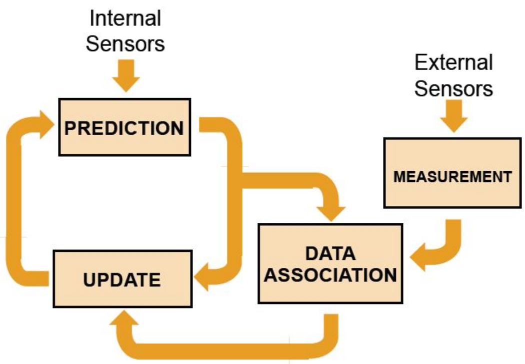
Pre-Lab5: EKF Simultaneous Localization and Mapping

Daniel Barmaimon
Probabilistic Robotics
Vibot Master Degree
University of Girona

April 30, 2014

1 Read and understand SLAM implementation

After reading the slam implementation, the process could be summarized with the two pictures that are shown in Fig.1



(a) Block representation of SLAM process

Algorithm 1 SLAM:

$\mathbf{x}_0^B = 0$; $\mathbf{P}_0^B = 0$ {*Map initialization*}
$[\mathbf{z}_0, \mathbf{R}_0]$ = get_measurements
$[\mathbf{x}_0^B, \mathbf{P}_0^B]$ = add_new_features($\mathbf{x}_0^B, \mathbf{P}_0^B, \mathbf{z}_0, \mathbf{R}_0$)
for $k = 1$ to steps **do**
  $[\mathbf{x}_{R_k}^{R_{k-1}}, \mathbf{Q}_k]$ = get_odometry
  $[\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B]$ = compute_motion($\mathbf{x}_{k-1}^B, \mathbf{P}_{k-1}^B, \mathbf{x}_{R_k}^{R_{k-1}}, \mathbf{Q}_k$) {*EKF prediction*}
  $[\mathbf{z}_k, \mathbf{R}_k]$ = get_measurements
  $\mathcal{H}_k$ = data_association($\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k$)
  $[\mathbf{x}_k^B, \mathbf{P}_k^B]$ = update_map($\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H}_k$) {*EKF update*}
  $[\mathbf{x}_k^B, \mathbf{P}_k^B]$ = add_new_features($\mathbf{x}_k^B, \mathbf{P}_k^B, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H}_k$)
end for

(b) Pseudo-code algorithm for SLAM

Figure 1: Summary for SLAM

2 Find the expression of F_k and G_k for the prediction

These matrices will be as follows:

$$F_k = J_{1\oplus}\left\{x_B^A, x_C^B\right\} = \left.\frac{\partial x_B^A \oplus x_C^B}{\partial x_B^A}\right|_{(\hat{x}_B^A, \hat{x}_C^B)} = \begin{bmatrix} 1 & 0 & -x_2 sin\phi_1 - y_2 cos\phi_1 \\ 0 & 1 & x_2 sin\phi_1 - y_2 cos\phi_1 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$G_k = J_{2\oplus}\left\{x_B^A, x_C^B\right\} = \left.\frac{\partial x_B^A \oplus x_C^B}{\partial x_C^B}\right|_{(\hat{x}_B^A, \hat{x}_C^B)} = \begin{bmatrix} cos\phi_1 & -sin\phi_1 & 0 \\ -sin\phi_1 & cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where the odometry arrays are,

$$x_B^A = \begin{bmatrix} x_1 \\ y_1 \\ \phi_1 \end{bmatrix} \qquad x_C^B = \begin{bmatrix} x_2 \\ y_2 \\ \phi_2 \end{bmatrix} \tag{3}$$

3 Given a feature in the robot frame zk with uncertainty Rk. Find the equation to increment the state vector with this feature

Knowing the uncertainty, it is possible to add a new feature if the last uncertainty is compounded with the measurement of the new feature. As it is shown in Fig.2

$$\mathbf{x}_k^B = \begin{pmatrix} \mathbf{x}_{R_k}^B \\ \mathbf{x}_{F_{1,k}}^B \\ \vdots \\ \mathbf{x}_{F_{n,k}}^B \end{pmatrix} \Rightarrow \mathbf{x}_{k+}^B = \begin{pmatrix} \mathbf{x}_{R_k}^B \\ \mathbf{x}_{F_{1,k}}^B \\ \vdots \\ \mathbf{x}_{F_{n,k}}^B \\ \mathbf{x}_{F_{n+1,k}}^B \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{R_k}^B \\ \mathbf{x}_{F_{1,k}}^B \\ \vdots \\ \mathbf{x}_{F_{n,k}}^B \\ \mathbf{x}_{R_k}^B \oplus \mathbf{z}_i \end{pmatrix}$$

Linearization:

$$\mathbf{x}_{k+}^B \simeq \hat{\mathbf{x}}_{k+}^B + \mathbf{F}_k(\mathbf{x}_k^B - \hat{\mathbf{x}}_k^B) + \mathbf{G}_k(\mathbf{z}_i - \hat{\mathbf{z}}_i)$$

$$\mathbf{P}_{k+}^B = \mathbf{F}_k \mathbf{P}_k^B \mathbf{F}_k^T + \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^T$$

Where:

$$\mathbf{F}_k = \frac{\partial \mathbf{x}_{k+}^B}{\partial \mathbf{x}_k^B} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \\ \mathbf{J}_{1\oplus}\left\{\hat{\mathbf{x}}_{R_k}^B, \hat{\mathbf{z}}_i\right\} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} ; \mathbf{G}_k = \frac{\partial \mathbf{x}_{k+}^B}{\partial \mathbf{z}_i} = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{J}_{2\oplus}\left\{\hat{\mathbf{x}}_{R_k}^B, \hat{\mathbf{z}}_i\right\} \end{pmatrix}$$

Figure 2: Adding a new feature to the state vector

It is much more convenient avoid the most of the calculations that involve moving through all the features. For that is needed the linearization, simplifying the problem as follows in Fig.3

$$\mathbf{P}_k^B = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} \\ \mathbf{P}_{RF_1}^T & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{RF_n}^T & \mathbf{P}_{F_1F_n}^T & \cdots & \mathbf{P}_{F_n} \end{pmatrix}$$

$$\mathbf{P}_{k+}^B = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} & \mathbf{P}_R \mathbf{J}_{1\oplus}^T \\ \mathbf{P}_{RF_1}^T & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1F_n} & \mathbf{P}_{RF_1}^T \mathbf{J}_{1\oplus}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{P}_{RF_n}^T & \mathbf{P}_{F_1F_n}^T & \cdots & \mathbf{P}_{F_n} & \mathbf{P}_{RF_n}^T \mathbf{J}_{1\oplus}^T \\ \mathbf{J}_{1\oplus}\mathbf{P}_R & \mathbf{J}_{1\oplus}\mathbf{P}_{RF_1} & \cdots & \mathbf{J}_{1\oplus}\mathbf{P}_{RF_n} & \mathbf{J}_{1\oplus}\mathbf{P}_R\mathbf{J}_{1\oplus}^T + \mathbf{J}_{2\oplus}\mathbf{R}_k\mathbf{J}_{2\oplus}^T \end{pmatrix}$$

Figure 3: Adding a new feature to the state vector after linearization

In the case that the robot get the measurements in polar coordinates (robot frame), compounding would be implemented by transforming to cartesian coordinates (world frame) and adding. That's the reason for using the function *self.tfPolarLine(tf,feature)*

4 Find the size of the matrices H, S, v and R in order to be able to update the filter with n features observations without using a for loop.

$$S_k = \underbrace{\underbrace{H_k}_{\text{2n x 3}} \; \underbrace{P_{k|k-1}}_{\text{3 x 3}} \; \underbrace{H_k^T}_{\text{3 x 2n}}}_{\text{2n x 2n}} + \underbrace{R_k}_{\text{2n x 2n}} \tag{4}$$

$$K_k = \underbrace{\underbrace{P_{k|k-1}}_{\text{3 x 3}} \; \underbrace{H_k^T}_{\text{3 x 2n}} \; \underbrace{S_k^{-1}}_{\text{2n x 2n}}}_{\text{3 x 2n}} \tag{5}$$

$$x_k^w = \underbrace{\underbrace{x_{k|k-1}^w}_{\text{3 x 1}} \; \underbrace{K_k}_{\text{3 x 2n}} \; \underbrace{V_k}_{\text{2n x 1}}}_{\text{3 x 1}} \tag{6}$$

$$P_k = \underbrace{(\underbrace{I}_{3 \text{ x } 3} - \underbrace{K_k}_{3 \text{ x } 2\text{n}} \underbrace{H_k}_{2\text{n x } 3})}_{3 \text{ x } 3} \underbrace{P_{k|k-1}}_{3 \text{ x } 3} \underbrace{(\underbrace{I}_{3 \text{ x } 3} - \underbrace{\underbrace{K_k}_{3 \text{ x } 2\text{n}} \underbrace{H_k}_{2\text{n x } 3}}_{3 \text{ x } 3})^T + \underbrace{K_k}_{3 \text{ x } 2\text{n}} \underbrace{R_k}_{2\text{n x } 2\text{n}} \underbrace{K_k^T}_{2\text{n x } 3}}_{3 \text{ x } 3} \tag{7}$$