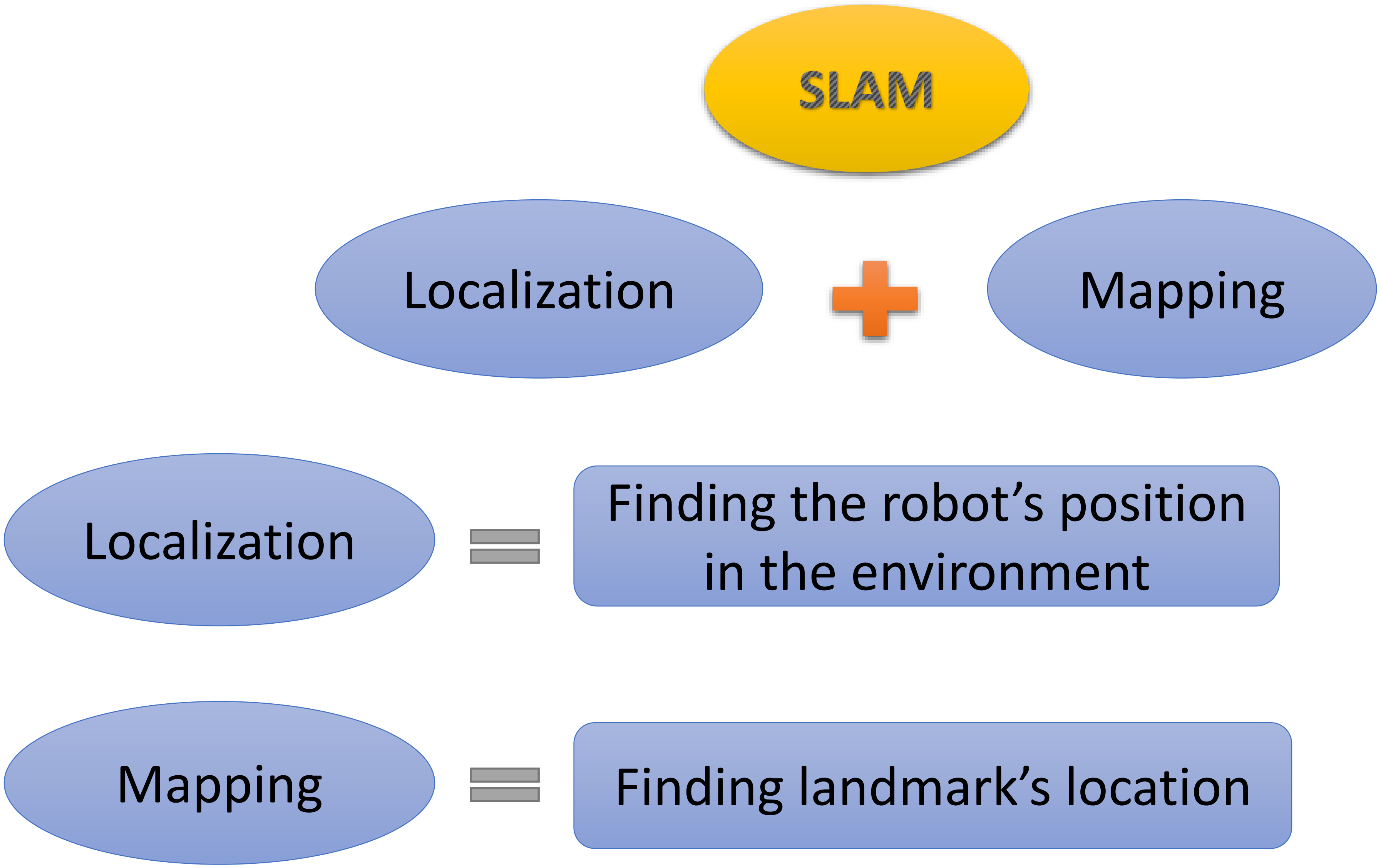
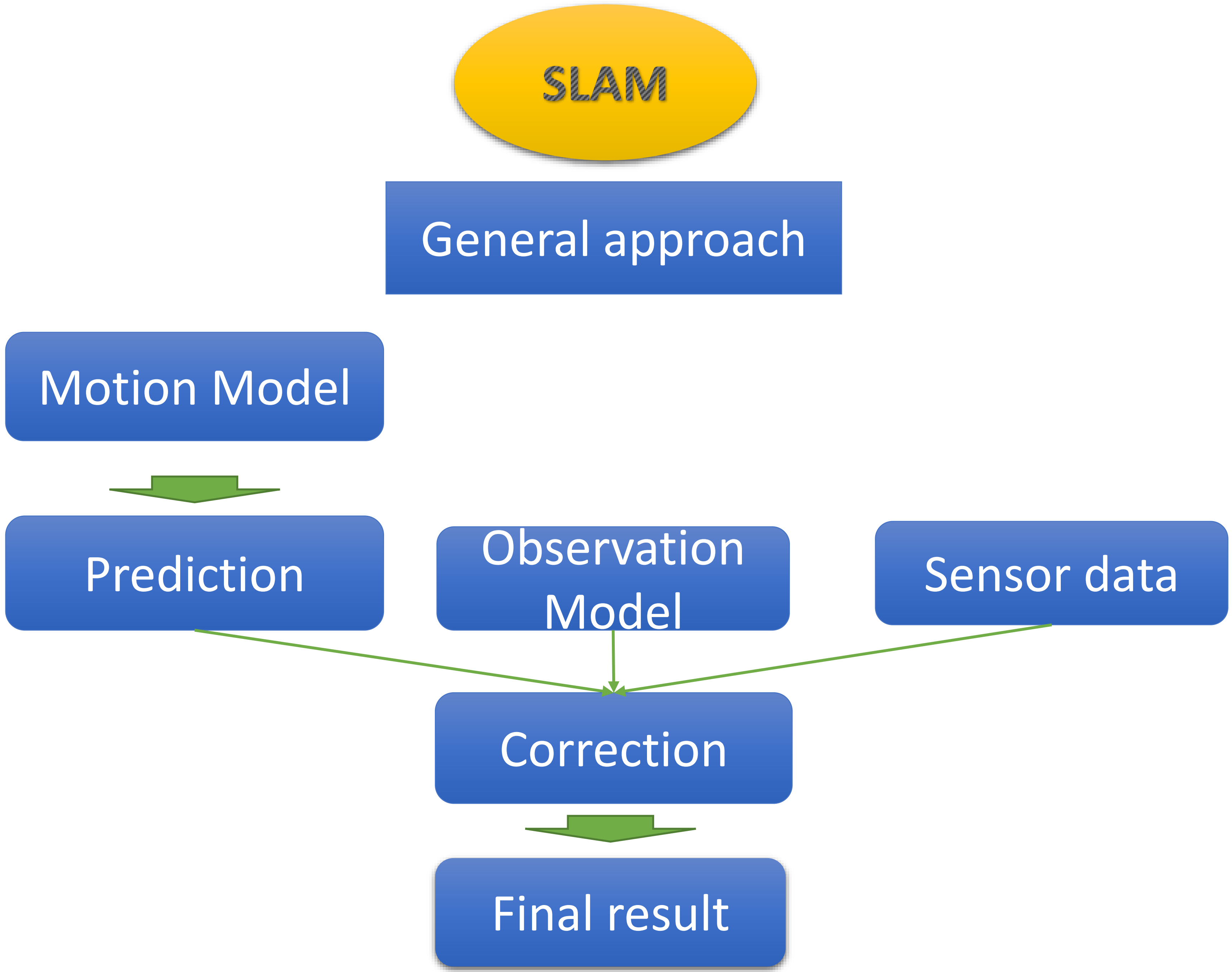
SLAM

By:
Hamed Ali Yaghini Bonabi
Supervisor: Prof. Khorrami


SLAM
Localization
Mapping
Localization
Finding the robot's position in the environment
Mapping
Finding landmark's location



SLAM
General approach
Motion Model
Prediction
Observation Model
Sensor data
Correction
Final result

Motion model

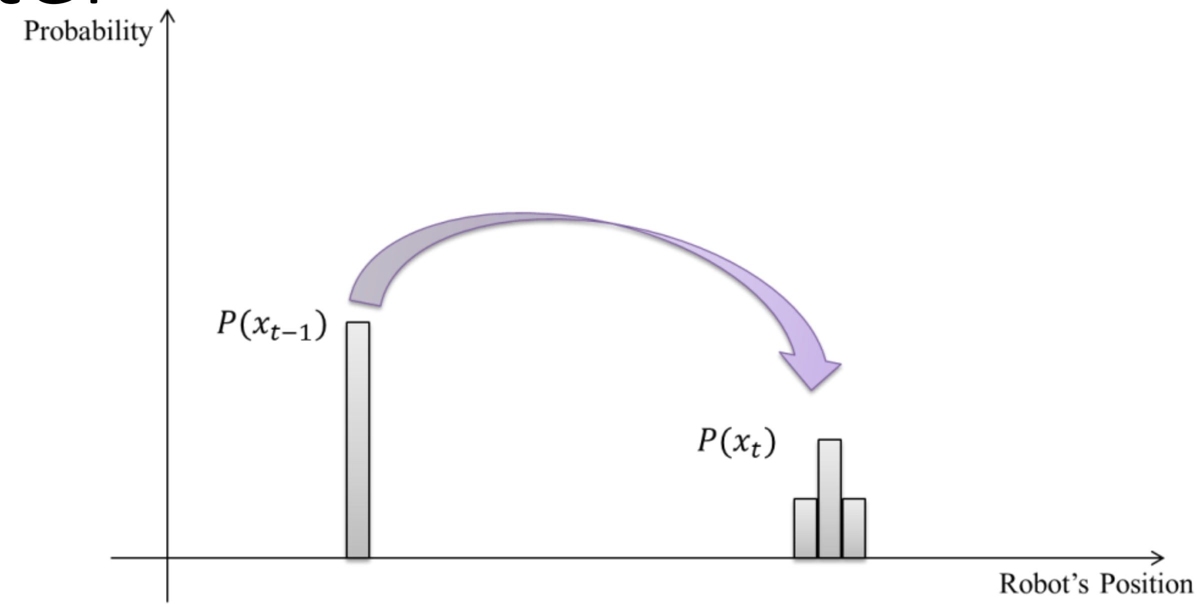
- Motion model is the relation between the previous state of the robot, control inputs and the robot's current state.

In general:

$$X_t = g(X_{t-1}, U_t)$$

where, X_t is the robot's current state. X_{t-1} is the robot's previous state, and U_t is the control input.

Bayes Filter



$$Bayes\ Filter(bel(x_{t-1}), u_t, z_t)$$
$$for\ all\ x_t:$$
$$\overline{bel}(x_t) = \sum_{x_{t-1}} P(x_t|x_{t-1}, u_t).\, bel(x_{t-1})$$
$$\overline{bel} = Convolve(move, bel)$$

- $\overline{bel}(x_t)$ is the probability that the robot is at position x at time instance t

Bayes Filter

- For reducing uncertainty and correcting robot’s position, measurement data are used in the correction step.

$$bel(x_t) = \underline{\alpha P}(z_t|x_t).\overline{bel}(x_t)$$
$$bel = mult(\overline{bel}, measurement)$$

where $bel(x_t)$ is the probability of being at position x at time instance t, after using the observation data, z.

Kalman Filter

$$x_t = a_t x_{t-1} + b_t u_t + \varepsilon_R$$
$$Z_t = C_t . \bar{x}_t + \varepsilon_Q$$

$$1D: P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}\frac{(x-\mu)^2}{\sigma^2}} = (2\pi\sigma^2)^{\frac{-1}{2}} e^{\frac{-1}{2}(x-\mu)\sigma^{-2}(x-\mu)}$$

$$Kalman\ Filter((\mu_{t-1}, \sigma_{t-1}^2), (u_t, \sigma_R^2), (Z, \sigma_Q^2))$$

Kalman Filter – prediction step

$$\overline{bel}(x_t)\begin{cases}\bar{\mu}_t = a_t.\mu_{t-1} + b_t.u_t \\ \bar{\sigma}_t^2 = a_t^2.\sigma_{t-1}^2 + \sigma_R^2\end{cases}$$

Kalman filter – correction step

$$K_t = \frac{C_t \bar{\sigma}_t^2}{C_t^2 \bar{\sigma}_t^2 + \sigma_Q^2}$$

$$bel(x_t) \begin{cases} \mu_t = \bar{\mu}_t + K_t(Z_t - C_t \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t C_t)\bar{\sigma}_t^2 \end{cases}$$

Kalman filter – n-dimensional state

$$nD: P(x) = \det(2\pi\Sigma)^{\frac{-1}{2}} . e^{\frac{-1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)}$$

$$x_t = A_t X_{t-1} + B_t U_t + \varepsilon_R$$

$$\overline{bel}(X_t) \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t \mu_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

Kalman filter – n-dimensional state

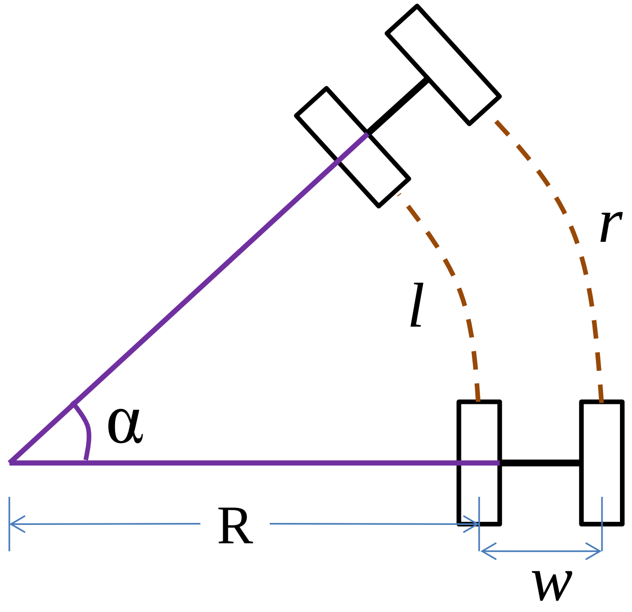
$$Z_t = C_t \bar{X}_t + \varepsilon_Q$$

$$K_t = \bar{\sum}_t C_t^T (C_t \bar{\sum}_t C_t^T + Q_t)^{-1}$$

$$bel(X_t) \begin{cases} \mu_t = \bar{\mu}_t + K_t (Z_t - C_t \bar{\mu}_t) \\ \sum_t = (I - K_t C_t) \bar{\sum}_t \end{cases}$$

Motion model

- An example: two wheeled robot:



$$\alpha = \frac{r-l}{w}$$

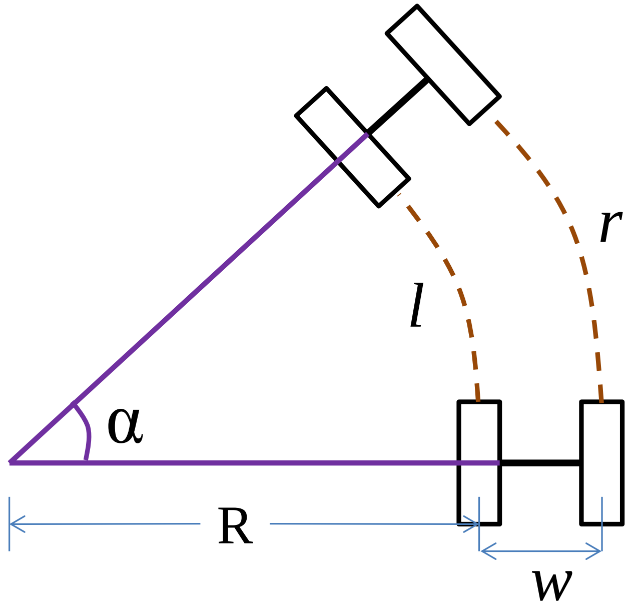
$$R = \frac{l}{\alpha}$$

$$if\ \alpha \neq 0\ \equiv r \neq l \rightarrow$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \left(R + \frac{w}{2}\right)(\sin(\theta + \alpha) - \sin(\theta)) \\ \left(R + \frac{w}{2}\right)(-\cos(\theta + \alpha) + \cos(\theta)) \\ \alpha \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = g(x, y, \theta, l, r)$$

Motion model

- An example: two wheeled robot:



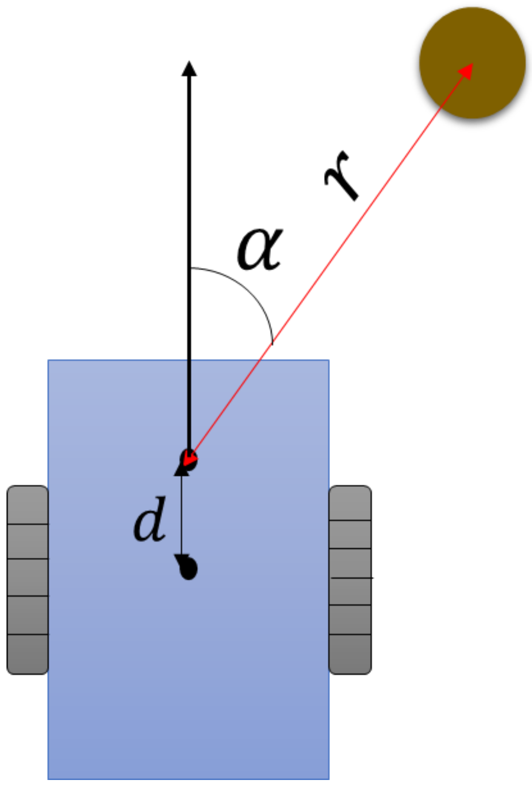
$$if\ r = l \rightarrow \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} l.\cos(\theta) \\ l.\sin(\theta) \\ 0 \end{bmatrix}$$

Observation model

- Observation model is the relation between the robot's position, location of the landmarks in the environment, and the measurement data.

- Measurement function examples for the two wheeled robot:

$$r = \sqrt{(x_m - x_l)^2 + (y_m - y_l)^2}$$
$$\alpha = \text{atan}\left(\frac{y_m - y_l}{x_m - x_l}\right) - \theta$$
$$x_l = x + d.\cos\theta$$
$$y_l = y + d.\sin\theta$$



Extended Kalman filter – Prediction step

- Our models are nonlinear ➔ nonlinear version of Kalman filter, extended Kalman filter, is used.

$$X_t = g(X_{t-1}, U_t)$$

$$X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} U = \begin{bmatrix} l \\ r \end{bmatrix}$$

$$\bar{\mu}_t = g(\mu_{t-1}, u_t)$$
$$\bar{\sum}_t = G_t \sum_{t-1} G_t^T + R_t$$

Extended Kalman filter – Prediction step

G_t is the Jacobian of g

$$G = \frac{\partial g}{\partial state}, state = [x, y, \theta]^T$$

$$G = \begin{bmatrix} \frac{\partial g_1}{\partial x} & \frac{\partial g_1}{\partial y} & \frac{\partial g_1}{\partial \theta} \\ \frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial y} & \frac{\partial g_2}{\partial \theta} \\ \frac{\partial g_3}{\partial x} & \frac{\partial g_3}{\partial y} & \frac{\partial g_3}{\partial \theta} \end{bmatrix}$$

Extended Kalman filter – Prediction step

$$G = \begin{bmatrix} 1 & 0 & (R+\frac{w}{2})(\cos(\theta+\alpha) - cos\theta) \\ 0 & 1 & (R+\frac{w}{2})(\sin(\theta+\alpha) - sin\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

$r = l$ → $\alpha = 0$, and $R = \infty$ → using L'Hôpital's rule →

$$G = \begin{bmatrix} 1 & 0 & -l.sin\theta \\ 0 & 1 & l.cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

Extended Kalman filter – Prediction step

$$R_t = V_t \Sigma_{control} V_t^T$$

$$V = \frac{\partial g}{\partial control}$$

- $\Sigma_{control}$ is the covariance matrix of the control signals, so R_t is:

$$R_t = V_t \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} V_t^T$$

Extended Kalman filter – Prediction step

$$V = \begin{bmatrix} \frac{\partial g_1}{\partial l} & \frac{\partial g_1}{\partial r} \\ \frac{\partial g_2}{\partial l} & \frac{\partial g_2}{\partial r} \\ \frac{\partial g_3}{\partial l} & \frac{\partial g_3}{\partial r} \end{bmatrix}$$

$$\theta' = \theta + \alpha$$

Extended Kalman filter – Prediction step

- $if\ r \neq l\ \rightarrow$

$$\frac{\partial g_1}{\partial l} = \frac{wr}{(r-l)^2}(sin\theta' - sin\theta) - \frac{r+l}{2(r-l)}cos\theta'$$
$$\frac{\partial g_2}{\partial l} = \frac{wr}{(r-l)^2}(-cos\theta' + cos\theta) - \frac{r+l}{2(r-l)}sin\theta'$$
$$\frac{\partial g_3}{\partial l} = -\frac{1}{w}$$
$$\frac{\partial g_1}{\partial r} = \frac{-wr}{(r-l)^2}(sin\theta' - sin\theta) + \frac{r+l}{2(r-l)}cos\theta'$$
$$\frac{\partial g_2}{\partial r} = \frac{-wr}{(r-l)^2}(-cos\theta' + cos\theta) + \frac{r+l}{2(r-l)}sin\theta'$$
$$\frac{\partial g_3}{\partial l} = \frac{1}{w}$$

Extended Kalman filter – Prediction step

- $if\ r = l\ \rightarrow$

$$\frac{\partial g_1}{\partial l} = \frac{1}{2}(cos\theta + \frac{l}{w}sin\theta)$$

$$\frac{\partial g_2}{\partial l} = \frac{1}{2}(sin\theta - \frac{l}{w}cos\theta)$$

$$\frac{\partial g_1}{\partial r} = \frac{1}{2}(-\frac{l}{w}sin\theta + cos\theta)$$

$$\frac{\partial g_2}{\partial r} = \frac{1}{2}(\frac{l}{w}cos\theta + sin\theta)$$

Extended Kalman filter – Prediction step

$$\Sigma_{Control} = \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$$

- σ_l^2 and σ_r^2 show the uncertainty in the control of left and right wheels. They can be defined as:

$$\sigma_l^2 = (\alpha_1 . l)^2 + \left(\alpha_2 (l - r)\right)^2$$
$$\sigma_r^2 = (\alpha_1 . r)^2 + \left(\alpha_2 (l - r)\right)^2$$

Extended Kalman filter – Correction step

measurement function:

$$Z_t = h(X_t)$$

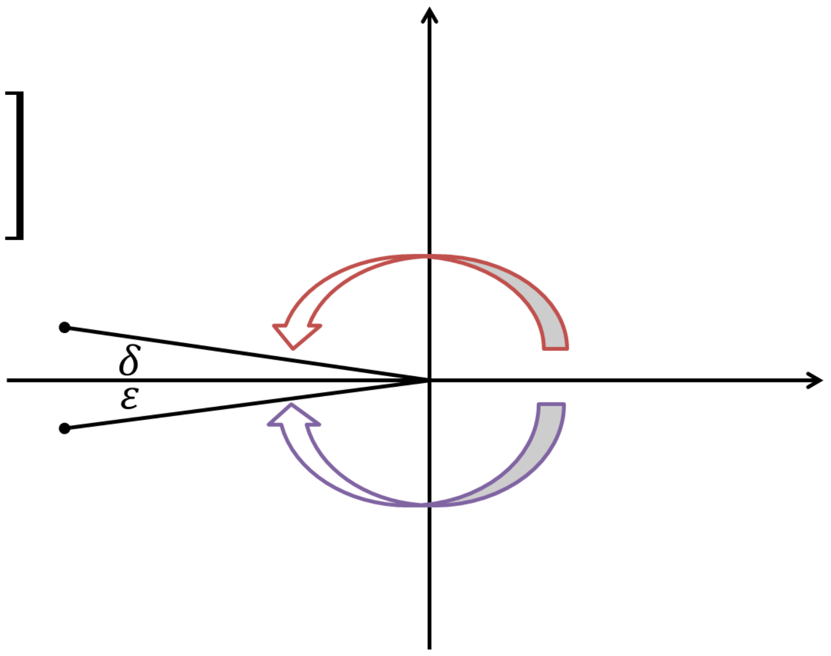
$$\begin{cases} \mu_t = \bar{\mu}_t + K_t(Z - h(\bar{\mu}_t)) \\ \sum_t = (I - K_t H_t)\bar{\sum}_t \end{cases}$$

$$H = Jacobian\ of\ h = \frac{\partial h}{\partial state}$$

Correction step hint

- for computing the new μ, in the correction step, we have to compute the difference between the measured and predicted values for range and angle, which is called the innovation.

$$Z - h(\bar{\mu}_t) = \begin{bmatrix} z_r - h_r(\bar{\mu}) \\ z_\alpha - h_\alpha(\bar{\mu}) \end{bmatrix}$$



Correction step hint

- Difference in the angle might lead to wrong result. The measurement tells us that:

$$z_\alpha = \pi - \delta$$

The prediction is:

$$h_\alpha = -\pi + \varepsilon$$

Subtraction of measurement and predicted values :

$$z_\alpha - h_\alpha = \pi - \delta - (-\pi + \varepsilon) = 2\pi - (\delta + \varepsilon)$$

which is far away from the true difference value:

$$-(\delta + \varepsilon)$$

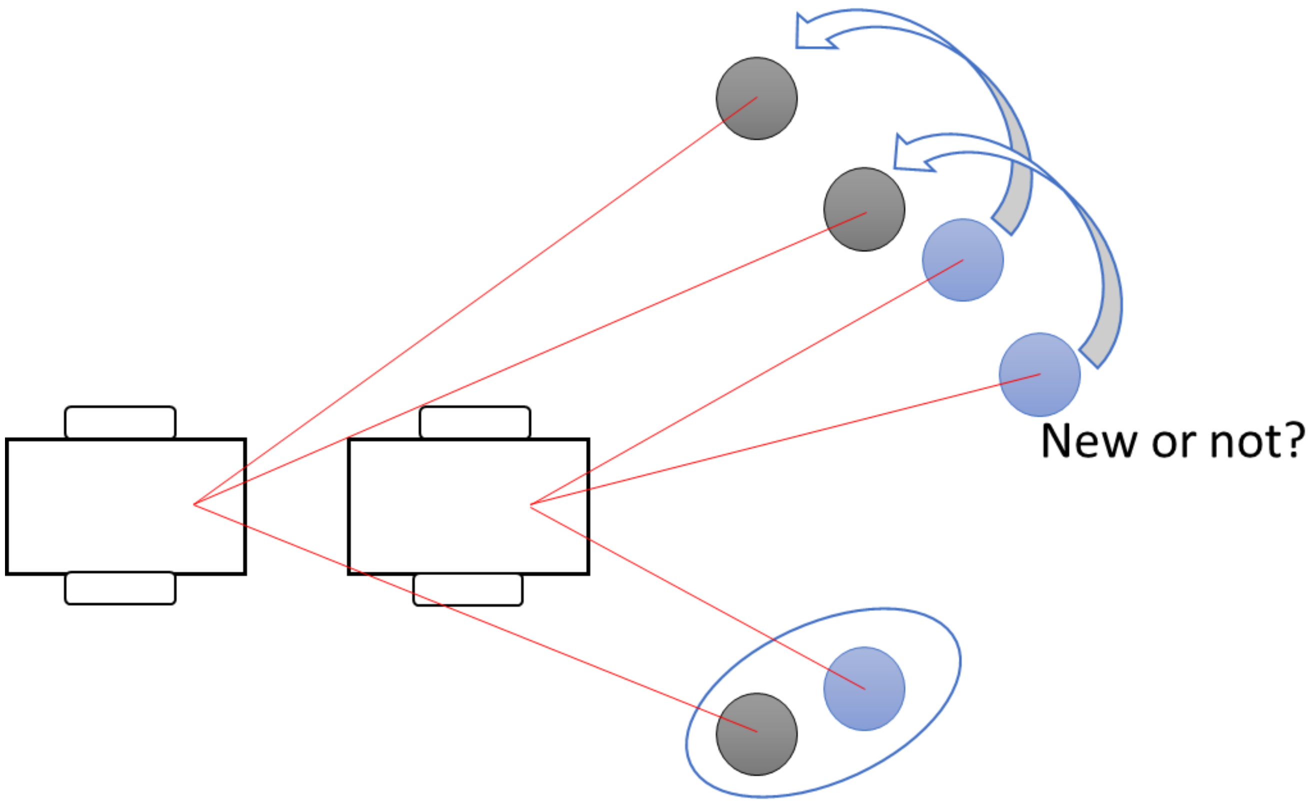
we must make sure that the final range of the angle difference is within $(-\pi, \pi)$

Simultaneous localization and mapping

- Put the world coordinates' origin in the robot's starting point.
- Start observing the environment
- While moving, add the landmarks as detected to the system's state, with uncertainty.
- Use prediction and correction methods of localization to finding the correct location of both robot and landmarks.

$$state = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \text{ add one landmark } \rightarrow \begin{bmatrix} x \\ y \\ \theta \\ x_1 \\ y_1 \end{bmatrix} \text{ add another landmark} \rightarrow \begin{bmatrix} x \\ y \\ \theta \\ x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} \rightarrow \cdots$$

Adding landmarks to model



Maximum likelihood landmark assignment

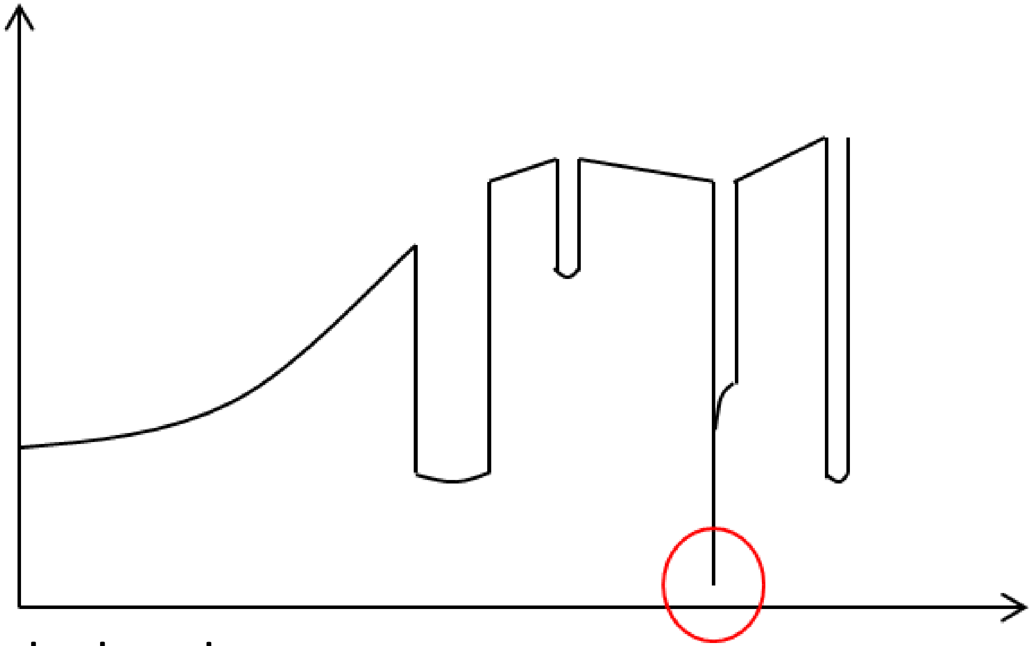
- Mahalanobis distance:

$$\big(z - h(x)\big)^T \psi^{-1} \big(z - h(x)\big) < \varepsilon$$

$$\psi = H\bar{\Sigma}H^T + Q$$

Detecting a landmark

- Dealing with noise



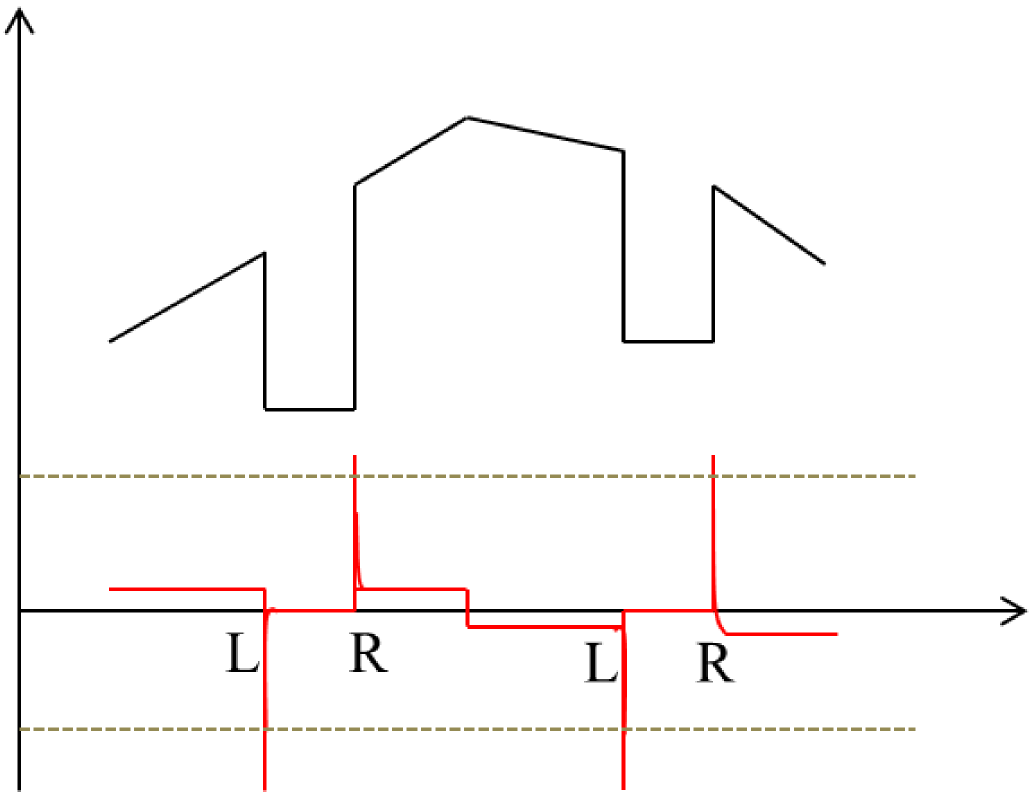
- derivative of the sensor data is calculated:

$$f'(i) \approx \frac{f(i+1) - f(i-1)}{2}$$

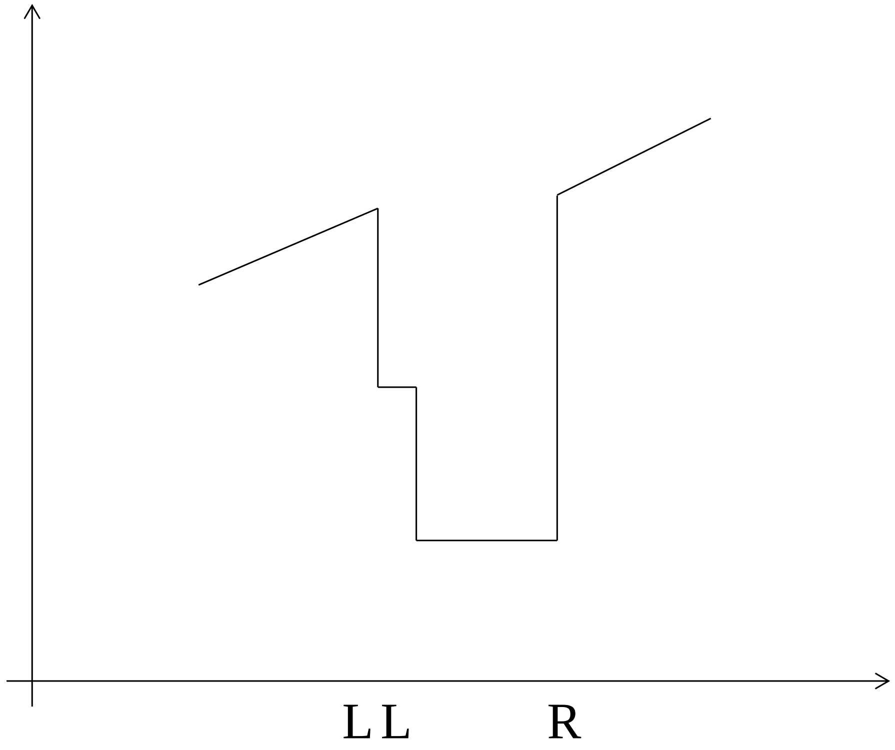
$f(i)$ is the distance measured by the ith beam.

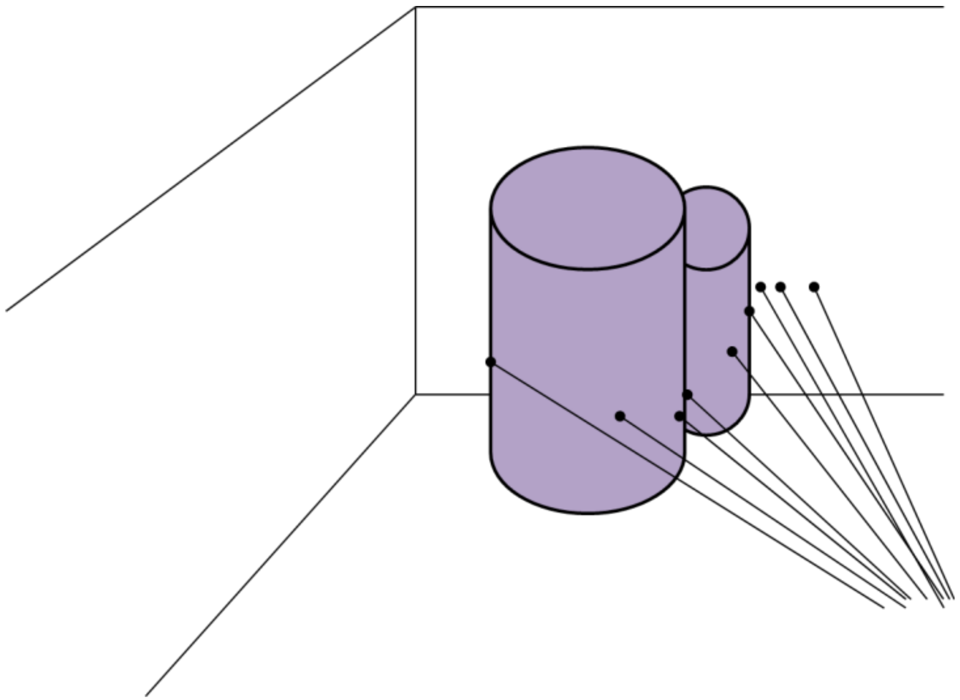
Detecting a landmark

- A threshold is defined, and when the derivative is larger than this threshold edge of a landmark is detected.
- Negative sharp values of derivative show start of a landmark, and positive ones show end of the landmark.



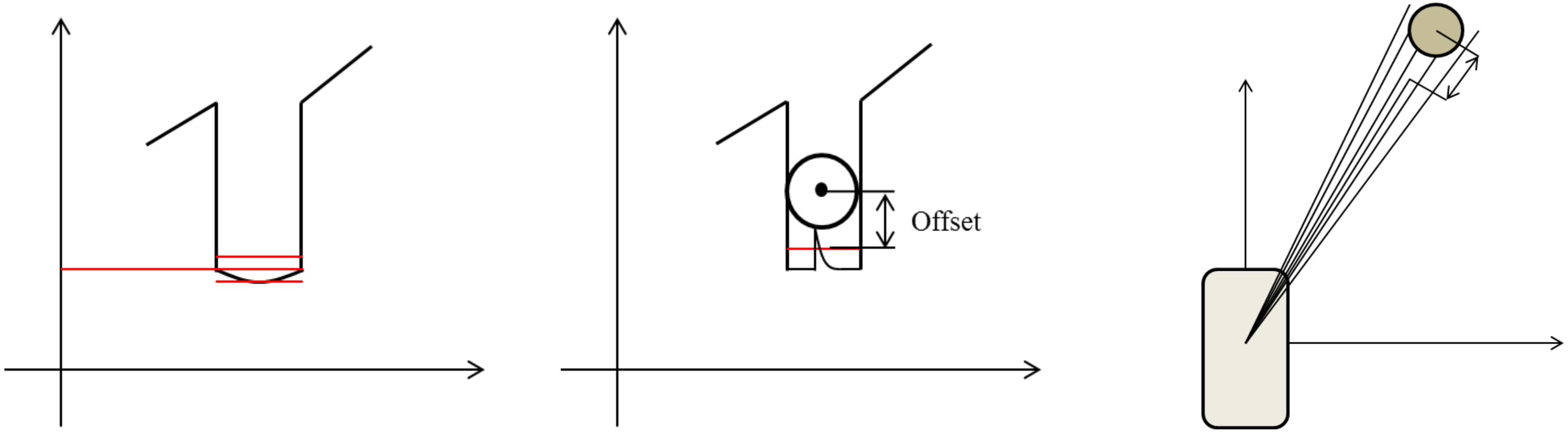
Unusual case in landmark detection





Practical issues

- In practice, there is an error between the average calculated value and the real distance from the landmark center. This offset values can be obtained empirically and must be added to the calculated value before using the observation information.



Semantic SLAM