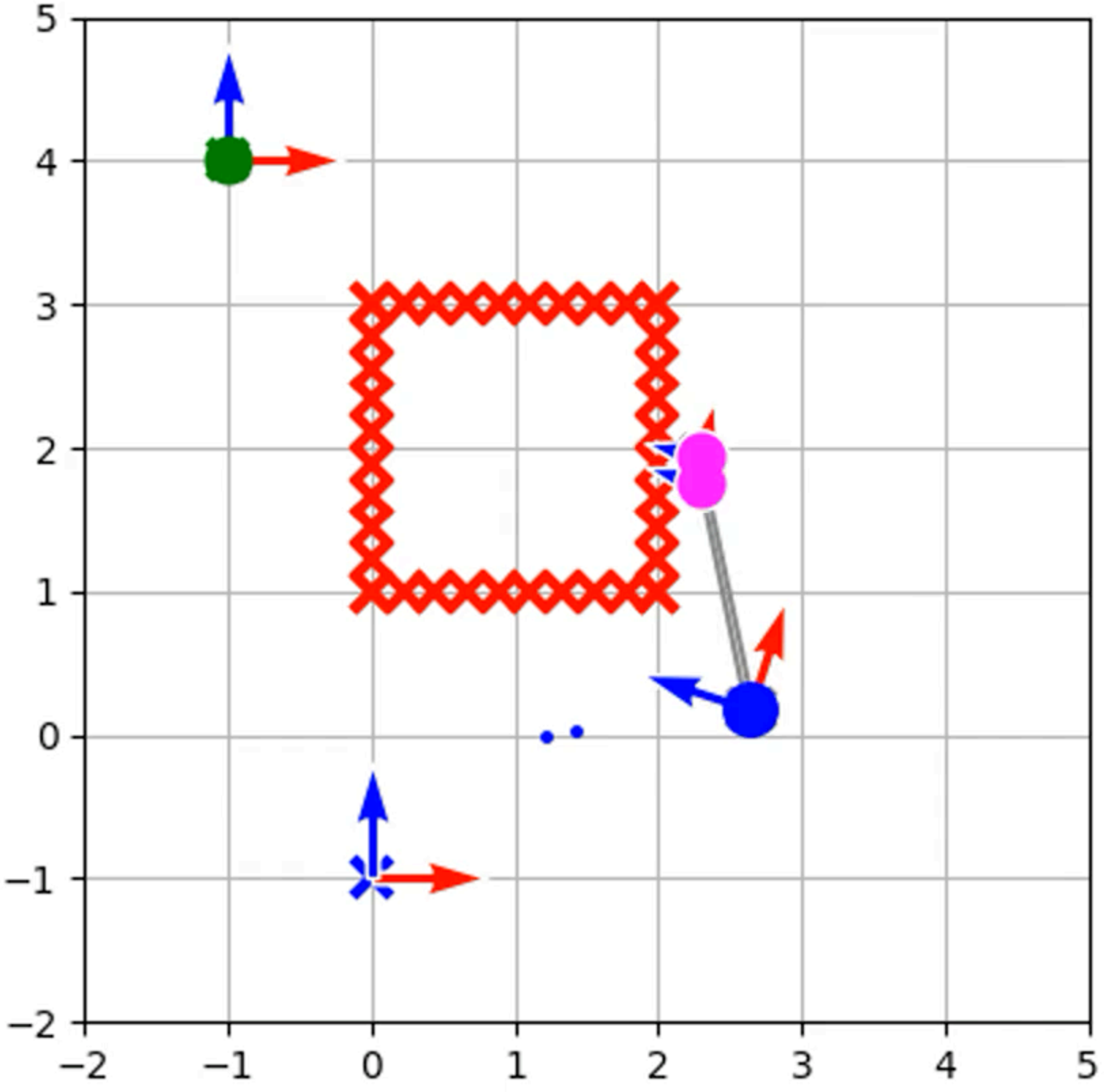
Optimization on SE(2)/SE(3) manifolds

Karel Zimmermann

graph-SLAM



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

● $\mathbf{m}^{\text{rel}}$........ estimated rel. marker position

● $\mathbf{m}^{\text{abs}}$........ known abs. marker position

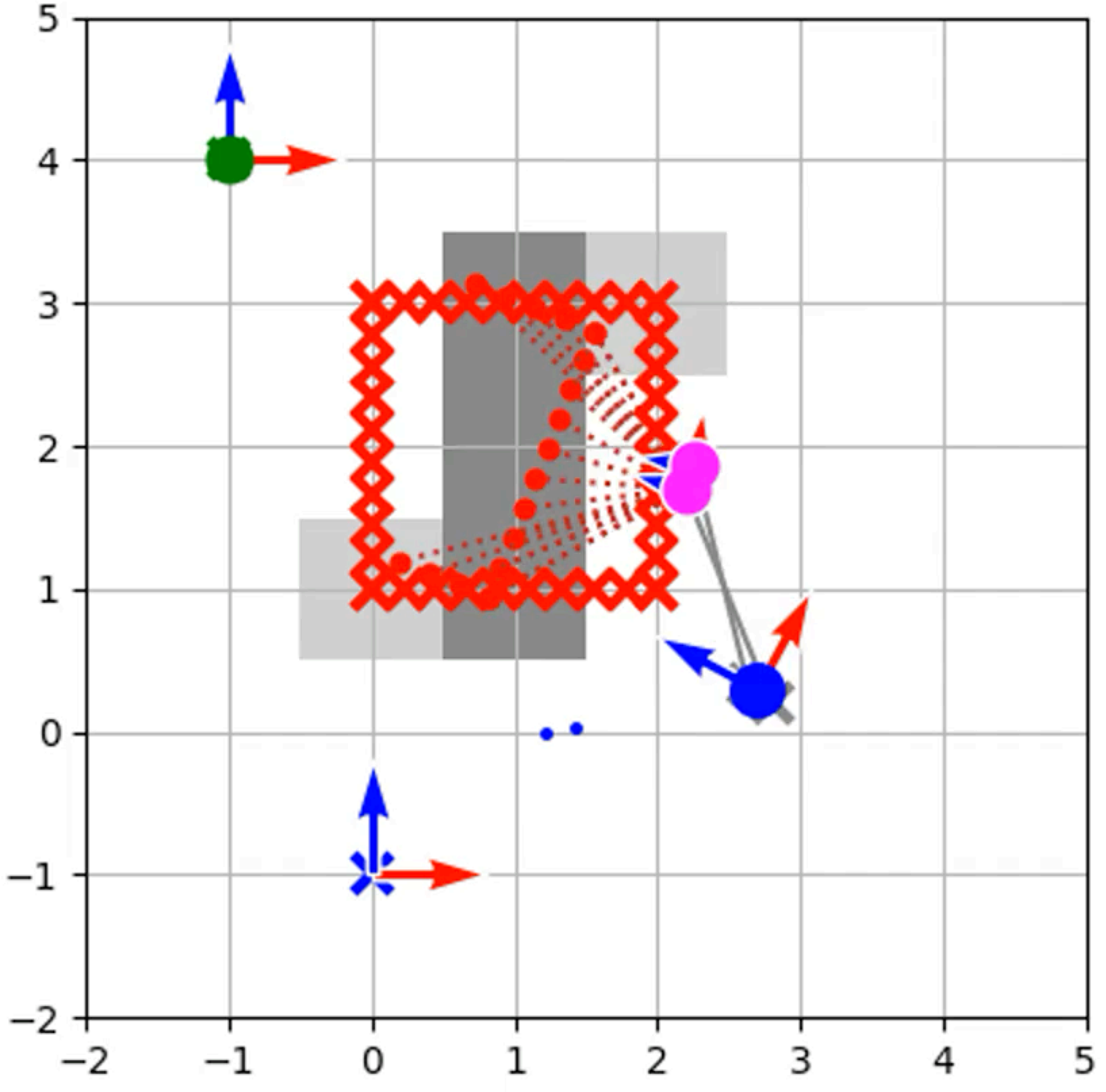
●— $\mathbf{z}_t^{\mathbf{m}^{\text{rel}}}, \mathbf{z}_t^{\mathbf{m}^{\text{abs}}}$... marker measurements

↑→ ...... local coordinate frame

× ...... ground truth pointcloud map

.... ground truth trajectory

graph-SLAM with dynamic gridmap construction



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

● $\mathbf{m}^{\text{rel}}$....... estimated rel. marker position

● $\mathbf{m}^{\text{abs}}$....... known abs. marker position

●— $\mathbf{z}_t^{\mathbf{m}^{\text{rel}}}, \mathbf{z}_t^{\mathbf{m}^{\text{abs}}}$... marker measurements

●··· $\mathbf{p}_t^i$ .... pointcloud measurements

↑→ ...... local coordinate frame

× ...... ground truth pointcloud map

••••• ...... ground truth trajectory

▨ ...... estimated gridmap

graph-SLAM formulations

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

$\mathbf{x} \nearrow$

$$+ \overset{\text{priors}}{\sum_t \|\mathbf{x}_t - \mathbf{x}_t^{prior}\|^2_{\Sigma_t^{prior}}} + \overset{\text{motion model}}{\sum_t \|g(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t^g}} + \overset{\text{loop-closures}}{\sum_t \|\mathrm{w2r}(\mathbf{x}_0, \mathbf{x}_T)\|^2_{\Sigma_t^{lc}}}$$

Optimization

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

$$+ \sum_t \|\mathbf{x}_t - \mathbf{x}_t^{prior}\|^2_{\Sigma_t^{prior}} + \sum_t \|g(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t^g} + \sum_t \|\mathrm{w2r}(\mathbf{x}_0, \mathbf{x}_T)\|^2_{\Sigma_t^{lc}}$$

what is f dimensionality?

$$= \arg\min_{\mathbf{x}} \sum_i \|f_i(\mathbf{x})\|^2 = \arg\min_{\mathbf{x}} \left\| \begin{matrix} f_1(\mathbf{x}) \\ \vdots \\ f_N(\mathbf{x}) \end{matrix} \right\|^2 = \arg\min_{\mathbf{x}} \|f(\mathbf{x})\|^2$$

where $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$

$f'(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^{m \times n}$

vector of residuals

Alternative formulation: $\arg\min_{\Delta\mathbf{x}} \|f(\mathbf{x}_k + \Delta\mathbf{x})\|^2$ where $\mathbf{x}_k$ is an initial solution

$$\approx \arg\min_{\Delta\mathbf{x}} \|f(\mathbf{x}_k) + f'(\mathbf{x}_k)\Delta\mathbf{x}\|^2 = -[f'(\mathbf{x}_k)]^+ f(\mathbf{x}_k)$$

GN: $\mathbf{x}_{k+1} = \mathbf{x}_k - [f'(\mathbf{x}_k)]^+ f(\mathbf{x}_k)$

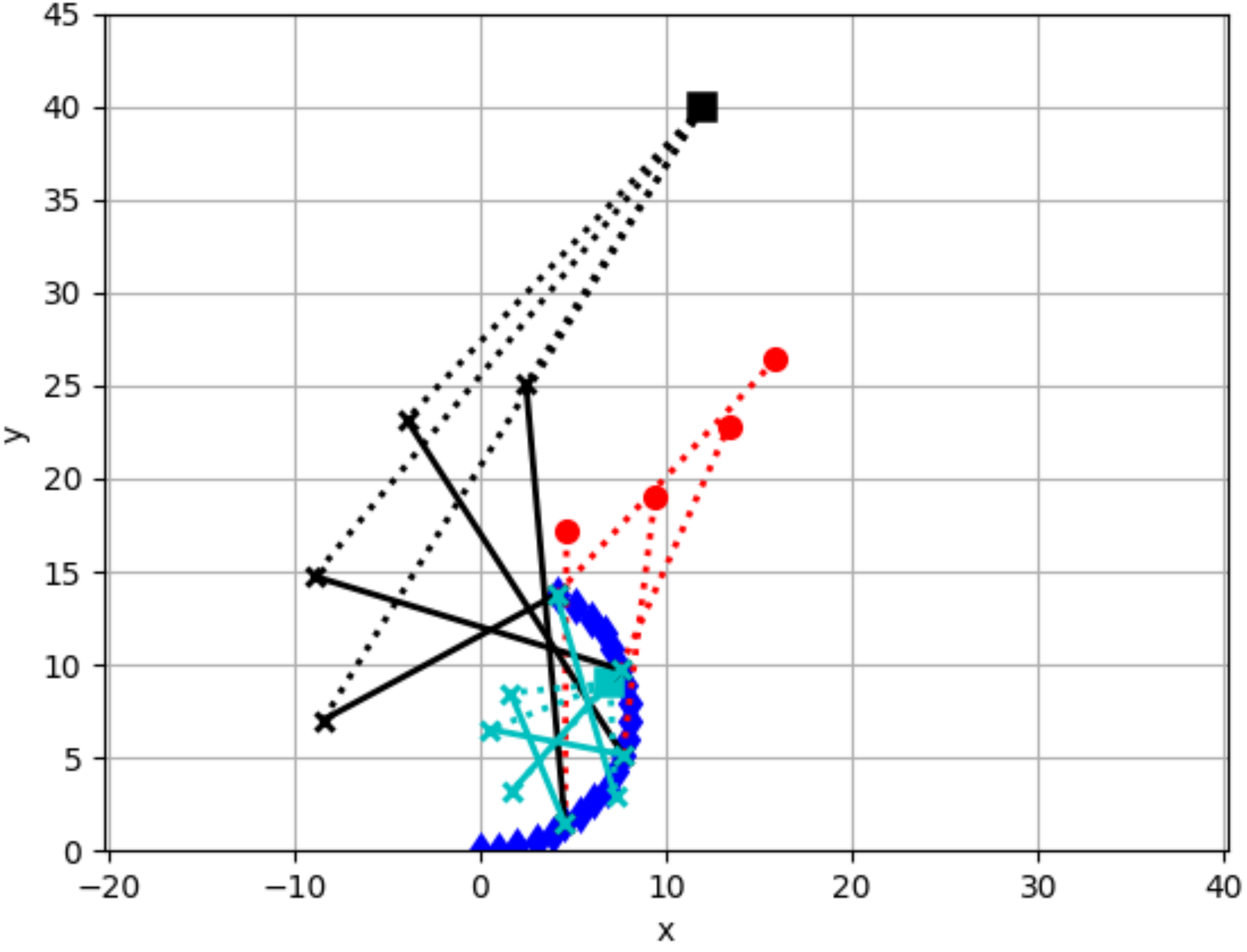
$$\approx \arg\min_{\Delta\mathbf{x}} \|f(\mathbf{x}_k) + f'(\mathbf{x}_k)\Delta\mathbf{x}\|^2 = -[f'(\mathbf{x}_k) + \lambda\mathbf{I}]^+ f(\mathbf{x}_k)$$

subject to $\|\Delta\mathbf{x}\|^2 \leq c$

LM:
TR: $\mathbf{x}_{k+1} = \mathbf{x}_k - [f'(\mathbf{x}_k) + \lambda\mathbf{I}]^+ f(\mathbf{x}_k)$

```
scipy.optimize.least_squares(fun, x0, jac, method='lm')
```

Optimization in SE(2) manifold trajectory length 21



■ absolute marker

■ relative marker

■ odometry

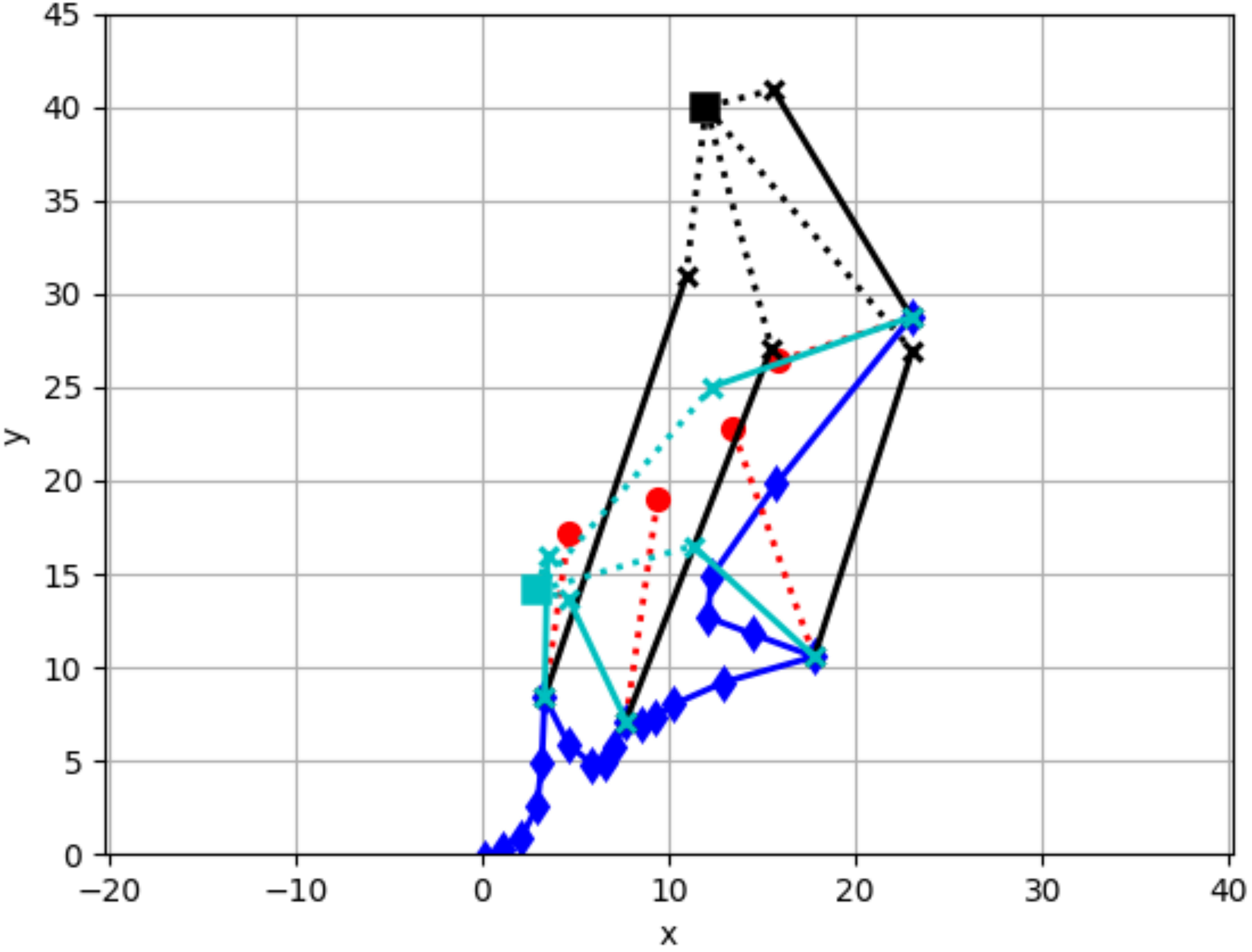
✕ some optimised poses

● some ground truth poses
(not used in optimisation)

noise:

- odom 0.2m / 0.2rad
- markers 0.3m / 0.3rad

Optimization in SE(2) manifold trajectory length 21



- ⬛ absolute marker
- 🟦 relative marker
- 🟦 odometry
- ☒ some optimised poses
- 🔴 some ground truth poses (not used in optimisation)

noise:

- odom 0.2m / 0.2rad
- markers 0.3m / 0.3rad

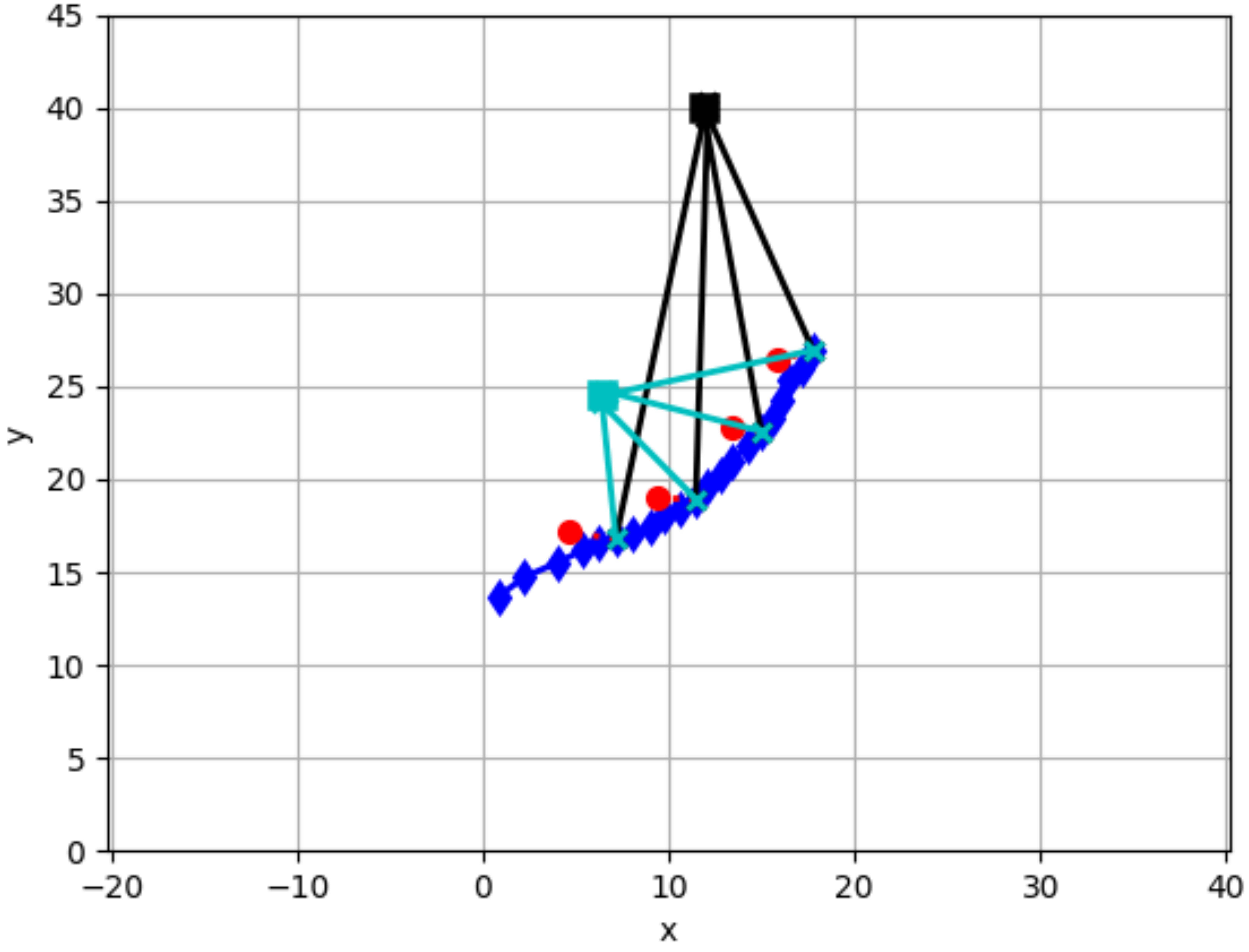
Optimization in SE(2) manifold trajectory length 21

45
40
35
30
25
20
15
10
5
0
y
−20
−10
0
10
20
30
40
x
■ absolute marker
■ relative marker
■ odometry
✕ some optimised poses
● some ground truth poses
(not used in optimisation)

noise:

- odom 0.2m / 0.2rad
- markers 0.3m / 0.3rad

Optimization in SE(2) manifold trajectory length 21

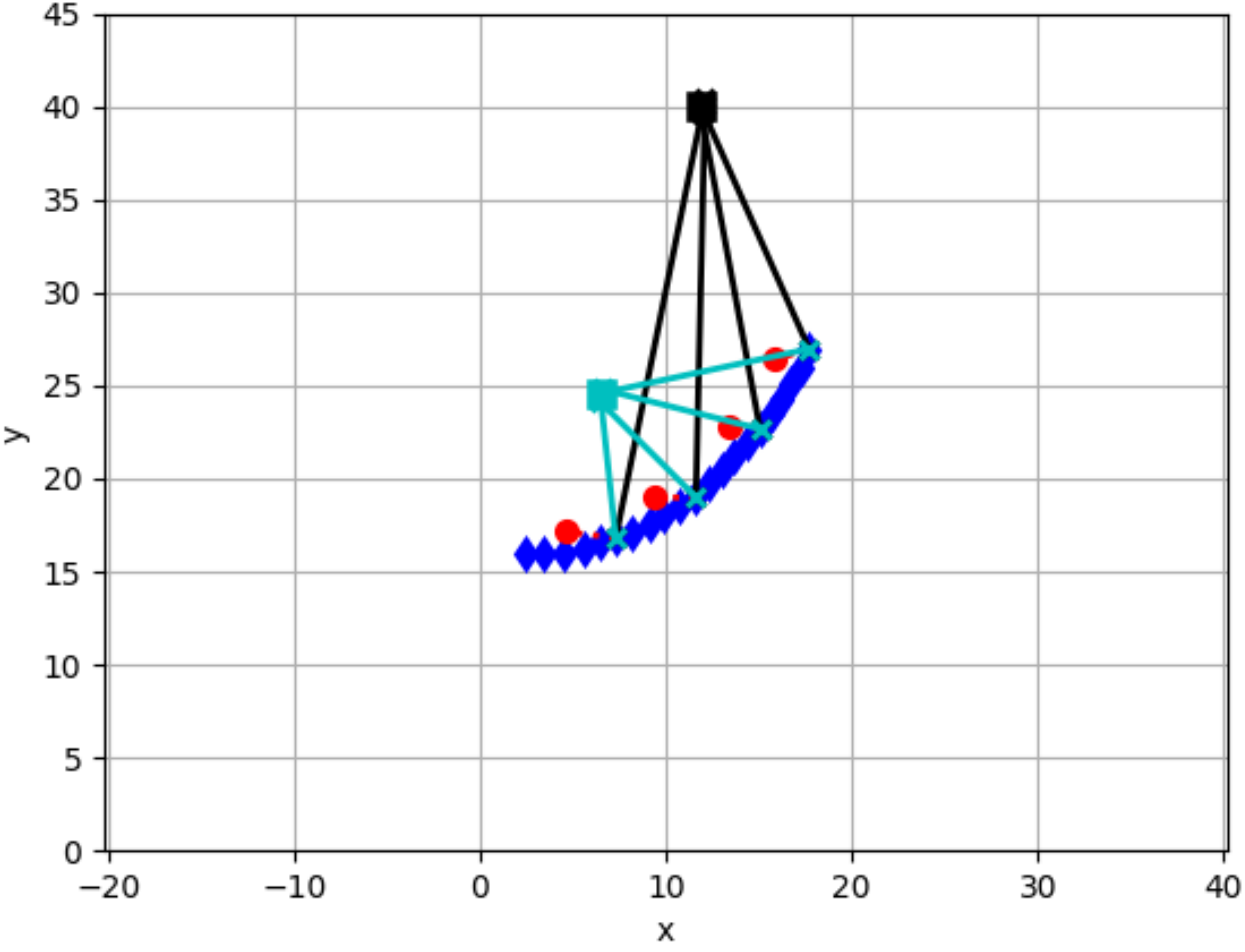


- ⬛ absolute marker
- 🟦 relative marker
- 🟦 odometry
- ✖ some optimised poses
- 🔴 some ground truth poses (not used in optimisation)

noise:

- ○ odom 0.2m / 0.2rad
- ○ markers 0.3m / 0.3rad

Optimization in SE(2) manifold trajectory length 21



- ■ absolute marker
- ■ relative marker
- ■ odometry
- ☒ some optimised poses
- ● some ground truth poses (not used in optimisation)

noise:

- odom 0.2m / 0.2rad
- markers 0.3m / 0.3rad

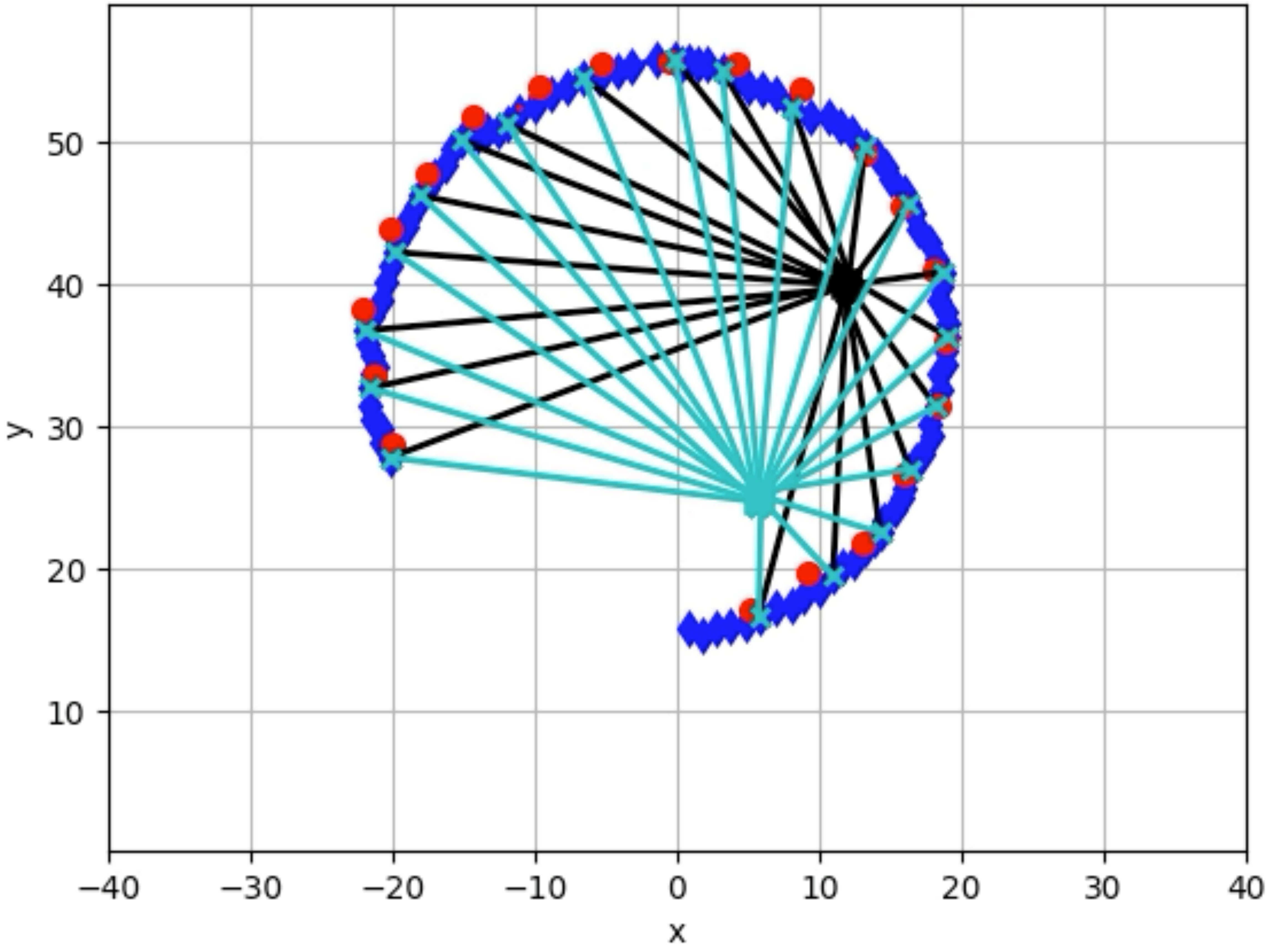
Optimization in SE(2) manifold trajectory length 101

What will break it????

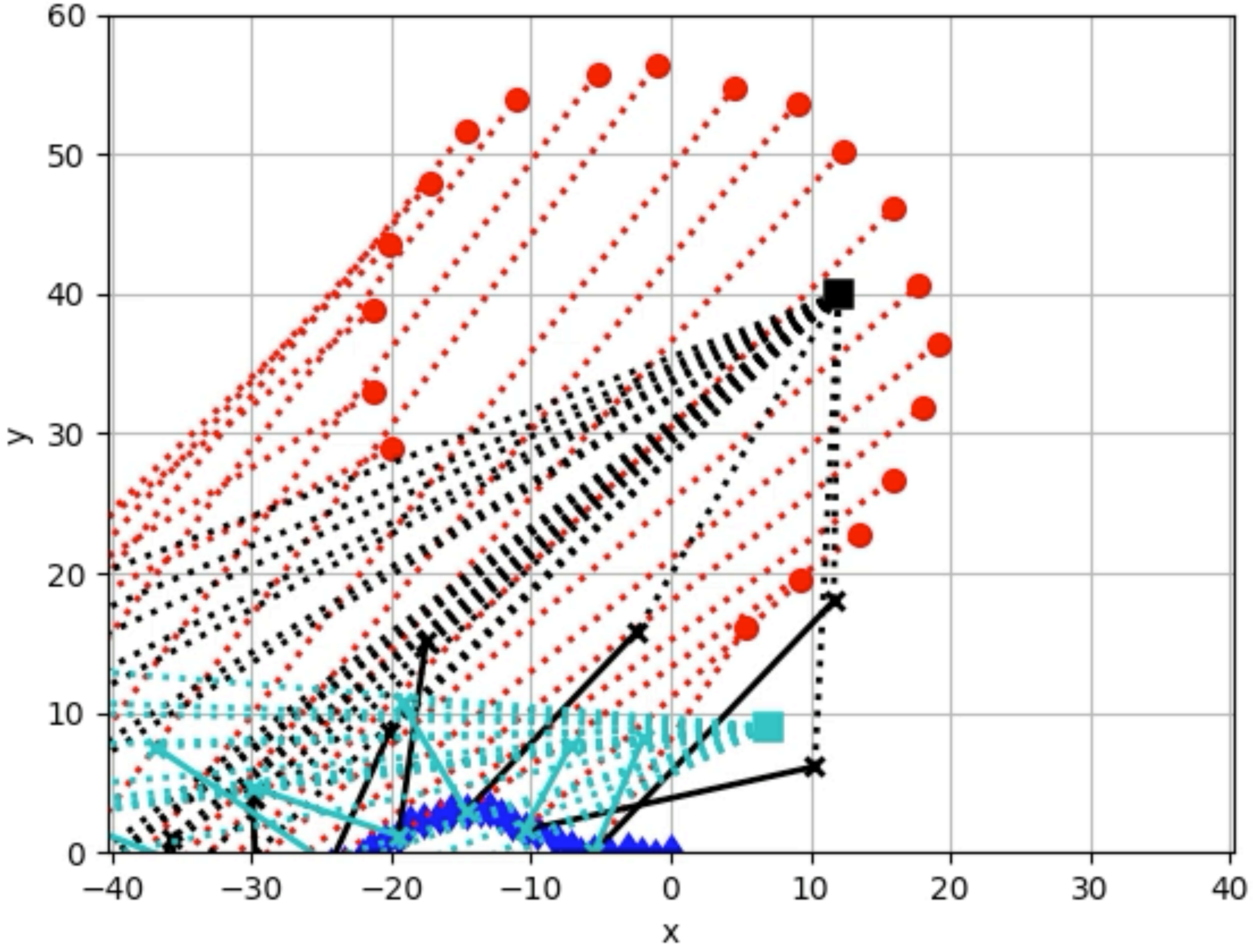
Optimization in SE(2) manifold trajectory length 101

noise 0.5
c_odom=0.2
c_ma=1
c_mr=1

odom/marker ini

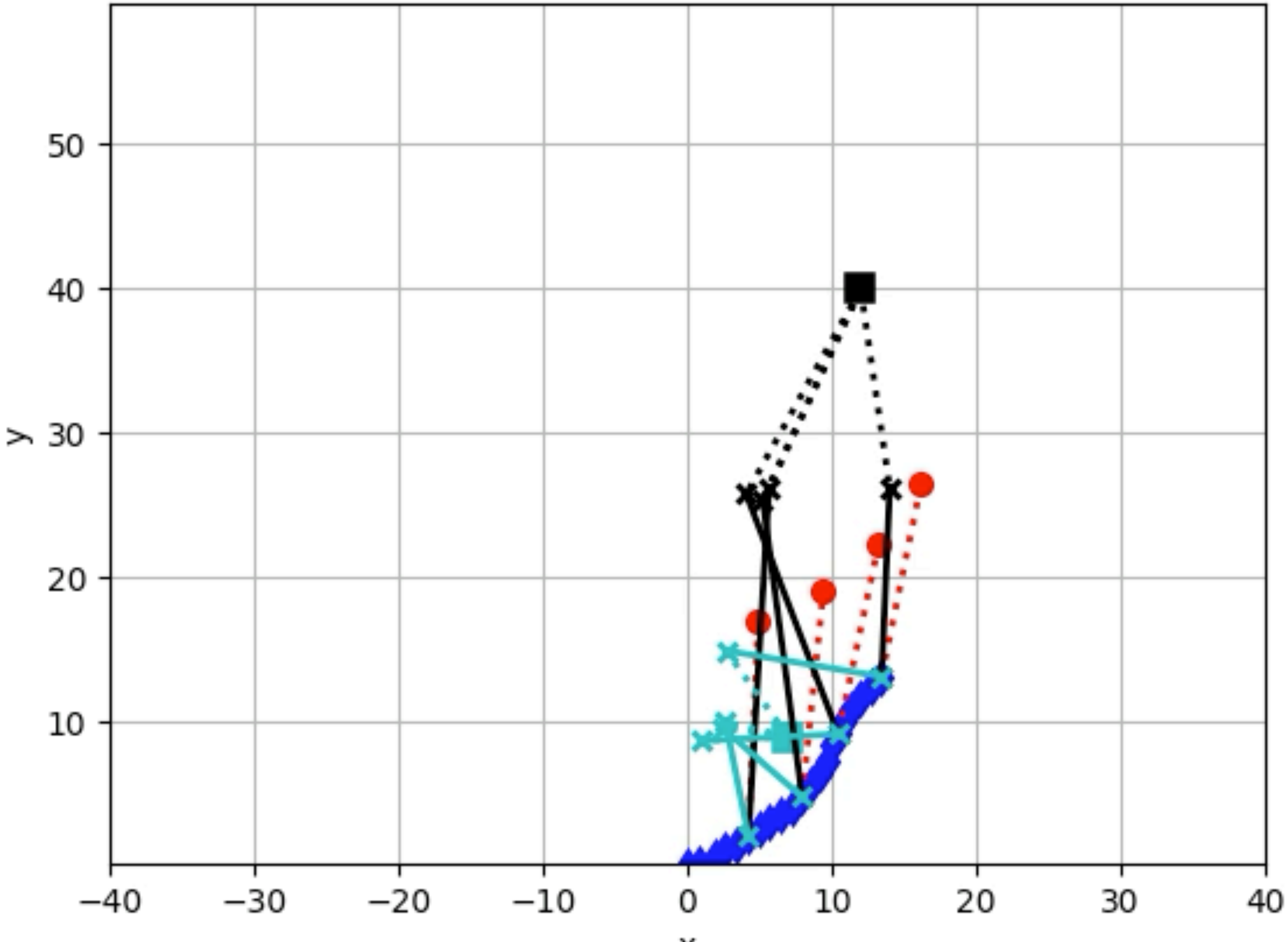


adversarial ini



Optimization in SE(2) manifold trajectory length 401 successive optimization with incoming measurements

noise_markers = 0.5   noise_odom = 0.1



Optimization

```
scipy.optimize.least_squares(fun, x0, jac, method=
```

Karel, you don’t know how to code stuff properly
Martin, then show me how nerds do the job!
What about 50ms?
Optimization time [s]
10
5
1
LM (jac:num)
LM (autodiff)
LM (jac:analytical)
LM (jac:analytical+sparse)
trajectory length
0
50
100
200
300
400

Optimization

```
scipy.optimize.least_squares(fun, x0, jac, method=‘lm’)
```

Solution time grows fast with:

- problem dimensionality (e.g. DOFxT+M)
- number of residual terms (e.g. number of measurements)

In practise you introduce simplifications:

- jacobian is extremely sparse => use sparse matrix to represent it
- when new measurement comes only a sub-graph is optimized
- pre-integrate some factor (e.g. sum up odometry measurements over 0.5s)
- sparsification of old factor graph
- to tackle the real-time requirements frontend and backend optimizers used
- limited temporal horizon considered

Summary

- **Understand** SLAM problem in SE(2)
- **Write down optimisation** criterion in negative log-space for gaussian prob. distr.
- **Solve** underlying opt. problem using non-linear least squares
- **Issues:**
  - covariance delivered by sensors is really bad
  - measurements are strongly correlated
  - gradient optimization converges to a local minimum
  - noise often non-gaussian => if modeled optimization issues
  - factor graph keep growing to infinity vs realtime requirements
- **Next lecture:** Adds lidar’s measurement probability