Advanced Techniques for Mobile Robotics

Simultaneous Calibration, Localization, and Mapping

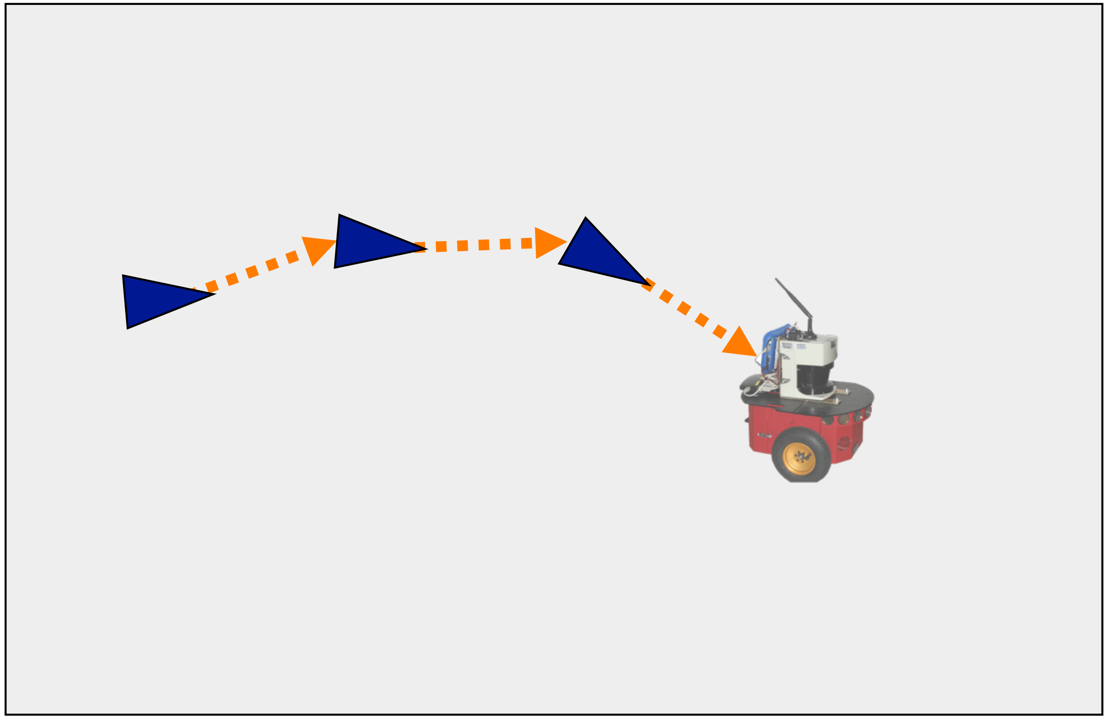
Wolfram Burgard, Cyrill Stachniss,

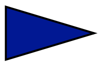
Kai Arras, Maren Bennewitz

UNI FREIBURG

SLAM

- Constraints connect the poses of the robot while it is moving
- Constraints are inherently uncertain

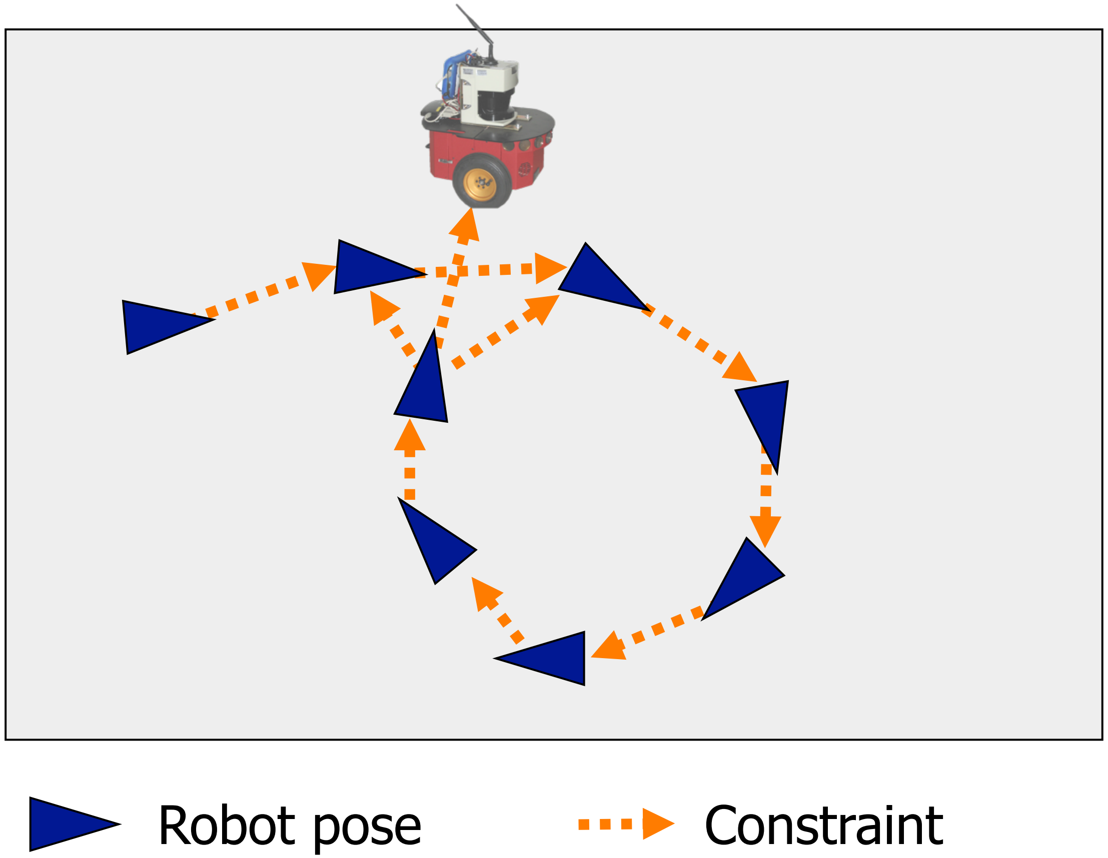



Robot pose


Constraint

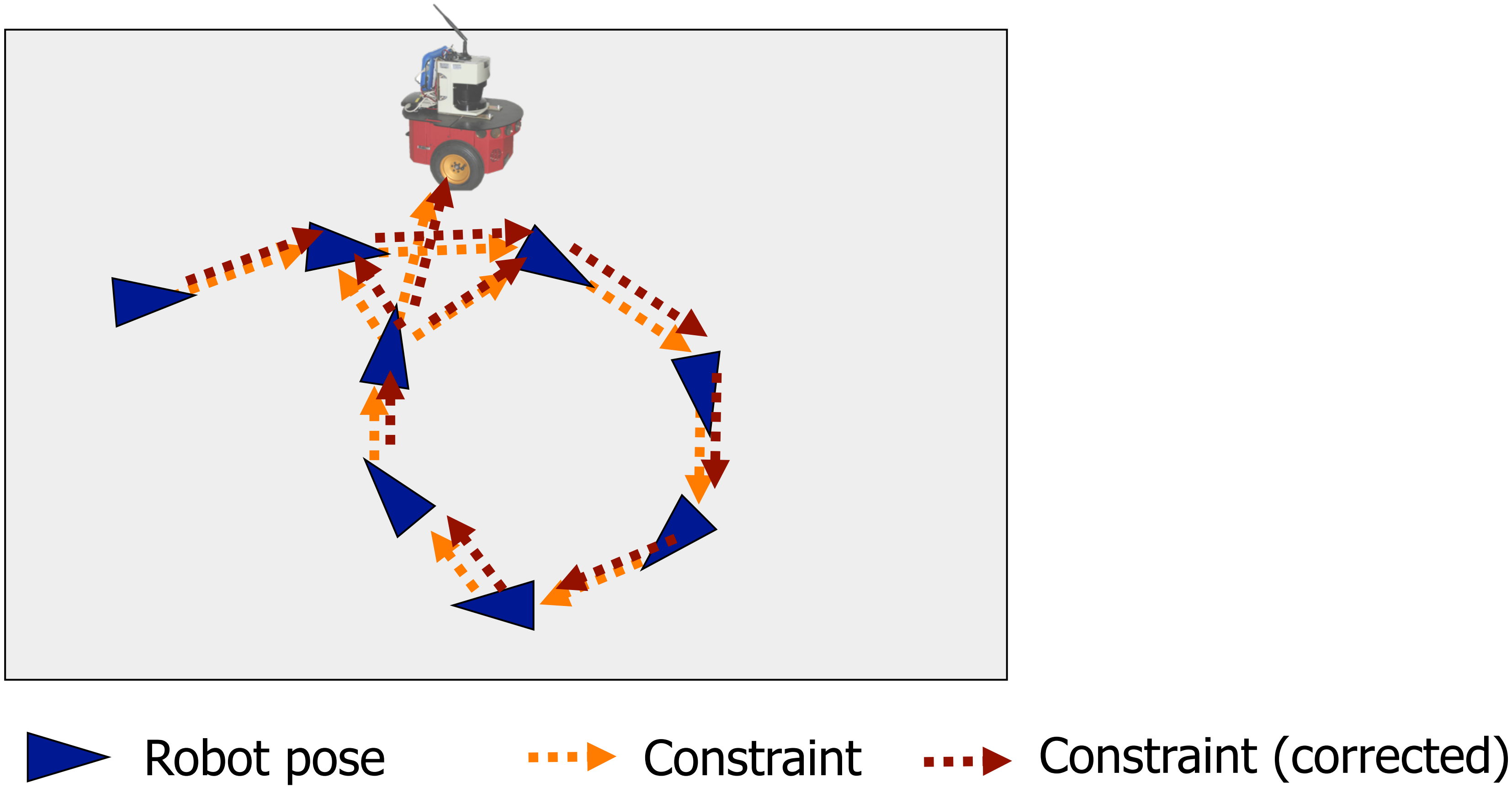
SLAM

- Observing previously seen areas generates constraints between non-successive poses
- Constraints are inherently uncertain



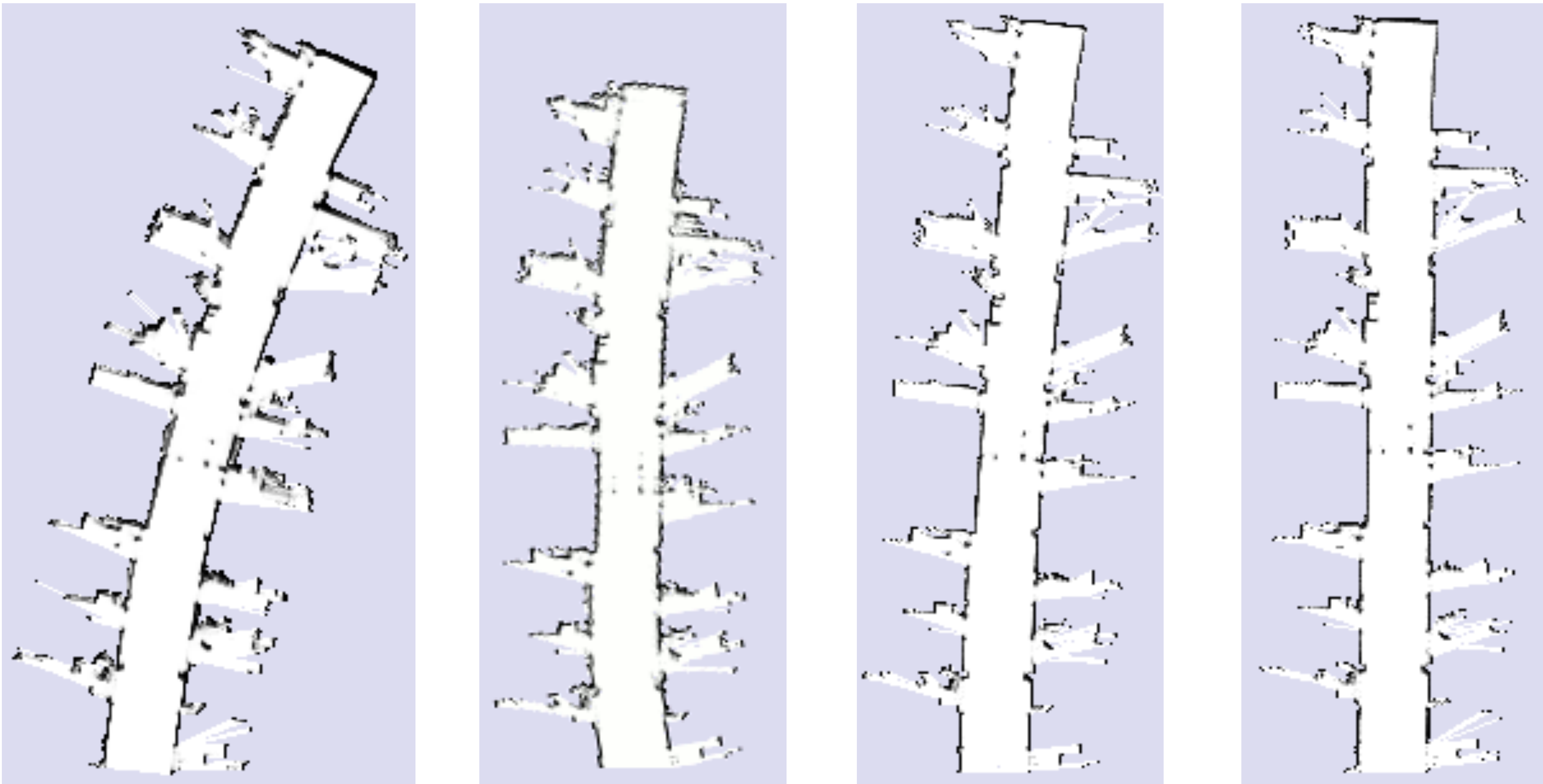
SCLAM – Adding Calibration

- Eliminate systematic errors
  - Location of the sensor on the robot
  - Systematic odometry errors



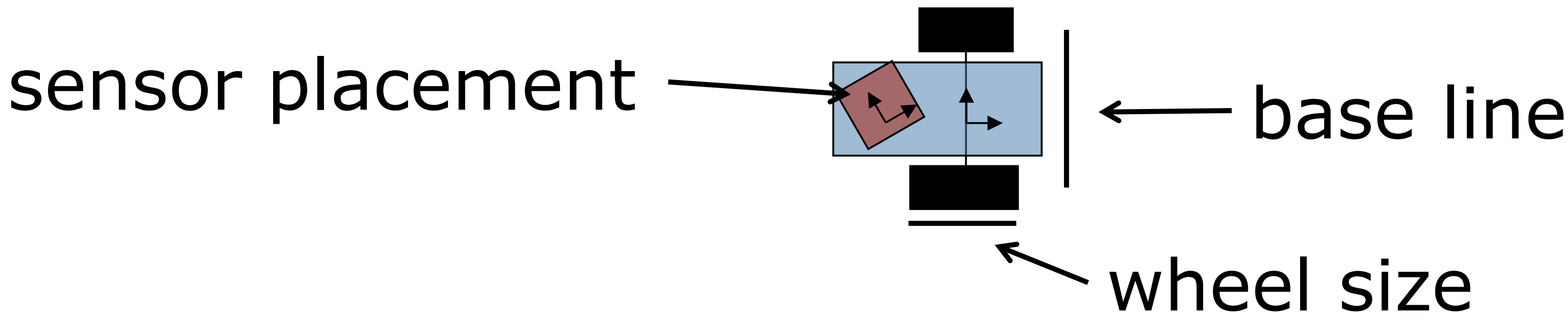
Relevance

- Systematic errors can strongly influence the results of a mapping system



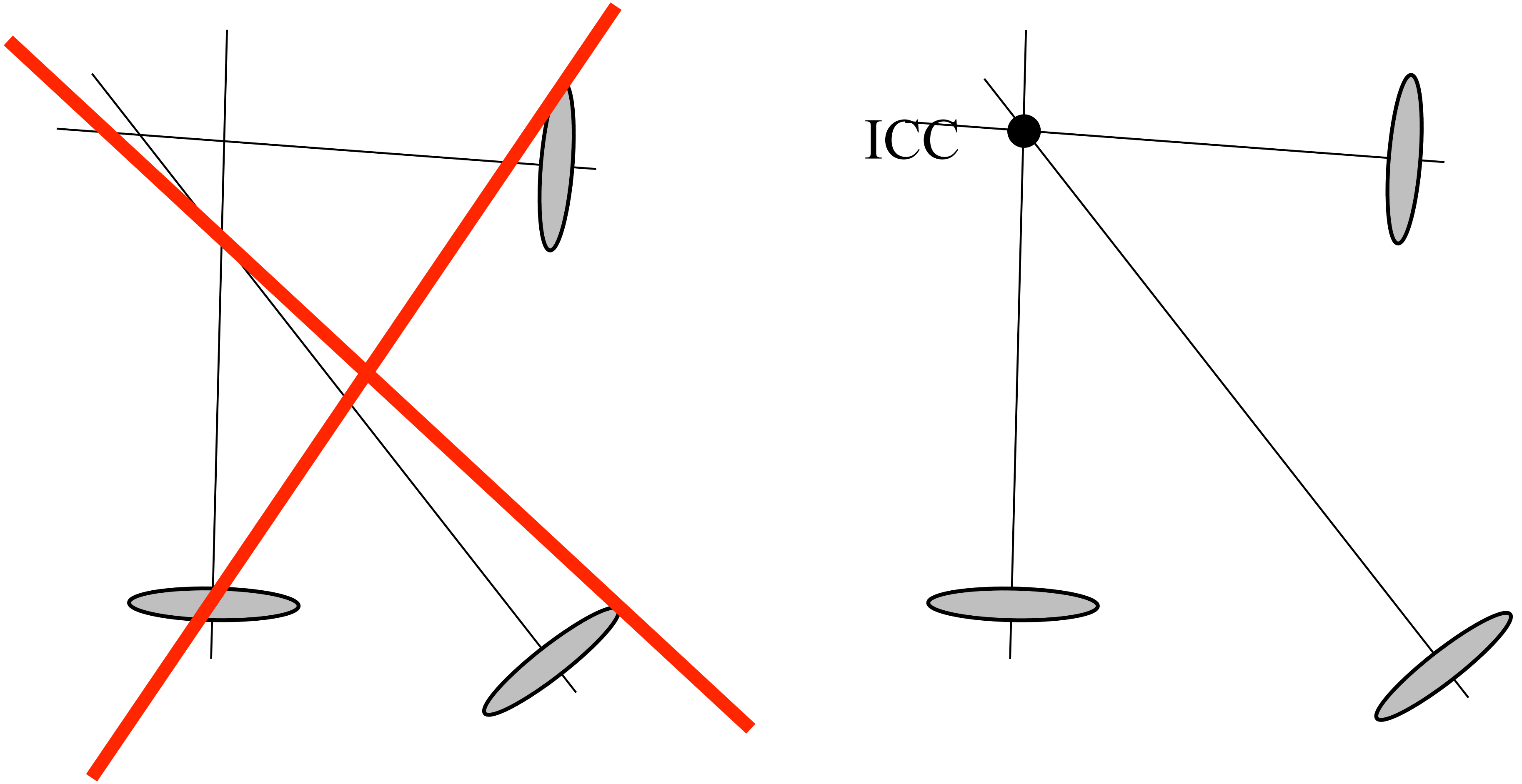
Key Idea

- Extend graph-based SLAM to estimate also systematic errors
- Explicitly model that the measurements are obtained in a different coordinate frame
- Estimate the forward kinematics parameters
- Allow for online optimization (e.g., when the robot carries a load)



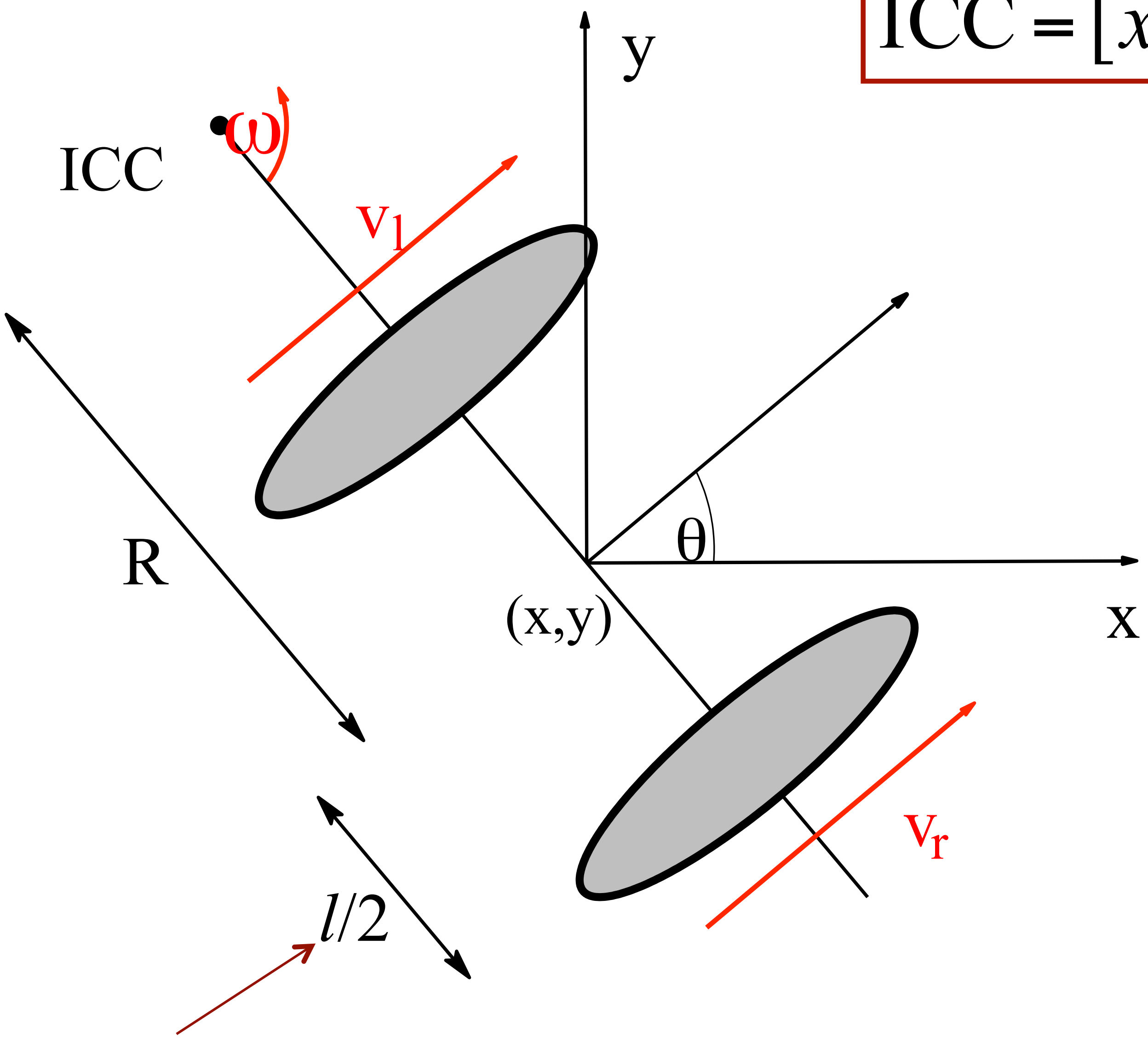
Robotics I

Instantaneous Center of Curvature



- For rolling motion to occur, each wheel has to move along its y-axis

Robotics I
Differential Drive



=b in this lecture

$$\boxed{\mathrm{ICC} = [x - R\sin\theta, y + R\cos\theta]}$$

$$\omega(R + l/2) = v_r$$

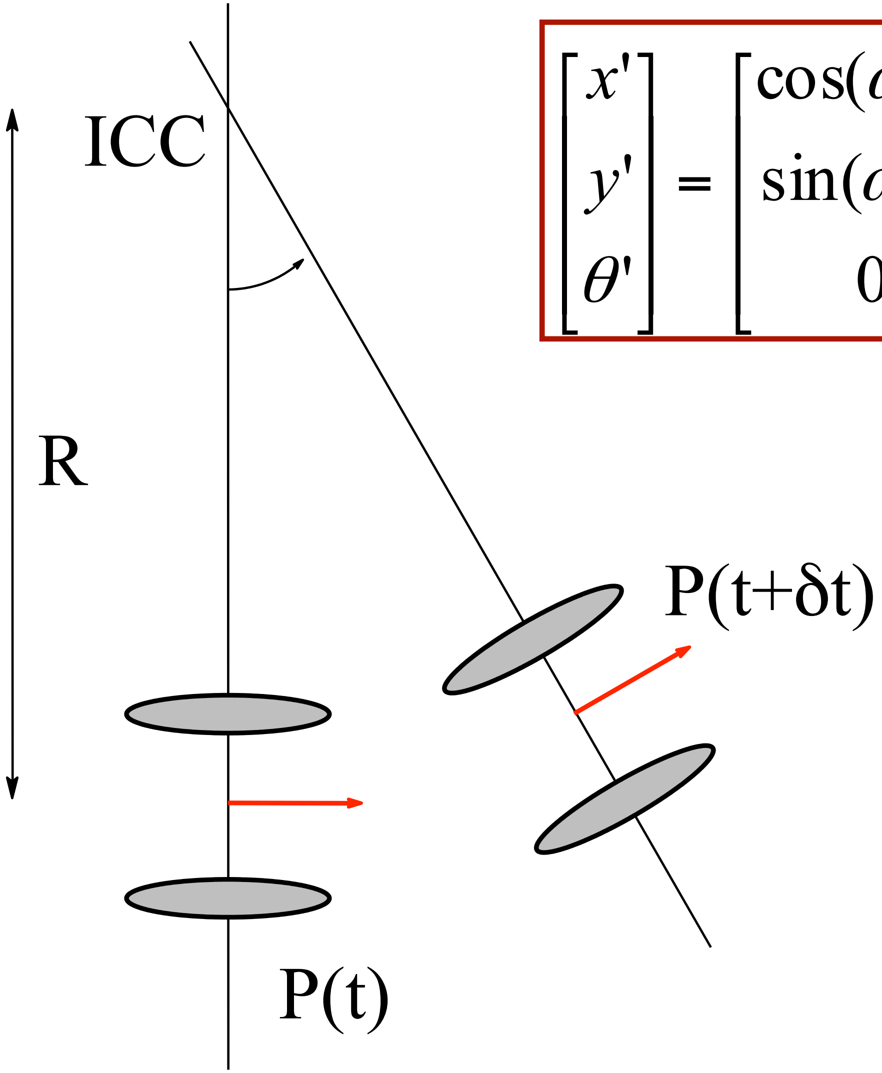
$$\omega(R - l/2) = v_l$$

$$R = \frac{l}{2}\frac{(v_l + v_r)}{(v_r - v_l)}$$

$$\omega = \frac{v_r - v_l}{l}$$

$$v = \frac{v_r + v_l}{2}$$

Robotics I
Forward Kinematics

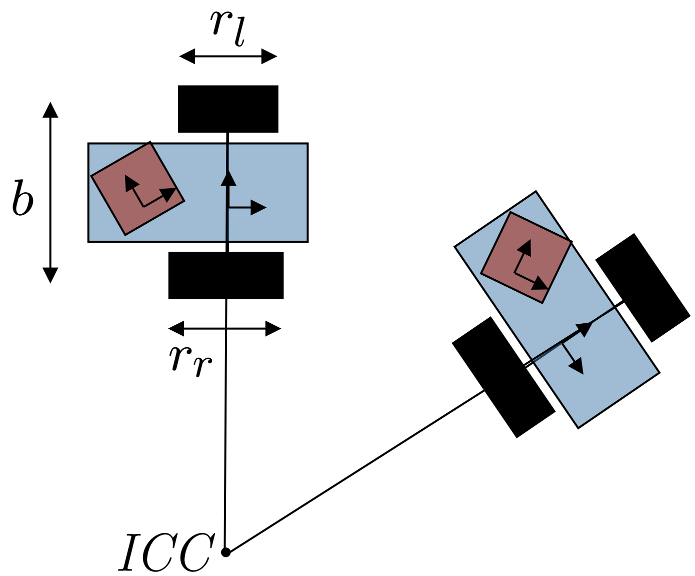


$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix}$$

Odometry Measurements

Forward kinematics for a differential drive robot:

$$
\begin{aligned}
K(\mathbf{u},\mathbf{k}) &= \begin{pmatrix} R(\Delta t\omega) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} -ICC \\ 0 \end{pmatrix} + \begin{pmatrix} ICC \\ \Delta t\omega \end{pmatrix} \\
\mathbf{k} &= \begin{pmatrix} r_l \\ r_r \\ b \end{pmatrix} \qquad \mathbf{u} = \begin{pmatrix} v_l \\ v_r \end{pmatrix} \\
\omega &= \frac{v_r r_r - v_l r_l}{b}
\end{aligned}
$$



Error function:

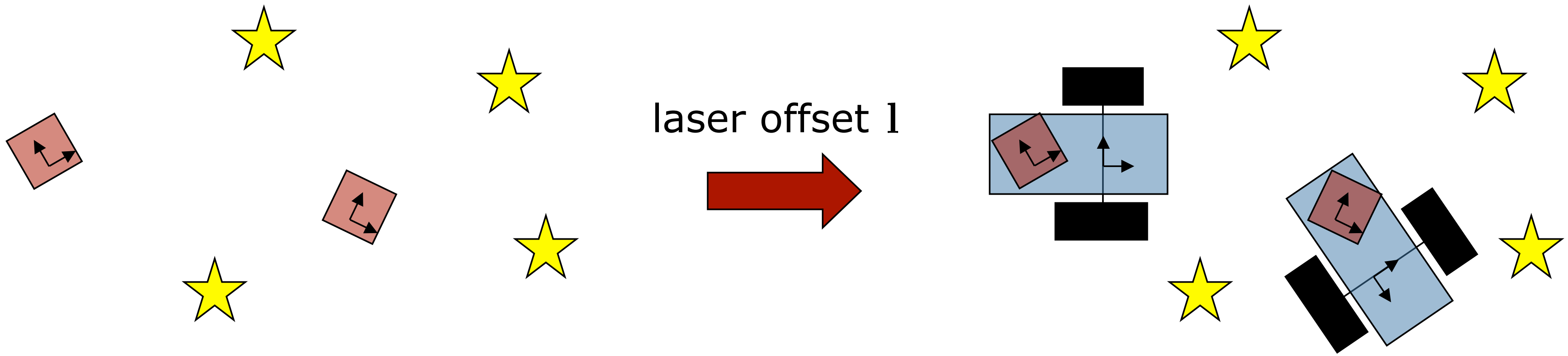
forward kinematics function and its parameters

$$\mathbf{e}_i^{\mathbf{u}}(\mathbf{x}) = \left(\mathbf{x}_{i+1} \ominus \mathbf{x}_i\right) \ominus K(\mathbf{u}_i, \mathbf{k})$$

robot poses odometry

Sensor Measurements

The observations (★) allow to estimate the ego-motion of the sensor and thus the motion of the robot, **given** the position of the sensor on the robot.



Error function:

position of the sensor on the robot

$$\mathbf{e}^{\mathbf{l}}_{ij}(\mathbf{x}) = \Big((\mathbf{x}_j \oplus \mathbf{l}) \ominus (\mathbf{x}_i \oplus \mathbf{l})\Big) \ominus \mathbf{z}_{ij}$$

robot poses

scan matching

Graph Optimization

- Combine both types of measurements
- Find the minimum of the error function:

$$\begin{aligned}\mathbf{F}(\mathbf{x},\mathbf{l},\mathbf{k}) &= \sum_{\langle i,j\rangle} \mathbf{e}^{\mathbf{l}}_{ij}(\mathbf{x})^\top \boldsymbol{\Omega}^{\mathbf{z}}_{ij} \mathbf{e}^{\mathbf{l}}_{ij}(\mathbf{x}) + \\ &\quad \sum_i \mathbf{e}^{\mathbf{u}}_i(\mathbf{x})^\top \tilde{\boldsymbol{\Omega}}^{\mathbf{u}}_i \mathbf{e}^{\mathbf{u}}_i(\mathbf{x})\end{aligned}$$

Graph Optimization (2)

- Without loss of generality: $\mathbf{y} := (\mathbf{x}, \mathbf{k}, \mathbf{l})^\top$
- Find the minimum of the error function

SPD information matrix

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}} \sum_k \mathbf{e}_k^T(\mathbf{y})\Omega_k\mathbf{e}_k(\mathbf{y})$$

error function

state vector

- Minimization by applying methods such as Gauss-Newton or Levenberg-Marquardt

Iterative Solution (1)

- Linearize the error around the current solution $\boldsymbol{y_0}$ by fixing $\boldsymbol{y}$ and varying a small increment $\boldsymbol{\Delta y}$

$$\mathbf{e}_k(\mathbf{y}_0 \oplus \boldsymbol{\Delta}\mathbf{y}) = \mathbf{e}_k + \mathbf{J}_k \boldsymbol{\Delta}\mathbf{y} \qquad \mathbf{J}_k = \left. \frac{\partial \mathbf{e}_k(\mathbf{y} \oplus \boldsymbol{\Delta}\mathbf{y})}{\partial \boldsymbol{\Delta}\mathbf{y}} \right|_{\boldsymbol{\Delta}\mathbf{y}=0}$$

- The error term in the neighborhood of the linearization becomes a quadratic form

$$\begin{aligned} e_k(\mathbf{y}_0 \oplus \boldsymbol{\Delta}\mathbf{y}) &= \mathbf{e}_k^T(\mathbf{y}_0 \oplus \boldsymbol{\Delta}\mathbf{y}) \boldsymbol{\Omega}_k \mathbf{e}_k(\mathbf{y}_0 \oplus \boldsymbol{\Delta}\mathbf{y}) \\ &\simeq \underbrace{\mathbf{e}_k^T \boldsymbol{\Omega}_k \mathbf{e}_k}_{c_k} + 2 \underbrace{\mathbf{e}_k^T \boldsymbol{\Omega}_k \mathbf{J}_k}_{\mathbf{b}_k^T} \boldsymbol{\Delta}\mathbf{y} + \boldsymbol{\Delta}\mathbf{y}^T \underbrace{\mathbf{J}_k^T \boldsymbol{\Omega}_k \mathbf{J}_k}_{\mathbf{H}_k} \boldsymbol{\Delta}\mathbf{y} \\ &= c_k + 2\mathbf{b}_k^T \boldsymbol{\Delta}\mathbf{y} + \boldsymbol{\Delta}\mathbf{y}^T \mathbf{H}_k \boldsymbol{\Delta}\mathbf{y} \end{aligned}$$

Iterative Solution (2)

- The same substitution can be applied to the global error function

$$
\begin{aligned}
F(\mathbf{y}_0 \oplus \mathbf{\Delta y}) &\simeq \sum_k \left(c_k + \mathbf{b}_k^T \mathbf{\Delta y} + \mathbf{\Delta y}^T \mathbf{H}_k \mathbf{\Delta y} \right) \\
&= \underbrace{\sum_k c_k}_{c} + 2 \underbrace{\left(\sum_k \mathbf{b}_k^T\right)^T}_{\mathbf{b}} \mathbf{\Delta y} + \mathbf{\Delta y}^T \underbrace{\left(\sum_k \mathbf{H}_k\right)}_{\mathbf{H}} \mathbf{\Delta y} \\
&= c + 2\mathbf{b}^T \mathbf{\Delta y} + \mathbf{\Delta y}^T \mathbf{H} \mathbf{\Delta y}
\end{aligned}
$$

Iterative Solution (3)

- The optimum of the quadratic form can be found by solving the linear system

$$\mathbf{H}\Delta\mathbf{y} \;=\; -\mathbf{b}$$

- or using the damped variant

$$(\mathbf{H} + \lambda\mathbf{I})\Delta\mathbf{y} \;=\; -\mathbf{b}$$

- The improved estimate is obtained by applying the perturbation to the previous guess

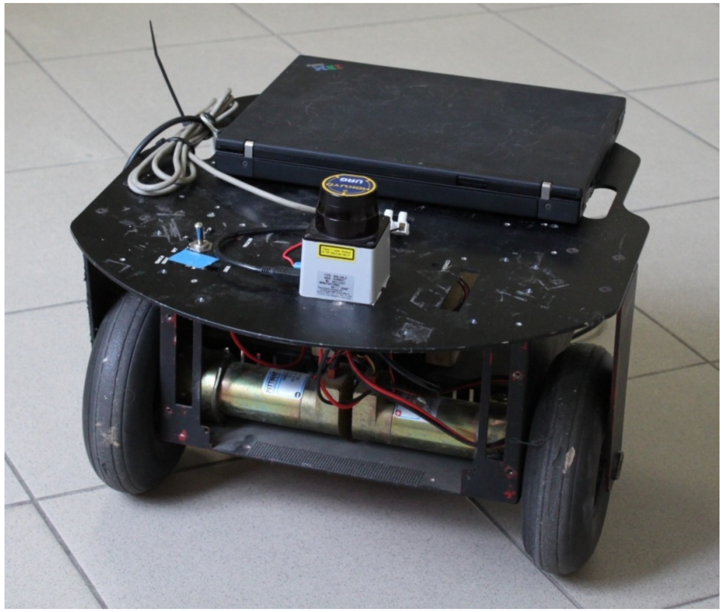
$$\mathbf{y}_0 \leftarrow \mathbf{y}_0 \oplus \Delta\mathbf{y}^*$$

How Does This Work in Practice?

- Three differential drive robots
- Equipped with laser range finders

PowerBot

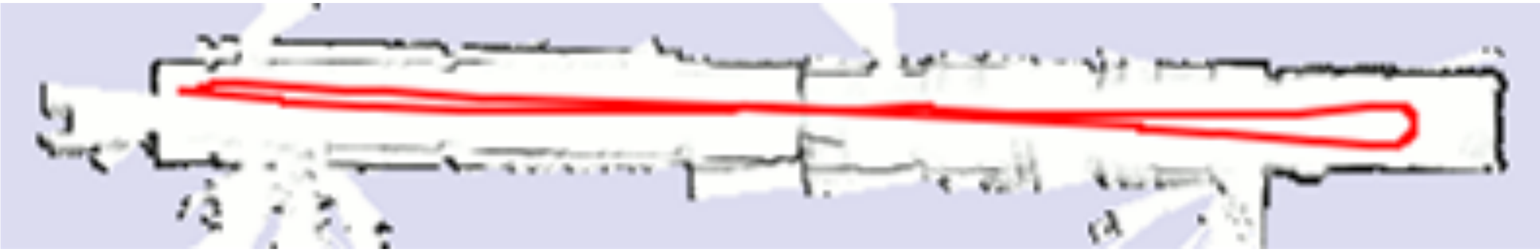
Custom



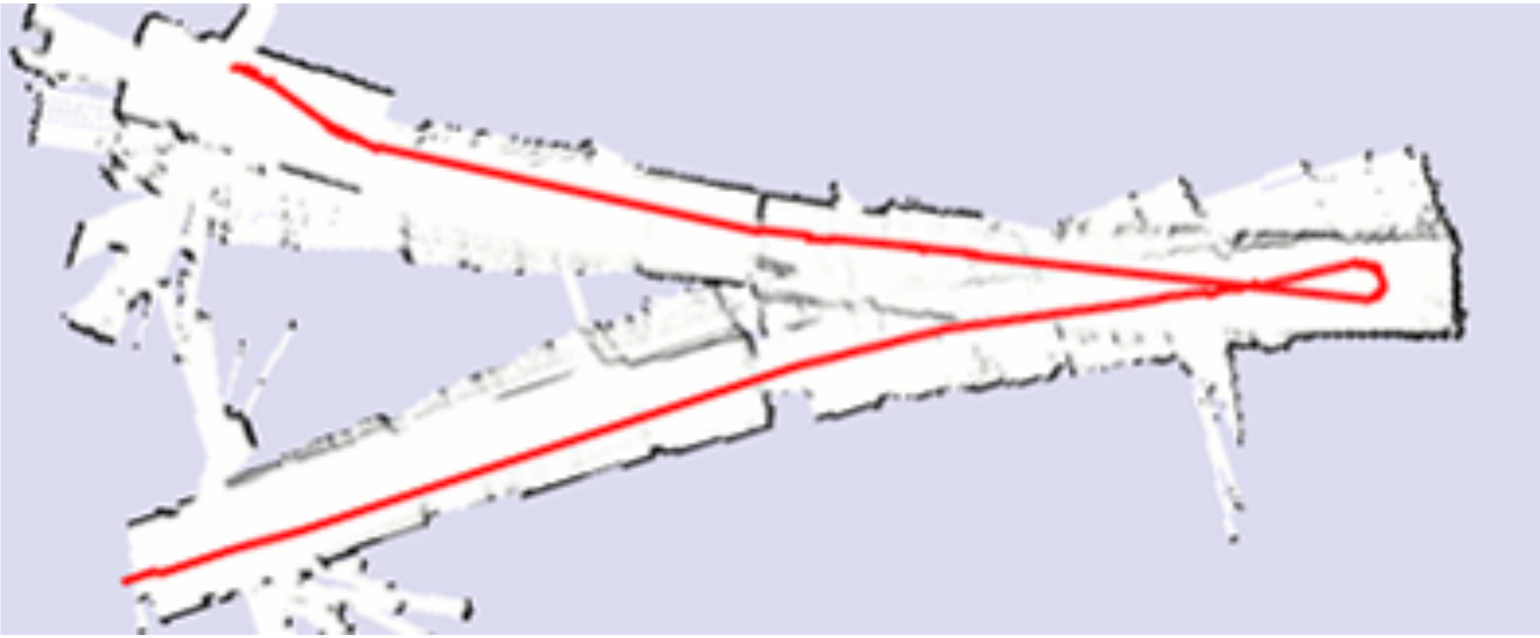
Pioneer I

Effect of the Odometry Parameters

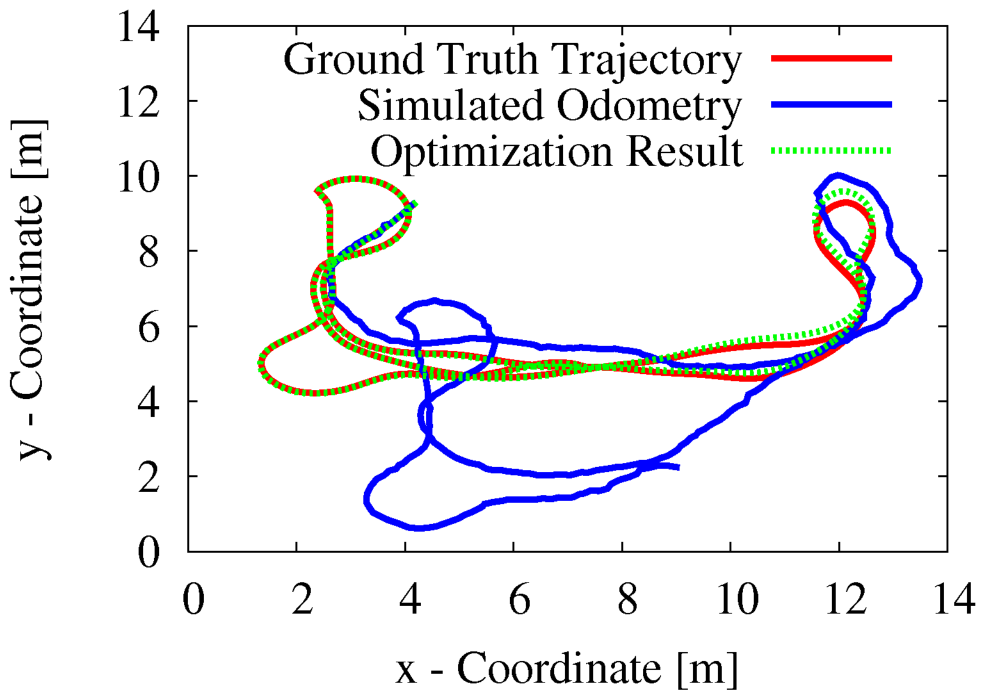
with online calibration

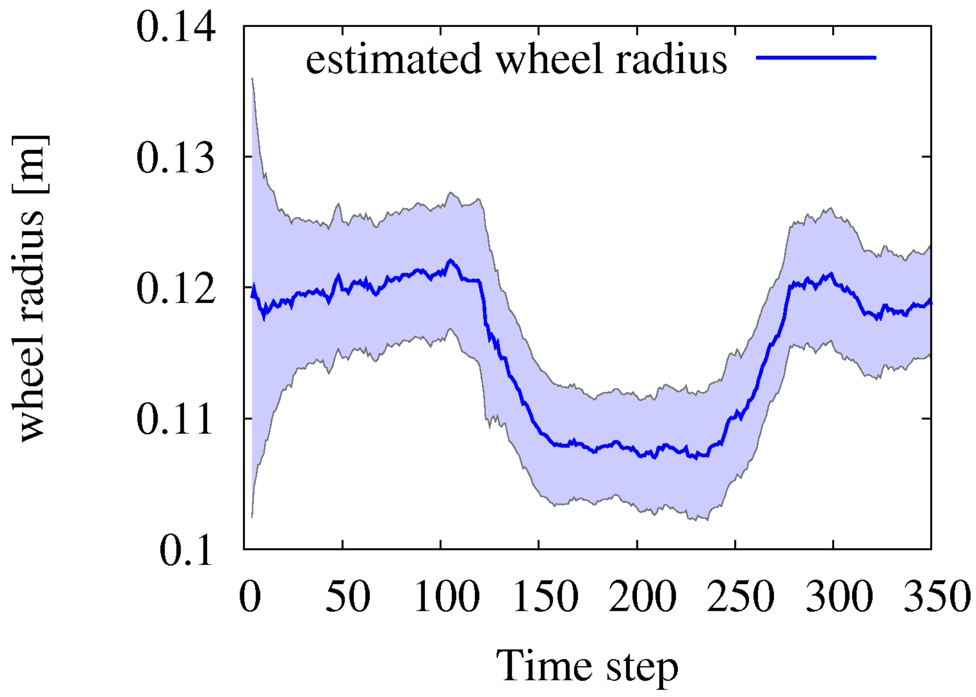


without online calibration



Simulation

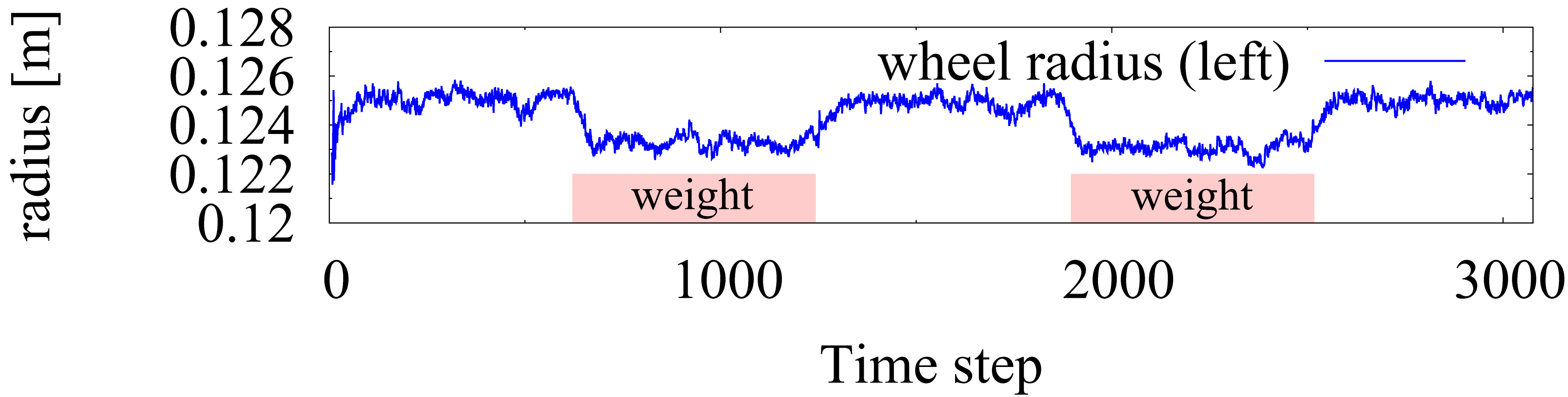




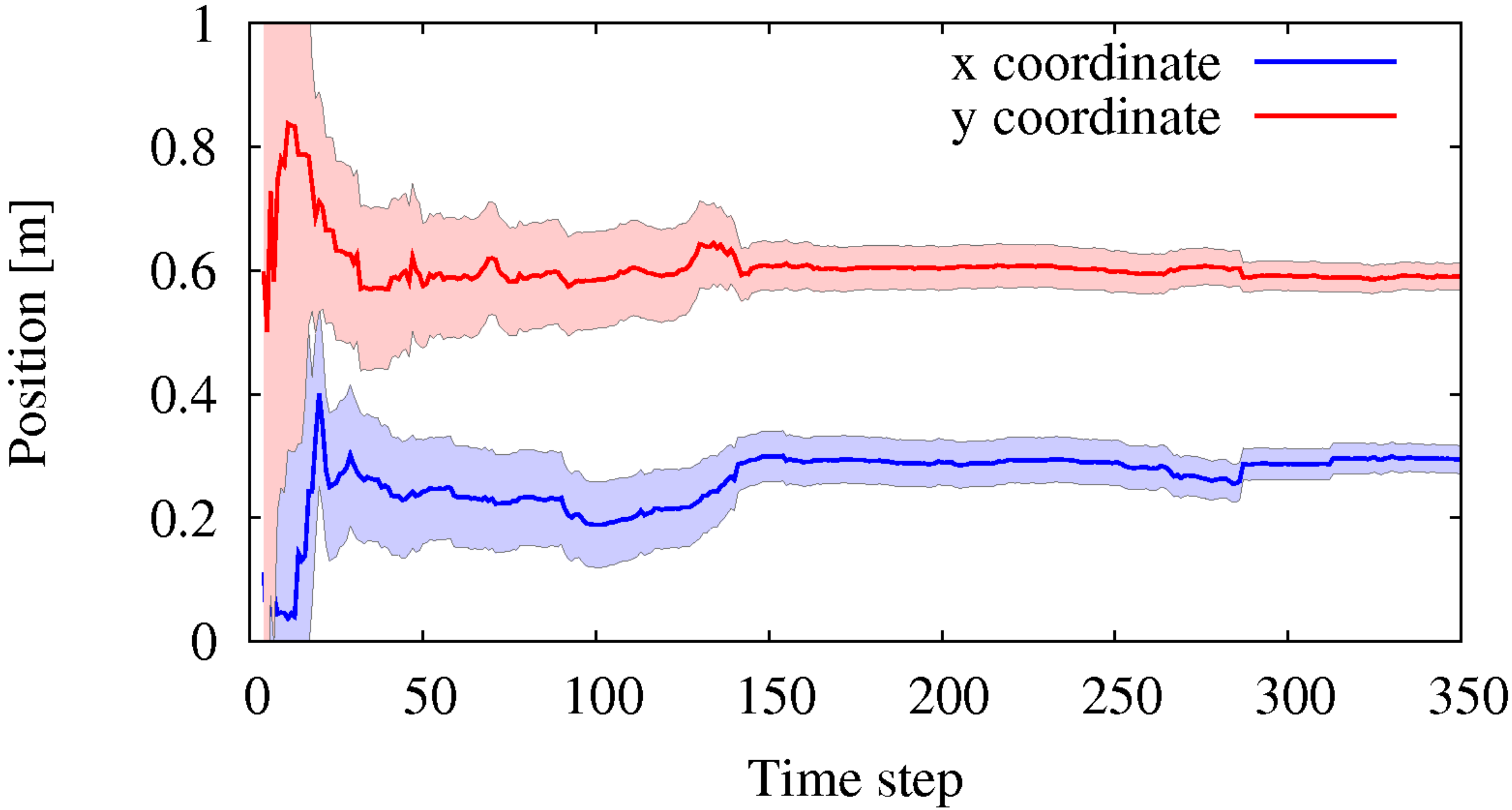
- Simulate a robot carrying a load
- Sliding windows for the wheel radii

Online Odometry Calibration

- Robot carries a load
- Additional weight compresses the tires
- Since the load is variable, the best performance can be obtained by estimating the parameters online



Position of the on-board sensor



Ground truth: (0.3, 0.6, 30°)

Offline Experiments – Real world

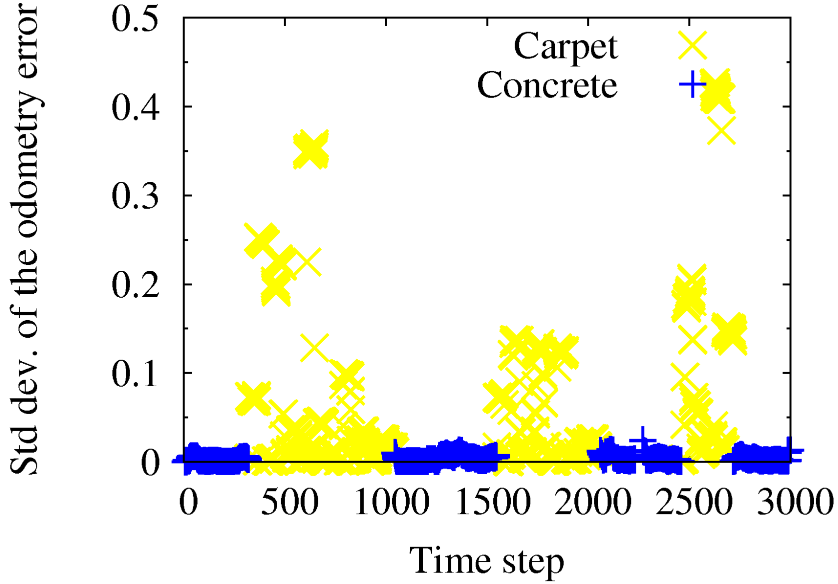
Robot parameters

	PowerBot	Custom	Pioneer
wheel radius [m]	0.125	0.16	0.065
wheel distance [m]	0.56	0.7	0.35
ticks per revolution	22835	20000	1970
laser offset [m, m, °]	(0.22, 0, 0)	(0.3, 0, 0)	(0.1, 0, 0)
laser scanner model	Sick LMS291	Sick LMS151	Hokuyo URG

Calibration results

	laser offset (m, m, °)	wheel radii (m, m)	distance m
PowerBot - 1	(0.2258, 0.0026, 0.099)	(0.1263, 0.1275)	0.5825
PowerBot - 2	(0.2231, -0.0031, 0.077)	(0.1243, 0.1248)	0.6091
Custom - 1	(0.3067, -0.0051, -0.357)	(0.1603, 0.1605)	0.6969
Custom - 2	(0.3023, -0.0087, -0.013)	(0.1584, 0.1575)	0.7109
Pioneer - 1	(0.1045, 0.009, -0.178)	(0.0656, 0.065)	0.3519
Pioneer - 2	(0.1066, -0.0031, -0.28)	(0.0658, 0.0655)	0.3461

Influence of the Underground



- Provides additional information about the noise induced by the floor

Summary

- Additional Parameters can be estimated during mapping
- Here: sensor offset and odometry parameters
- Parameter estimation can be easily integrated into the error minimization framework