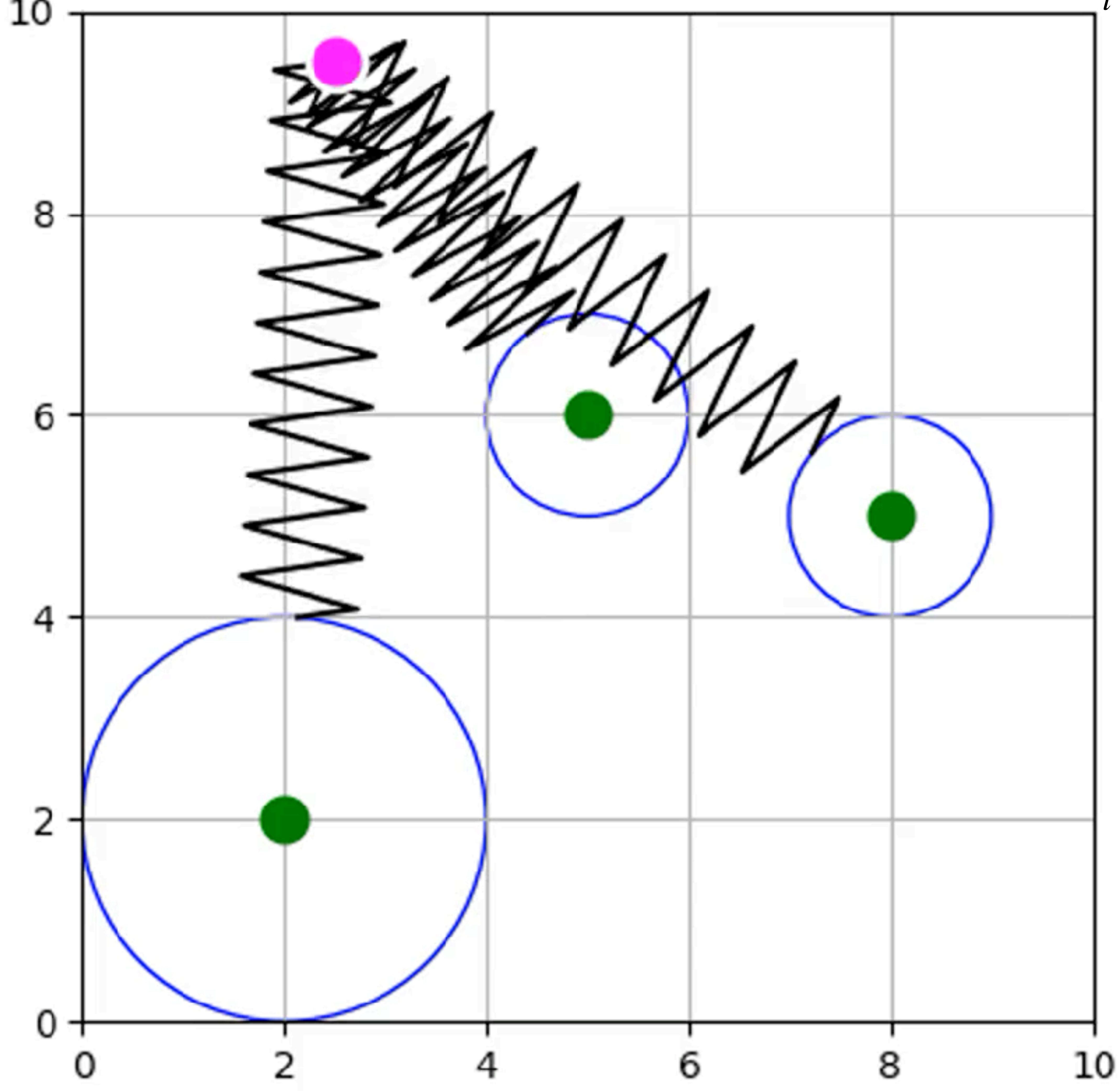
motion model | UWB | 2D marker(s)

$$+ \sum_t \|g(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t^g} + ??? + ???$$

e.g. camera detections

UWB

$$\mathbf{x}^{\star} = \arg\min_{\mathbf{x}_t} \sum_i \| \|\mathbf{x}_t - \mathbf{m}_i\| - \mathbf{z}_t^{UWB_i} \|^2_{\Sigma_t^{UWB_i}}$$



$\mathbf{x}_t$ … robot poses

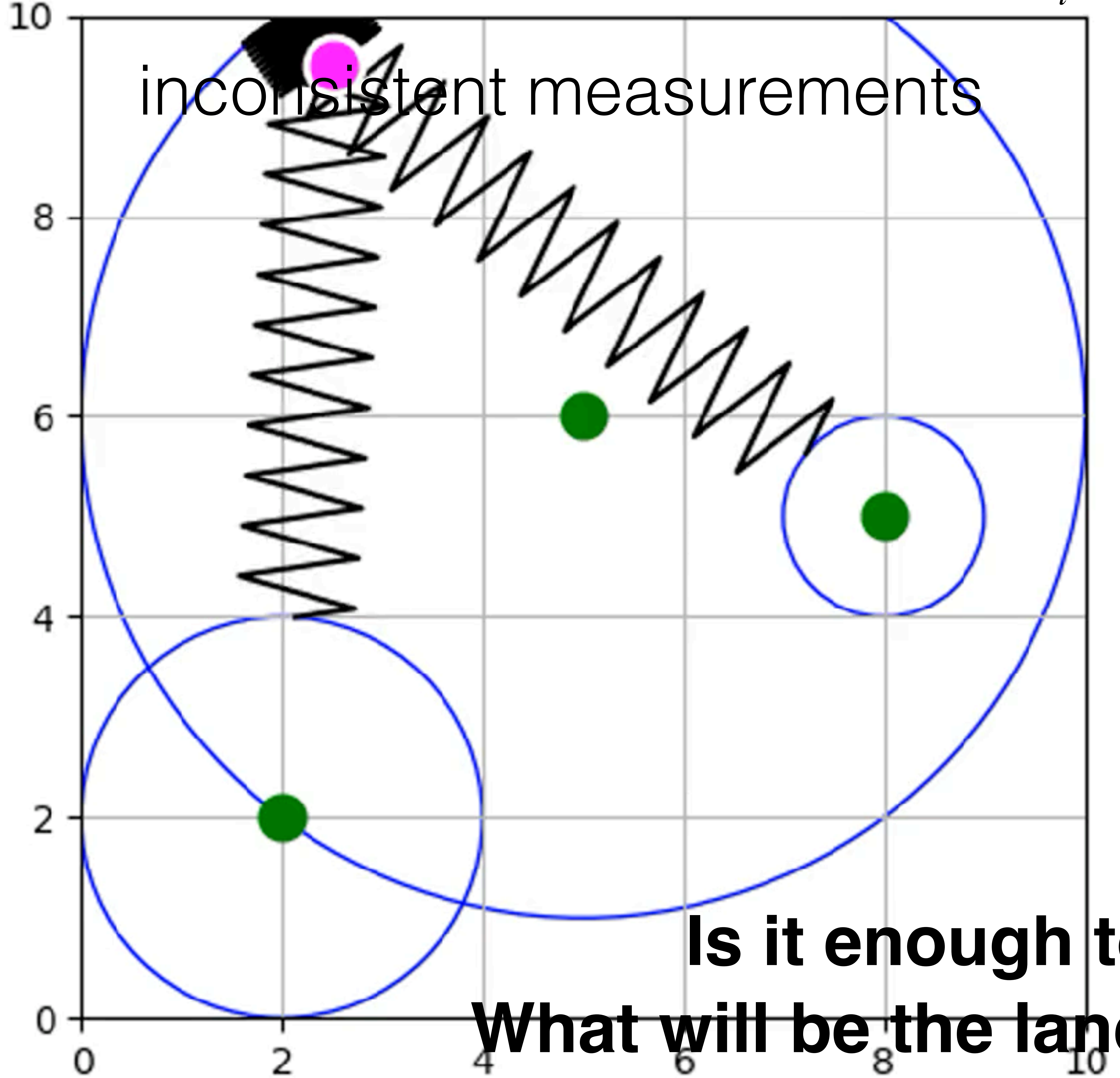
$\mathbf{m}_i$ … known marker positions

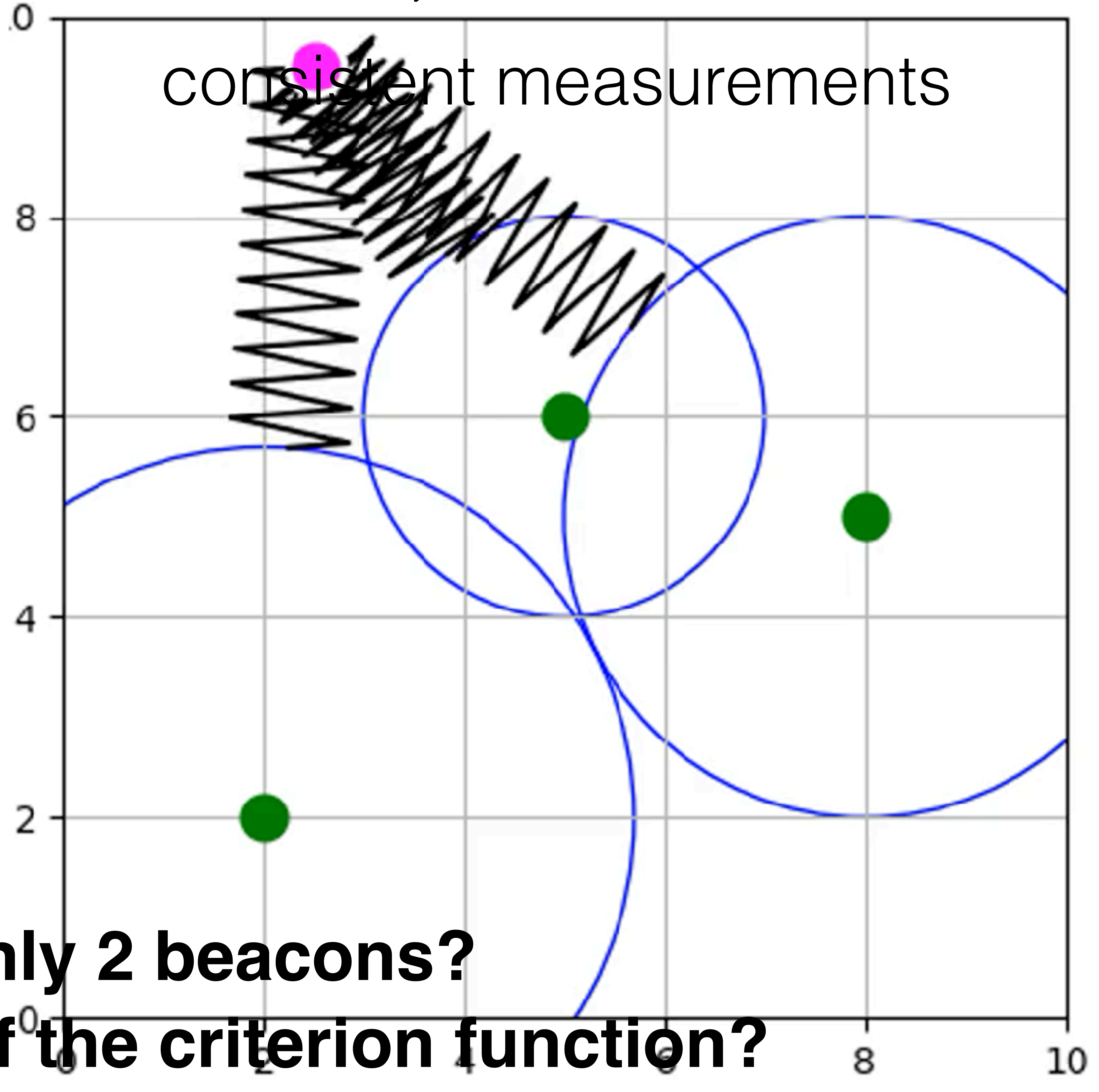
$\mathbf{z}_t^{UWB}$ … UWB measurements (distance)

… UWB loss

UWB

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}_t} \sum_i \| \|\mathbf{x}_t - \mathbf{m}_i\| - \mathbf{z}_t^{UWB_i} \|^2_{\Sigma_t^{UWB_i}}$$

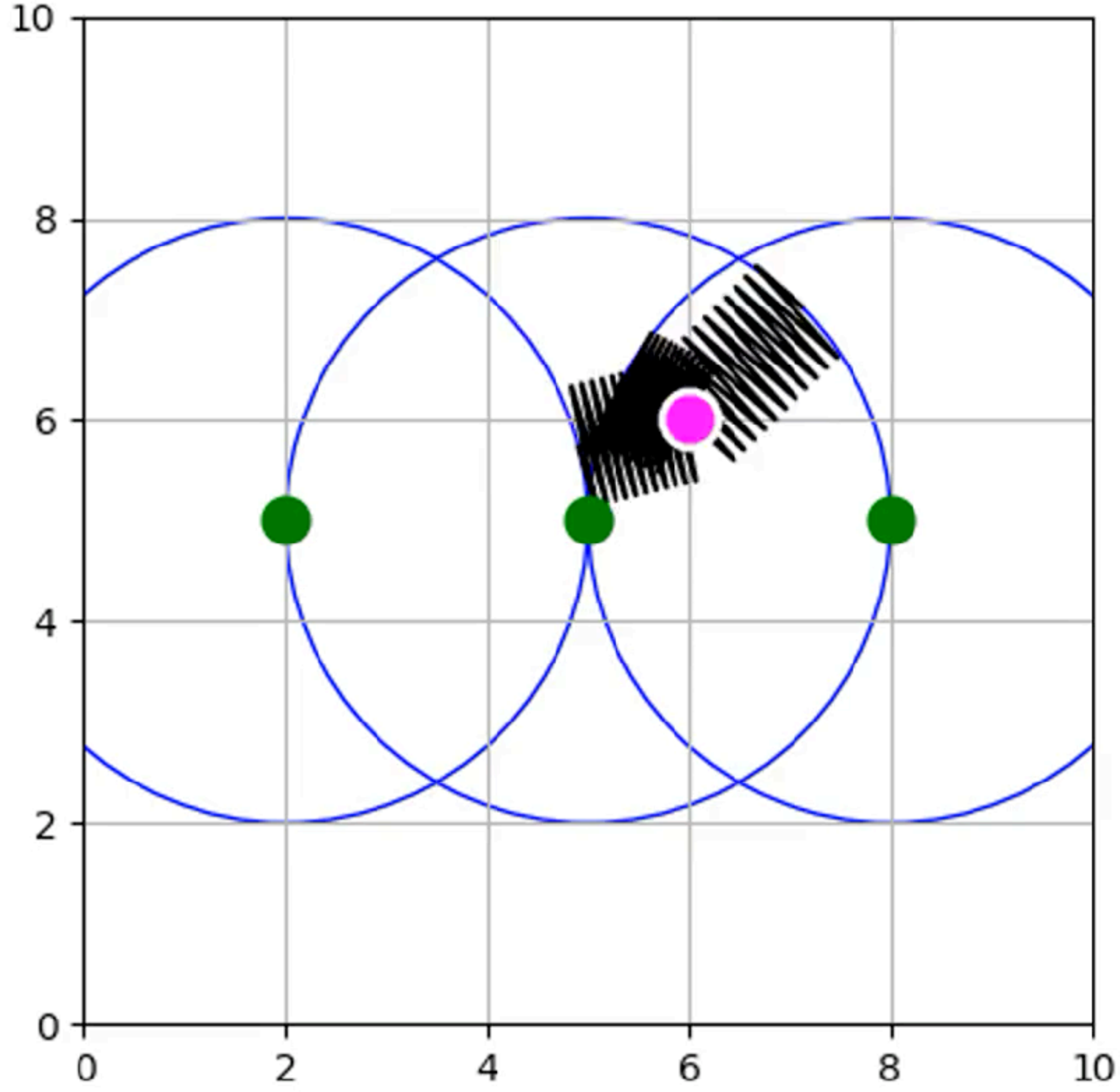




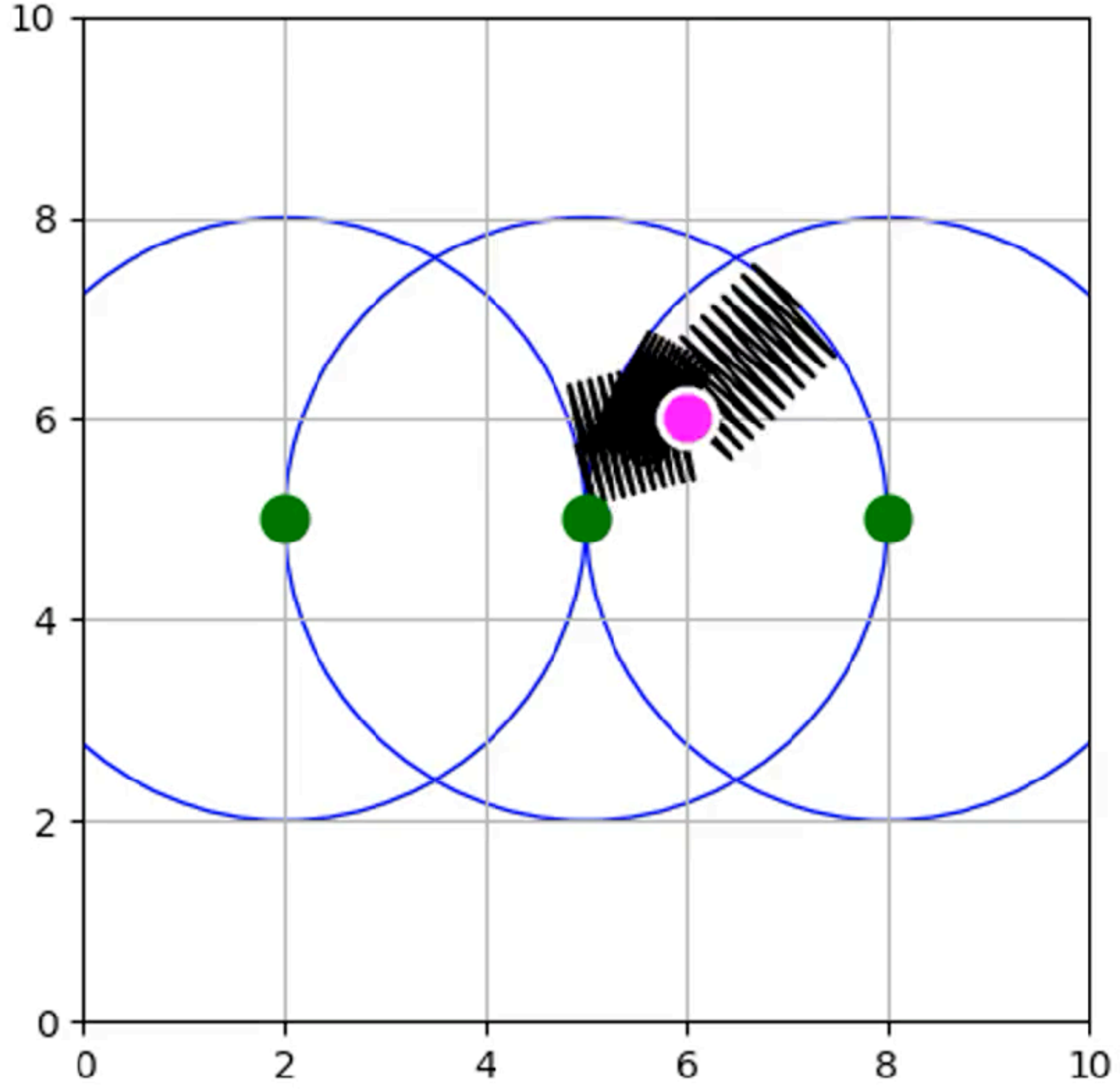
Is it enough to have only 2 beacons?

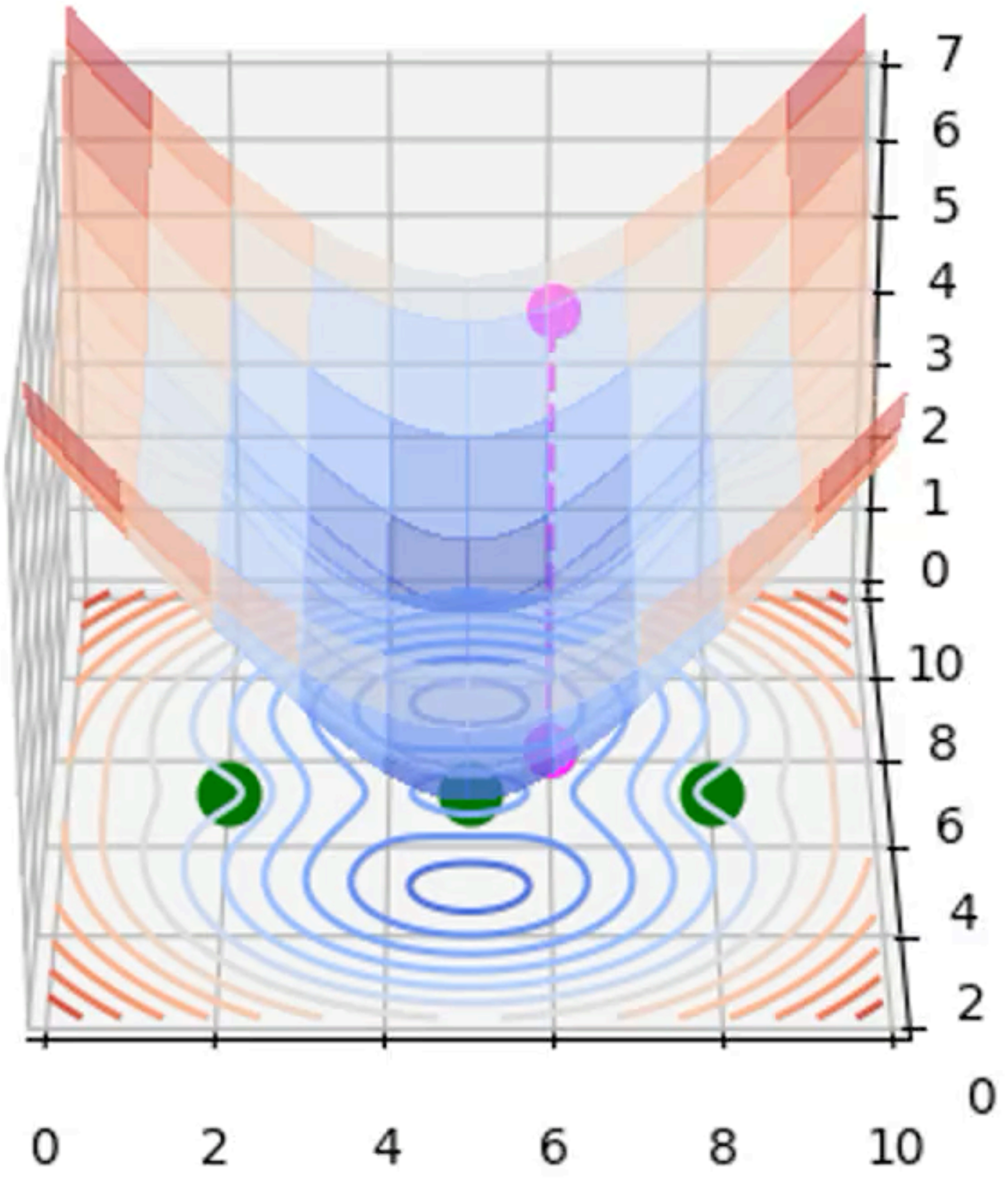
What will be the landscape of the criterion function?

What will be the landscape of the criterion function?

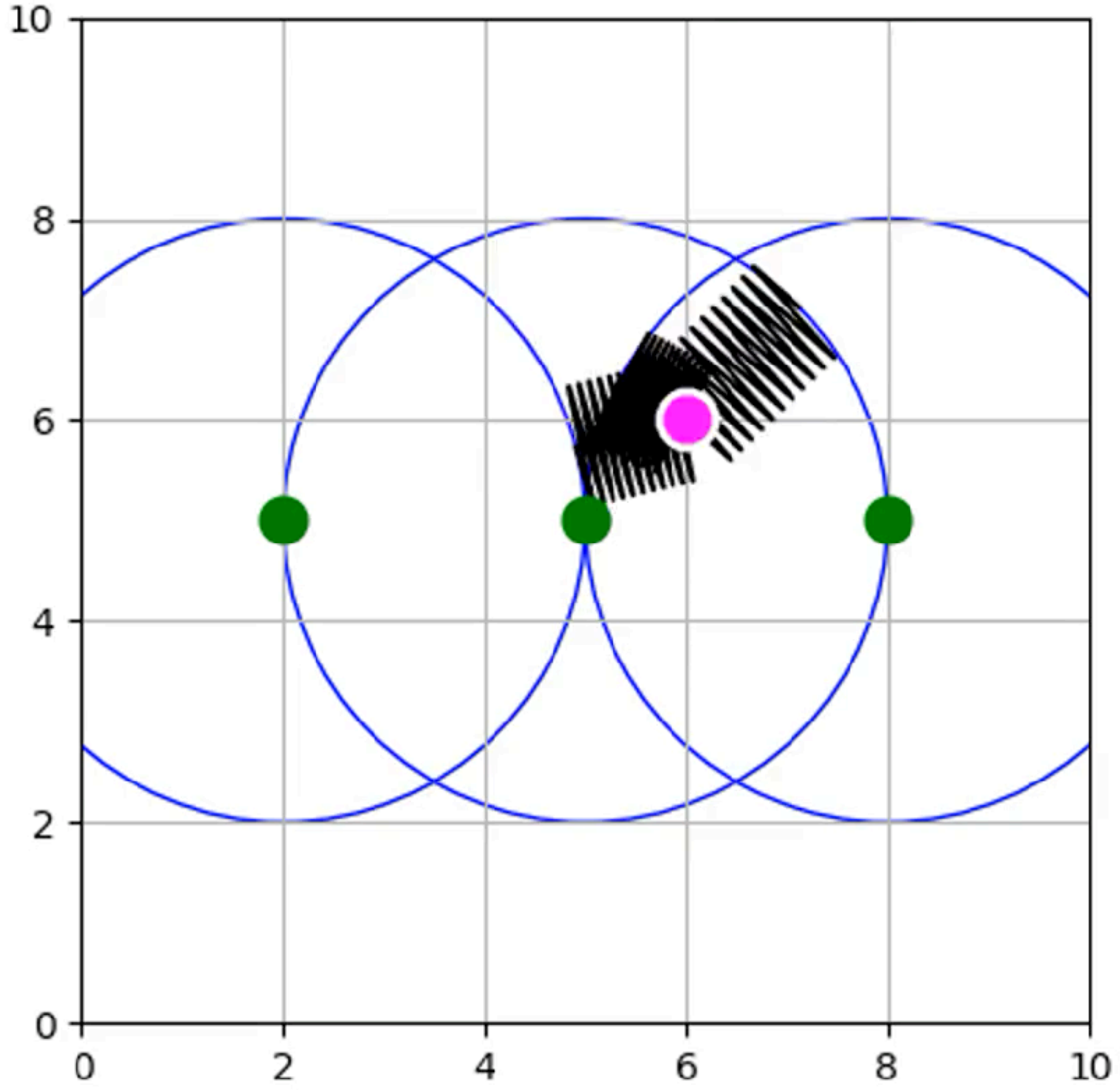


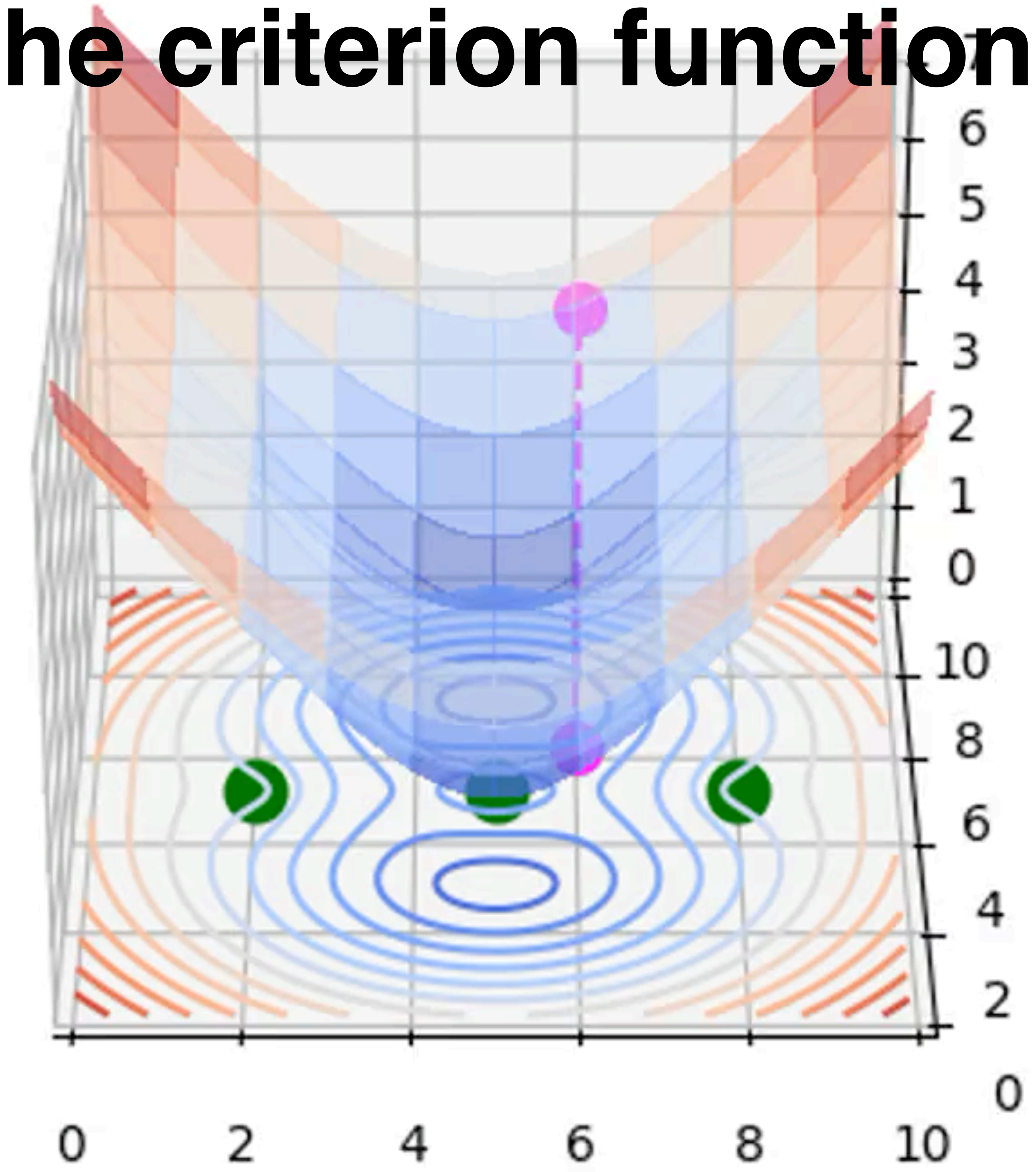
What will be the landscape of the criterion function?

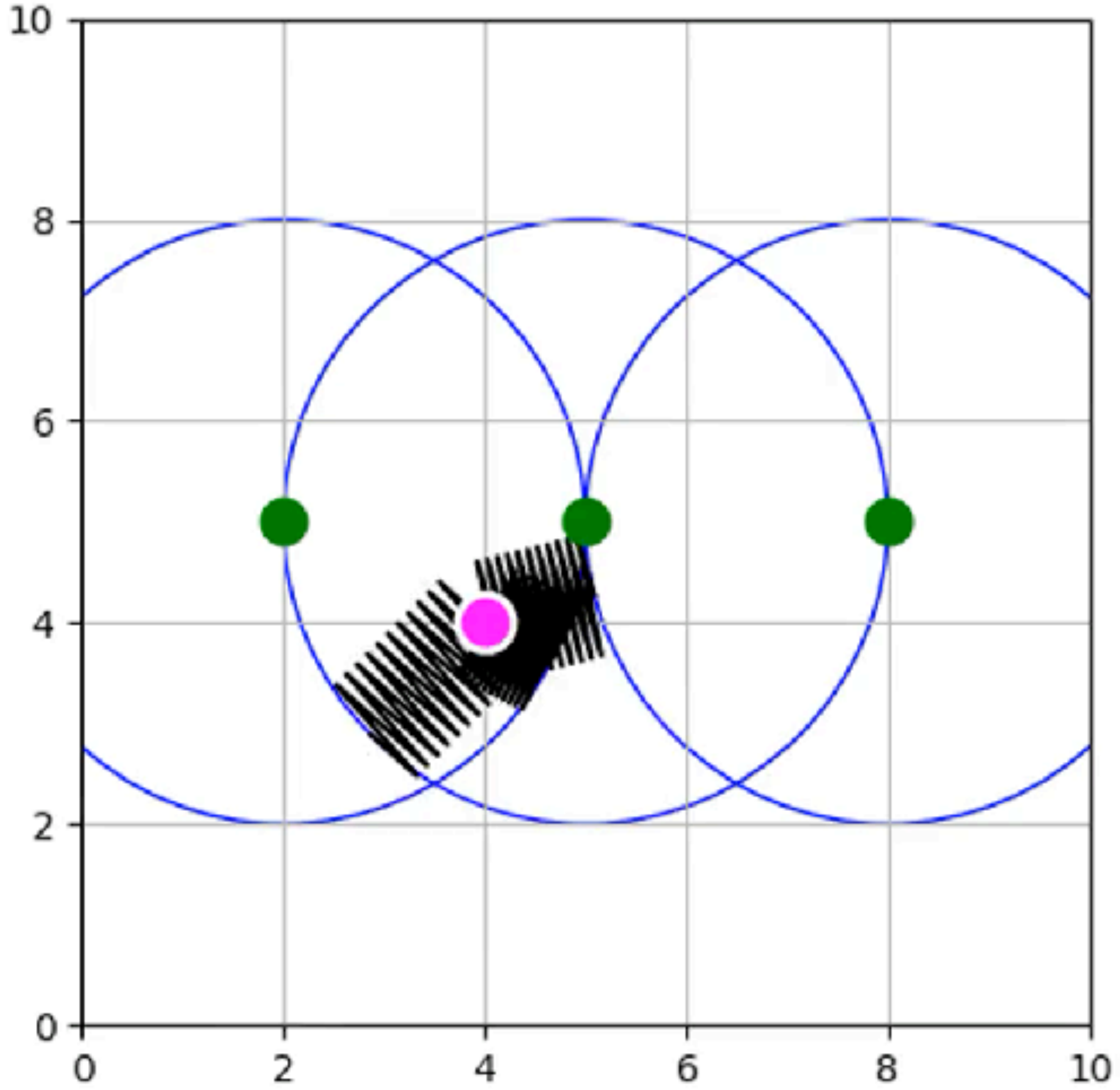


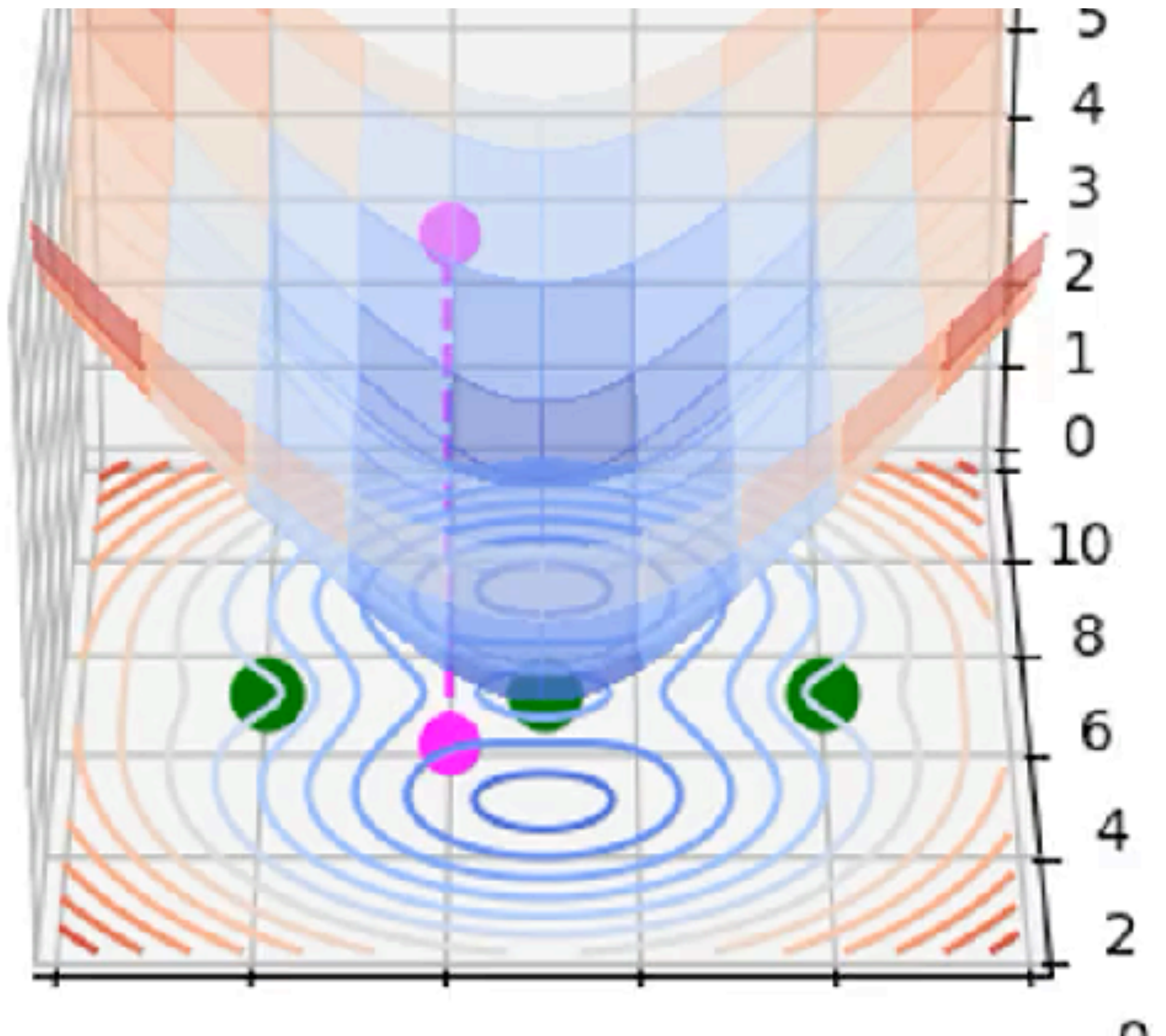


What will be the landscape of the criterion function?

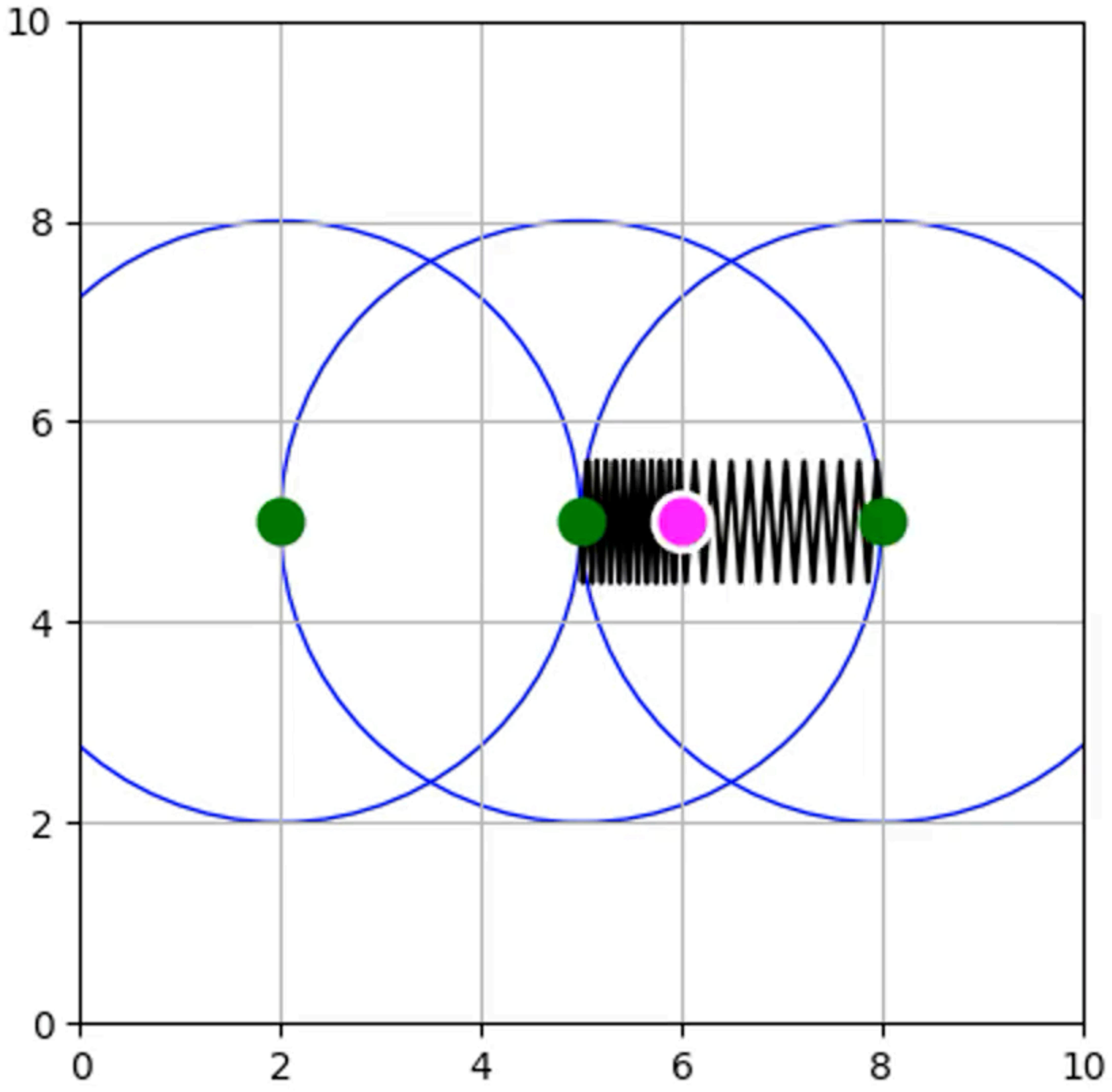


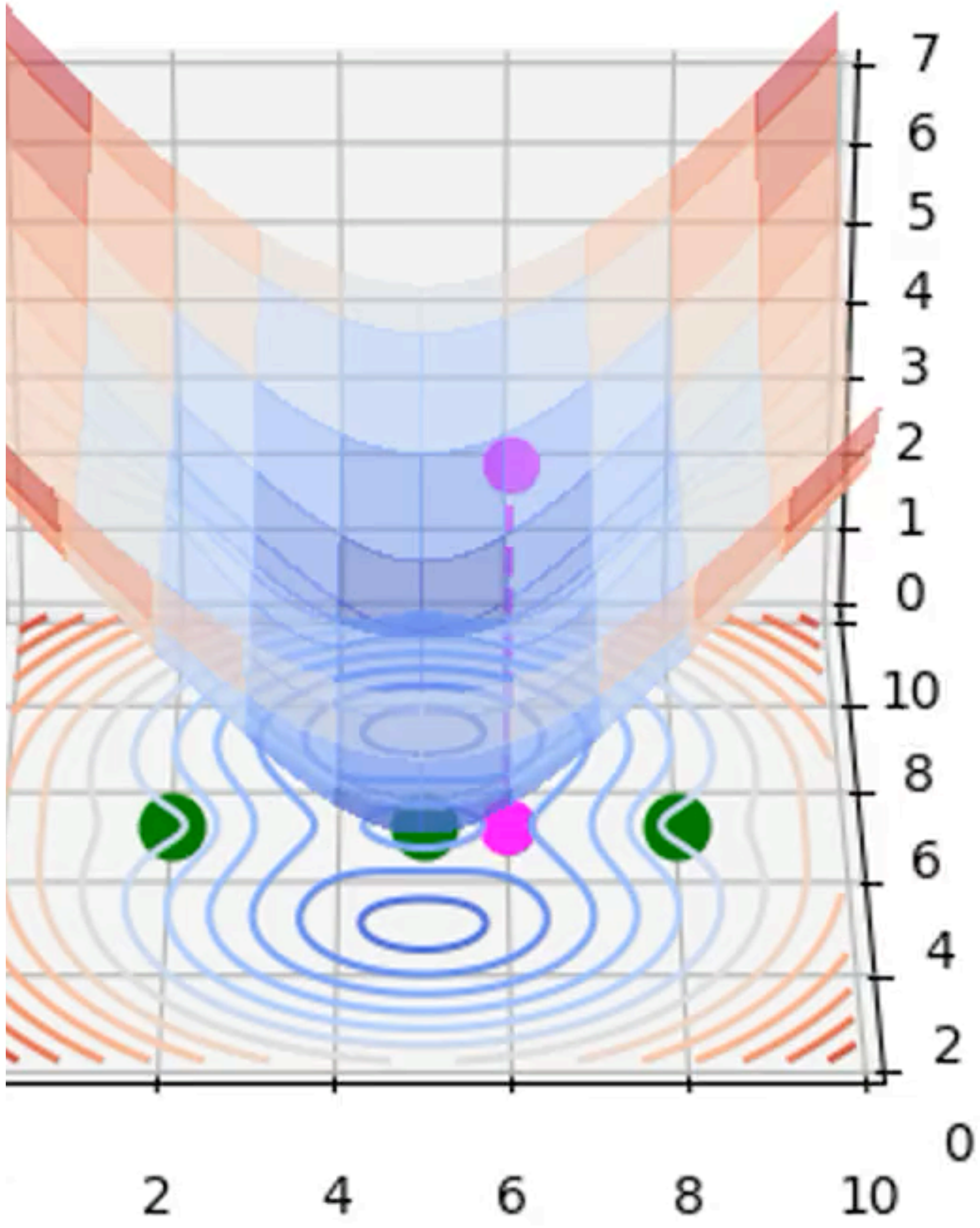






Saddle points





UWB

$$\mathbf{x}^{\star} = \arg\min_{\mathbf{x}_t, \mathbf{m}_i} \sum_{i,t} \| \|\mathbf{x}_t - \mathbf{m}_i\| - \mathbf{z}_t^{UWB_i} \|^2_{\Sigma_t^{UWB_i}}$$

UWB SLAM

What is the dimensionality of zero-loss subspace?

"8-dim space" - "3 DOF from measurements" = "5-dim"

Straightforward extensions

GPS — odometry — 3D marker(s)

$$= \arg\min_{\substack{\mathbf{x}_0,\ldots\mathbf{x}_T \\ \mathbf{m}^1\ldots\mathbf{m}^J}} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}^v\|^2_{\Sigma_t^v} + \sum_{t,j} \|\mathrm{w2r}(\mathbf{m}^j, \mathbf{x}_t) - \mathbf{z}\|^2_{\Sigma_t^{m^j}}$$

priors — loop-closures

$$+ \sum_t \|\mathbf{x}_t - \mathbf{x}_t^{prior}\|^2_{\Sigma_t^{prior}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_0, \mathbf{x}_T)\|^2_{\Sigma_t^{lc}}$$

motion model — UWB

$$+ \sum_t \|g(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t^g} + \sum_{i,t} \| \|\mathbf{x}_t - \mathbf{m}_i\| - \mathbf{z}_t^{UWB_i}\|^2_{\Sigma_t^{UWB_i}}$$

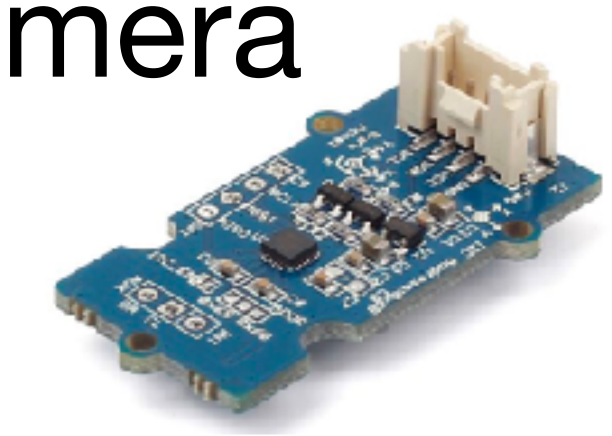
+ 2D marker(s)

???

e.g. camera detections

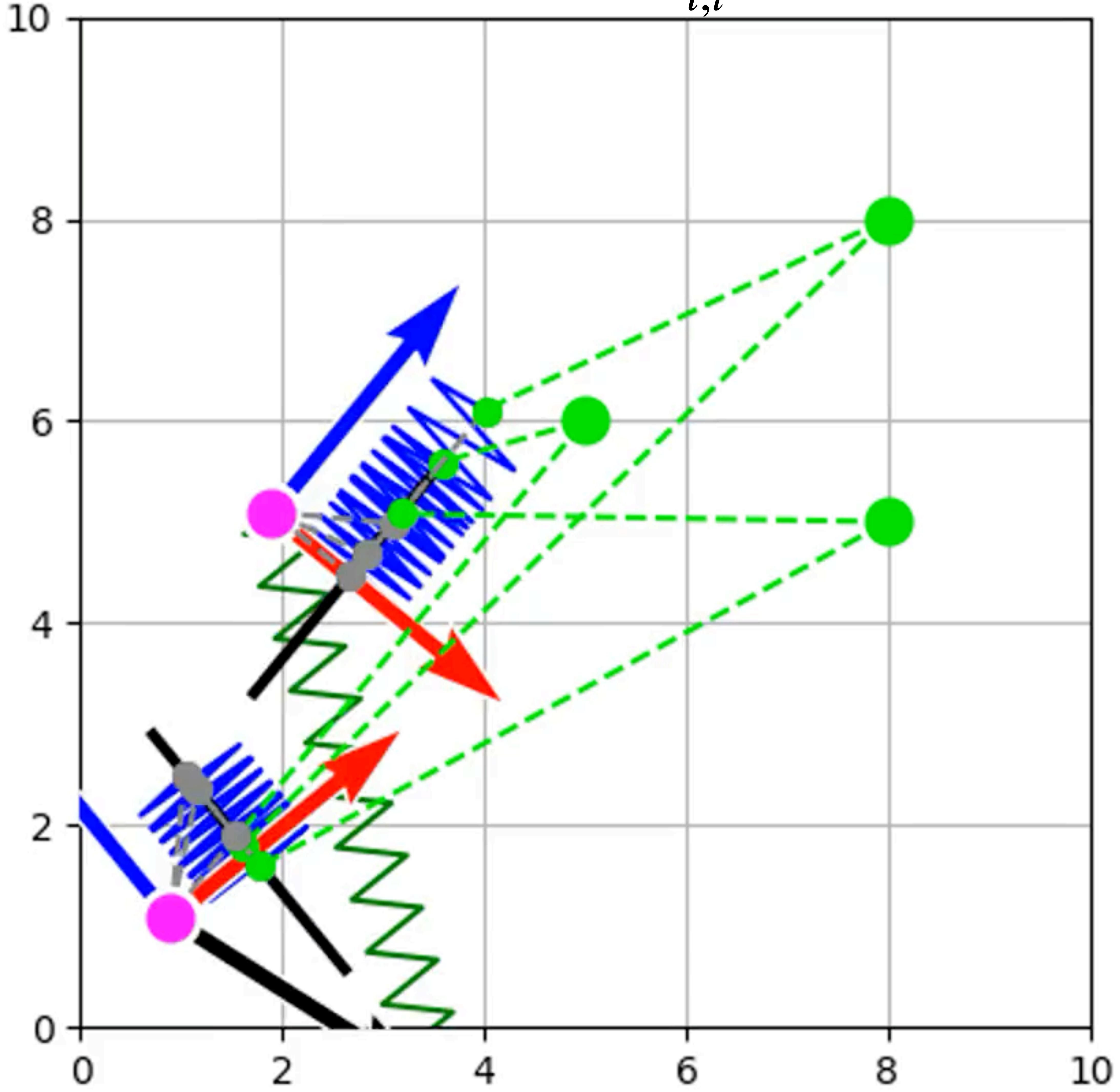
Localization from camera

2D marker detector (RGB camera) +



Odometry (IMU)

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}_t} \sum_{i,t} \|\text{w2cam}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m}_i}\|^2 + \|\text{w2r}(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{z}_{12}^{odom}\|^2$$



$\mathbf{x}_t$ …robot poses

$\mathbf{m}_i$ …known marker positions

$\mathbf{z}_t^{\mathbf{m}_i}$ …marker measurements

$\sum_{i,t} \|\text{w2r}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m}_i}\|^2$ … marker loss

$\|\text{w2r}(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{z}_{12}^{odom}\|^2$ … odom loss

camera coordinate frame + img. plane

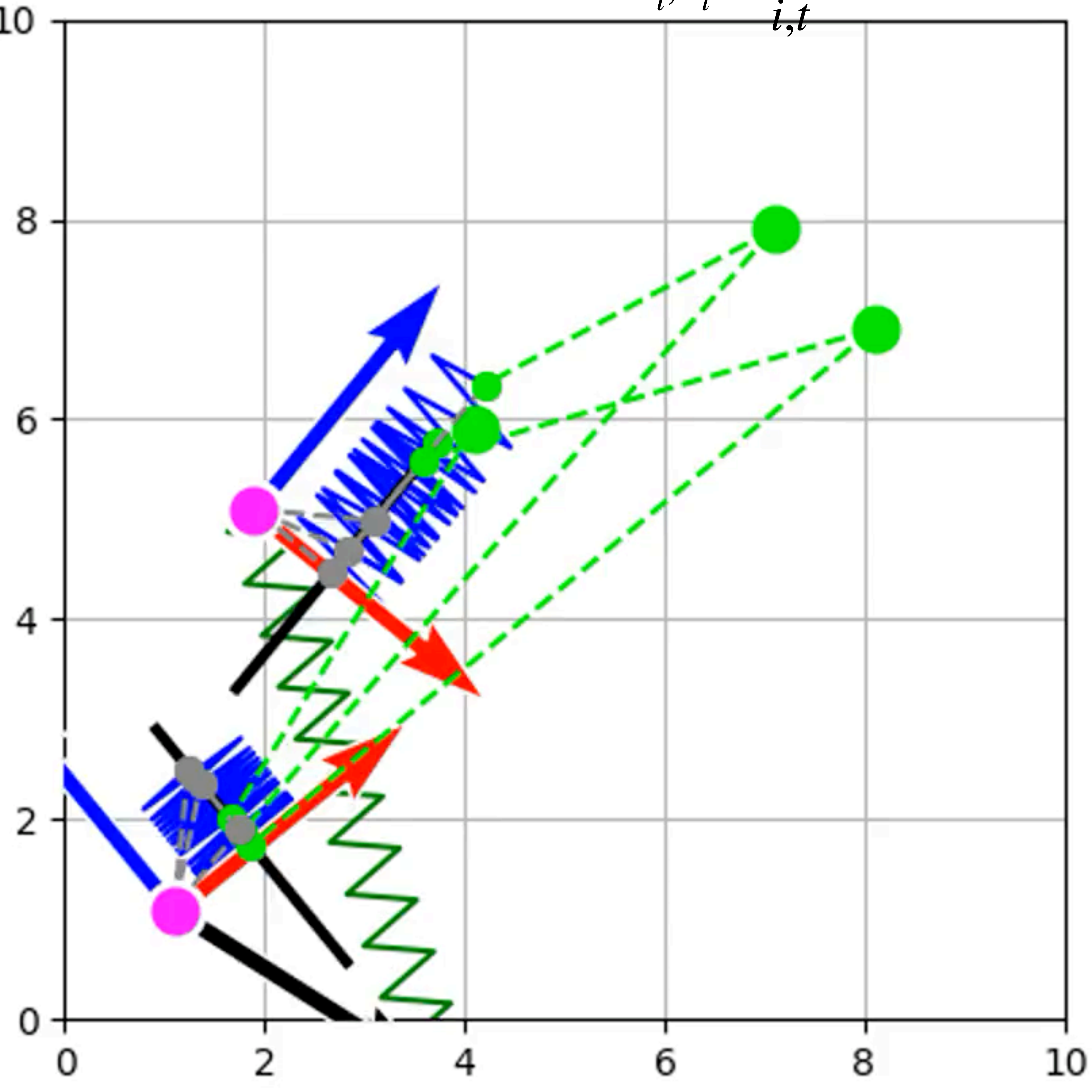
odometry

SLAM from camera (bundle adjustment)

2D marker detector (RGB camera) +

Odometry (IMU)

$$\mathbf{x}^\star = \arg\min_{\mathbf{m}_i, \mathbf{x}_t} \sum_{i,t} \|\text{w2cam}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m}_i}\|^2 + \|\text{w2r}(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{z}_{12}^{odom}\|^2$$



$\mathbf{x}_t$ ...robot poses

$\mathbf{m}_i$ ...known marker positions

$\mathbf{z}_t^{\mathbf{m}_i}$ ...marker measurements

$\sum_{i,t} \|\text{w2r}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m}_i}\|^2$ ... marker loss

$\|\text{w2r}(\mathbf{x}_2, \mathbf{x}_1) - \mathbf{z}_{12}^{odom}\|^2$ ... odom loss

camera coordinate frame + img. plane

odometry

Straightforward extensions

GPS — odometry — 3D marker(s)

$$= \arg\min_{\substack{\mathbf{x}_0,\ldots\mathbf{x}_T \\ \mathbf{m}^1\ldots\mathbf{m}^J}} \sum_t \|\mathbf{x}_t - \mathbf{z}_t^{gps}\|^2_{\Sigma_t^{gps}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}^v\|^2_{\Sigma_t^v} + \sum_{t,j} \|\mathrm{w2r}(\mathbf{m}^j, \mathbf{x}_t) - \mathbf{z}\|^2_{\Sigma_t^{m^j}}$$

priors — loop-closures

$$+ \sum_t \|\mathbf{x}_t - \mathbf{x}_t^{prior}\|^2_{\Sigma_t^{prior}} + \sum_t \|\mathrm{w2r}(\mathbf{x}_0, \mathbf{x}_T)\|^2_{\Sigma_t^{lc}}$$

motion model — UWB

$$+ \sum_t \|g(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t^g} + \sum_{i,t} \|\|\mathbf{x}_t - \mathbf{m}_i\| - \mathbf{z}_t^{UWB_i}\|^2_{\Sigma_t^{UWB_i}}$$

2D marker(s)

$$+ \sum_{i,t} \|\mathrm{w2cam}(\mathbf{m}_i, \mathbf{x}_t) - \mathbf{z}_t^{\mathbf{m_i}}\|^2$$

e.g. camera detections

Problems for students

Show that: $\mathbf{z}^w = \begin{bmatrix} R(\theta_t) & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{z}^r + \mathbf{x}_t$ => $\mathbf{z}^r = \begin{bmatrix} R(\theta_t)^\top & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} (\mathbf{z}^w - \mathbf{x}_t)$

When does it matter, in which coordinate frame (rcf/wcf) the residual is measured?

Does the (non)-linear measurement and/or motion function always imply (non)-convex criterium?

Given problem with measurement and motion function with gaussian noise:

- Derive maximum likelihood estimate
- Draw corresponding factorgraph
- Write down criterion function
- Discuss its non-convexity