Kalman Filter Used in Mobile Robots

Jesus Savage[1], Diego Cordero[1], Marco Negrete[1], Luis Contreras[2], Hiroyuki Okada[2], Roberto Valenti[3] and Jose Avendaño[3]

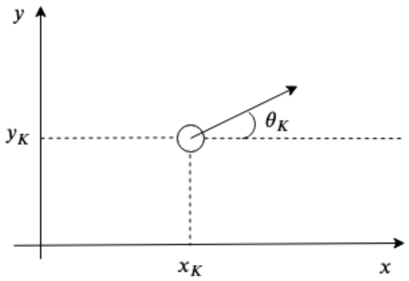
[1] Bio-Robotics Laboratory, School of Engineering, National Autonomous University of Mexico, UNAM.
[2] Advance Intelligence and Robotics Research Center, Tamagawa University, Japan.
[3] Mathworks Research Center, USA.

Outline

Introduction

In this lecture we will find out how to localize a mobile robot using the Kalman filter. The robot is originally found with the pose $\underline{x}(k)$ as shown in the following figure:

$$\underline{x}(k) = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}$$

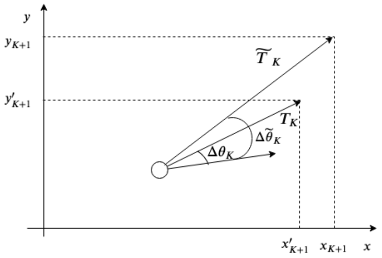


Introduction

The robot receives a forward and turn command:

$$\underline{u}(k) = \begin{bmatrix} T_k \\ \Delta\theta_k \end{bmatrix} = \begin{bmatrix} \textit{Advance command} \\ \textit{Turn command} \end{bmatrix}$$

When moving forward and turning small errors are made, in reality $\tilde{T}_k$ was advanced and $\tilde{\Delta}\theta_k$ was turned, as it is shown in the next figure:

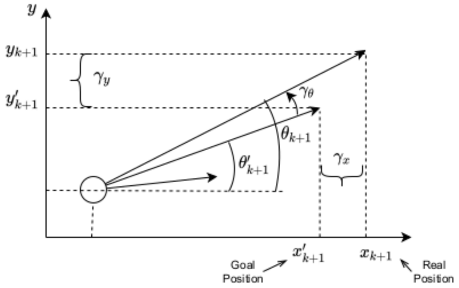


Introduction

The desired pose is given by $\underline{x}'(k+1)$; the actual pose is $\underline{x}(k+1)$

$$\underline{x}'(k+1) = \begin{bmatrix} x'_{k+1} \\ y'_{k+1} \\ \theta'_{k+1} \end{bmatrix}$$

$$\underline{x}(k+1) = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix}$$

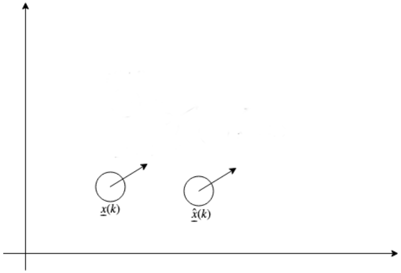


Introduction

As the robot advances these errors accumulate and therefore the true position of the robot is unknown. Thus, it is necessary to predict the pose of the robot.
The robot is at $\underline{x}(k)$ and its prediction is at $\hat{\underline{x}}(k)$, as shown in the following figure:

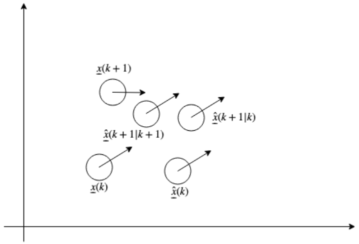
$$\underline{x}(k) = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}$$

$$\hat{\underline{x}}(k) = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{bmatrix}$$



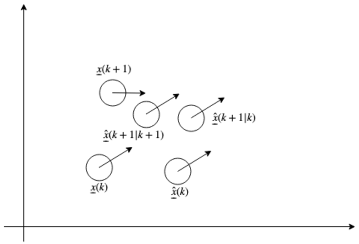
Introduction

The robot moves to $\underline{x}(k+1)$ and its movement is predicted with $\hat{\underline{x}}(k+1|k)$ using the motion model. A simulated reading is made with the map in this position and compared with a real reading with the current position of the robot, generating an error. With this, the estimate of the new position $\hat{\underline{x}}(k+1|k+1)$ is calculated.

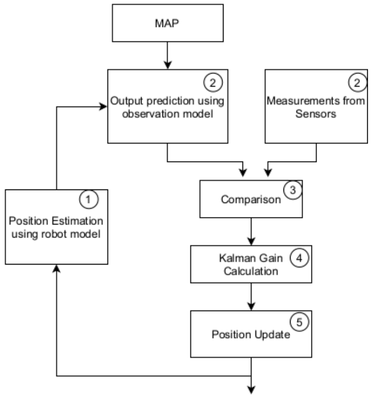


The basic localization cycle

Given an estimate of the position $\underline{\hat{x}}(k|k)$ and its covariance (uncertainty of this position) $P(k|k)$ for time k, a move command $\underline{u}(k)$, a set of observations $\underline{z}(k+1)$ and a map $M(k)$, representing the environment where the robot operates, is calculated the new prediction position of the robot $\hat{x}(k+1|k+1)$ and its covariance (uncertainty of this position) $P(k+1|k+1)$.



The basic localization cycle


MAP
2
Output prediction using observation model
2
Measurements from Sensors
3
Comparison
1
Position Estimation using robot model
4
Kalman Gain Calculation
5
Position Update

Robot's Model

$$\underline{u}(k) = \begin{bmatrix} T_k \\ \Delta\theta_k \end{bmatrix} = \begin{bmatrix} \textit{Advance command} \\ \textit{Turn command} \end{bmatrix}$$

New robot's position

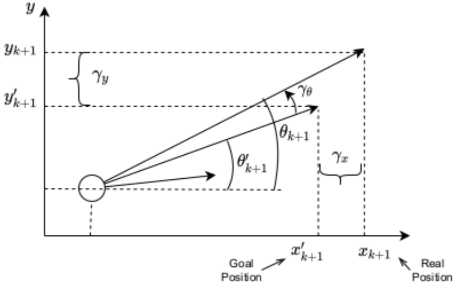
$$\underbrace{\underline{x}(k+1)} = f(\underline{x}(k),\ \underline{u}(k)) + \underline{\gamma}(k)$$

Where $f()$ is a robot motion function and $\underline{\gamma}(k)$ is a Gaussian noise vector $\backsim N(0, Q(k))$.

Robot's Model

The noise vector $\underline{\gamma}(k)$ has the following components, as shown in the following figure:

$$\underline{\gamma}(k) = \begin{bmatrix} \gamma_x(k) \\ \gamma_y(k) \\ \gamma_\theta(k) \end{bmatrix}$$



Robot's Model

$\underline{\gamma}(k)$ has the following covariance $Q(k)$

$$Q(k) = \begin{bmatrix} E\{\gamma_{x^2}(k)\} \; E\{\gamma_x(k)\,\gamma_y(k)\} \; E\{\gamma_x(k)\,\gamma_\theta(k)\} \\ E\{\gamma_y(k)\,\gamma_x(k)\} \; E\{\gamma_{y^2}(k)\} \; E\{\gamma_y(k)\,\gamma_\theta(k)\} \\ E\{\gamma_\theta(k)\,\gamma_x(k)\} \; E\{\gamma_\theta(k)\,\gamma_y(k)\} \; E\{\gamma_\theta^2(k)\} \end{bmatrix}$$

γ_x, γ_y and γ_θ are assumed to be uncorrelated and have mean 0, therefore:

$$Q(k) = \begin{bmatrix} \sigma^2_{\gamma_x}(k) & 0 & 0 \\ 0 & \sigma^2_{\gamma_y}(k) & 0 \\ 0 & 0 & \sigma^2_{\gamma_\theta}(k) \end{bmatrix}$$

Robot's Model

The robot's movement function has as parameters the robot pose $\underline{x}(k)$ and the command $\underline{u}(k)$:

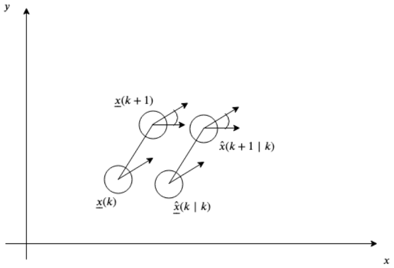
$$f\left(\underline{x}(k), \underline{u}(k)\right) = \begin{bmatrix} X_k + T_k \; cos \; \theta_k \\ Y_k + T_k \; sen \; \theta_k \\ \theta_k + \Delta\theta \end{bmatrix}$$

Note: here the robot is considered that it first moves forward and then it turns.
Using a simulator, the position of the robot is predicted having given the advance and turn command in time k, taking into account the previous prediction position $\underline{\hat{x}}(k|k)$:

$$\underline{\hat{x}}(k+1|k) = f\left(\underline{\hat{x}}(k|k)\;,\; \underline{u}(k)\right)$$

Robot's Model

The following figure shows the real position of the robot $\underline{x}(k+1)$ and its prediction $\hat{\underline{x}}(k+1|k)$:



There is an uncertainty of this position, this can be measured using the covariance:

$$P(k+1|k) = E[(\underline{x}(k+1) - \hat{\underline{x}}(k+1|k))\,(\underline{x}(k+1) - \hat{\underline{x}}(k+1|k))^T]$$

Robot's Model

$$P(k+1|k) = E[(\underline{x}(k+1) - \hat{\underline{x}}(k+1|k))\,(\underline{x}(k+1) - \hat{\underline{x}}(k+1|k))^T]$$

Here it is assumed that $\underline{x}(k+1)$ is known, which is actually unknown. The Kalman filter does not work well for nonlinear $f(\ )$ functions, like the movement function shown before, and when motion and sense noise do not correspond to random variables of the Gaussian type. In this case, the Extended Kalman filter is then used, in which the function $f(\ )$ is linearized by means of its Jacobian $\nabla f(\ )$.

Robot's Model

$$\nabla f = \left[\frac{\partial}{\partial \underline{x}} \, f\left(x(k)\,,\; u(k)\right) \right] = \begin{bmatrix} \frac{\partial \, f_1}{\partial x} & \frac{\partial \, f_1}{\partial y} & \frac{\partial \, f_1}{\partial \theta} \\ \frac{\partial \, f_2}{\partial x} & \frac{\partial \, f_2}{\partial y} & \frac{\partial \, f_2}{\partial \theta} \\ \frac{\partial \, f_3}{\partial x} & \frac{\partial \, f_3}{\partial y} & \frac{\partial \, f_3}{\partial \theta} \end{bmatrix}$$

With:

$$f\left(\underline{x}(k), \underline{u}(k)\right) = \begin{bmatrix} X_k + T_k \; cos \; \theta_k \\ Y_k + T_k \; sen \; \theta_k \\ \theta_k + \Delta\theta \end{bmatrix}$$

the Jacobian of the function f is:

$$\nabla f = \begin{bmatrix} 1 & 0 & -T_k \; sen \; \theta_k \\ 0 & 1 & T_k \; cos \; \theta_k \\ 0 & 0 & 1 \end{bmatrix}$$

Robot's Model

Therefore the covariance (position uncertainty) $P(k+1|k)$ is calculated as follows using the $P(k|k)$ above.

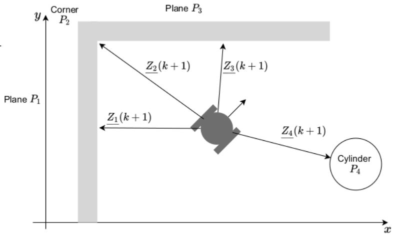
$$P(k+1|k) = \nabla f P(k|k) \nabla f^T + Q(k)$$

Measurement Model

While the robot is in a new position, a reading is made with its sensors to determine: 1) the position where the robot is; 2) find the position where there are known landmarks in the environment.

$$\underline{z}(k+1) = \{\underline{z}_j(k+1) \quad ; \quad 1 \leq j \leq N_M\}$$

Where N_M is the number of known landmarks.

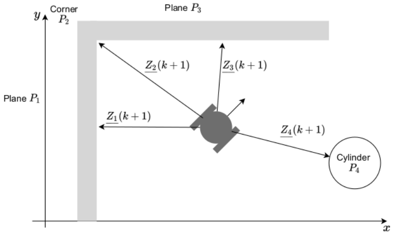


Measurement Model

1) Using the sensor readings the position of the robot is found. In this case, each $\underline{z}_j(k+1)$ represents the position of the robot from the sensor readings:

$$\underline{z}_j(k+1) = \; h_{map}\left(\underline{x}(k+1),\; M_j\right) + \underline{\omega}_j(k+1)$$

$$\underline{z}_j(k{+}1) = \begin{bmatrix} x_j \\ y_j \\ \theta_j \end{bmatrix} + \underline{\omega}_j(k{+}1)$$



Measurement Model

Where h_{map} is a function that relates the sensors reading to a known reference (landmark) and the position of the robot.
M_j are the known references in the map h_{map}.
$\underline{\omega}_j(k+1)$ represents the noise added to the position calculated by the sensors, this is a random variable, $N(0\ ,\ R_j(k+1))$.
With covariance matrix:

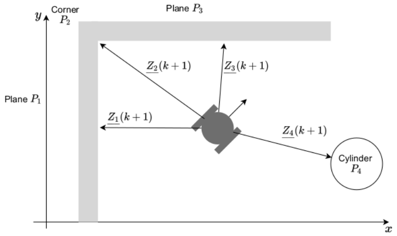
$$R_j(k+1) = \begin{bmatrix} \sigma_{\omega_x}^2 & 0 & 0 \\ 0 & \sigma_{\omega_y}^2 & 0 \\ 0 & 0 & \sigma_{\omega_\theta}^2 \end{bmatrix}$$

Measurement Model

2) While the robot is in its new position, readings are made with its sensors to determine the position of known landmarks in the environment.

$$\underline{z}(k+1) = \{\underline{z}_j(k+1) \quad ; \quad 1 \leq j \leq N_M\}$$

Where N_M is the number of known landmarks.



Measurement Model

In this case, each $\underline{z}_j(k+1)$ represents the distance $r_j(k+1)$ and the angle $\alpha_j(k+1)$ of a mark known data obtained from sensor readings:

$$\underline{z}_j(k+1) = \; h_{map}\left(\underline{x}(k+1),\; M_j\right) + \underline{\omega}_j(k+1)$$

$$\underline{z}_j(k+1) = \begin{bmatrix} r_j(k+1) \\ \alpha_j(k+1) \end{bmatrix} + \underline{\omega}_j(k+1)$$

Where h_{map} is a function that relates the reading of the sensors with a known reference (landmark), obtaining the distance $r_j(k+1)$ and the angle $\alpha_j(k+1)$ of this.
M_j are the known references in the map h_{map}.

Measurement Prediction

Where $\underline{\omega}_j(k+1)$ represents the noise added to the distance r and angle α, this is a random variable, $N(0\ ,\ R_j(k+1))$.

$$R_j(k+1) = \begin{bmatrix} \sigma^2_{\omega_d} & 0 \\ 0 & \sigma^2_{\omega_\alpha} \end{bmatrix}$$

Measurement Prediction

With the sensor simulator it is predicted, either, the estimated position of the robot or the distances and angles of the known environmental landmarks, from simulating the readings of the new estimated position of the robot $\hat{x}(k+1|k)$ after simulating its movement from position $\hat{x}(k)$.

$$\underline{\hat{z}}_i(k+1) = \ h_{map}\left(\underline{\hat{x}}(k+1|k)\right),\ M_i)$$

These predictions are made for all known landmarks on the map:

$$\{\underline{\hat{z}}_i(k+1)\};\ 1 \leq i \leq N_M$$

Comparison

The objective of the comparison is to have a measure of what is predicted with what is observed:

$$\underline{v}_{ij}(k+1) = \left[\underline{z}_j\ (k+1)\ -\ \underline{\hat{z}}_i(k+1)\right]$$

$$= \left[\underline{z}_j\ (k+1)\ -\ h_{map}\ (\underline{\hat{x}}_i(k+1|k),\ M_i)\right]$$

With the $\underline{v}_{ij}$ the vector $\underline{\mathbf{v}}(k+1)$ formed with each of the elements $\underline{v}_{ij}$ is formed:

$$\underline{\mathbf{v}}(k+1) = \{\underline{v}_{ij}\}$$

Comparison

$$\underline{\mathbf{v}}(k+1) = \{\underline{v}_{ij}\}$$

$$\underline{\mathbf{v}}(k+1) = \begin{bmatrix} \underline{v}_{11} \\ \underline{v}_{12} \\ \cdots \\ \underline{v}_{ij} \\ \cdots \\ \underline{v}_{NmNm} \end{bmatrix}$$

The uncertainty of this error or covariance is:

$$S_{ij}(k+1) = E\left[\underline{v}_{ij}(k+1)\underline{v}_{ij}^{T}(k+1)\right]$$

Comparison

$$S_{ij}(k+1) = E\left[\underline{v}_{ij}(k+1)\underline{v}_{ij}^T(k+1)\right]$$

This covariance can be found with the Jacobian of the function $\underline{h}_i = h_{map}\left(\hat{\underline{x}}(k+1|k),\ P_i\right)$ and the covariance $P(k+1|k)$.
Thus:

$$S_{ij} = \nabla \underline{h}_i P(k+1|k) \nabla \underline{h}_i^T + R_i(k+1)$$

where $R_i(k+1)$ is the variance of the noise $\underline{w}_i(k+1)$

Position Estimation Using the Kalman Filter

Then, to estimate the next position of the robot, the following expression is proposed:

$$\underline{\hat{x}}(k+1|k+1) = \underline{\hat{x}}(k+1|k) + \mathbf{K}(k+1)\underline{\mathbf{v}}(k+1)$$

where:
$\underline{\hat{x}}(k+1|k)$ = Estimation of the position of the robot given the movement command.
$\mathbf{K}(k+1)$ = Gain matrix of the Kalman filter
$\underline{\mathbf{v}}(k+1)$ = Error between position estimate from sensor readings and simulated readings.

Kalman gain

In the Kalman filter the objective is to find a matrix $\mathbf{K}(k+1)$ which minimizes the mean square error between the present robot's position and predicted position.
The prediction error is given by:

$$P(k+1|k+1) = E\{(\underline{x}(k+1) - \hat{\underline{x}}(k+1|k+1))(\underline{x}(k+1) - \hat{\underline{x}}(k+1|k+1))^T\}$$

$$P(k+1|k+1) = E\{(\underline{x}(k+1) - (\hat{\underline{x}}(k+1|k) + \mathbf{K}(k+1)\underline{\mathbf{v}}(k+1)))$$
$$(\underline{x}(k+1) - (\hat{\underline{x}}(k+1|k) + \mathbf{K}(k+1)\underline{\mathbf{v}}(k+1)))^T\}$$

Kalman gain

To find the Kalman gain matrix $\mathbf{K}(k+1)$ it is necessary to minimize $P(k+1|k+1)$ with respect to this matrix. Since the true pose of the robot $\underline{x}(k+1)$ is not known, the minimization gives the following result:

$$\mathbf{K}(k+1) = P(k+1|k)\nabla h^T \cdot \left[\nabla h P(k+1|k)\nabla h^T + R(k+1)\right]^{-1}$$

And the covariance associated with this position is:

$$P(k+1|k+1) = P(k+1|k) + \mathbf{K}(k+1)S(k+1)\mathbf{K}^T(k+1)$$

Example: Motion Function

There is a mobile robot with the following motion function:

$$f\left(\underline{x}(k), \underline{u}(k)\right) = \begin{bmatrix} X_k + T(k)\ cos\ \theta_k \\ Y_k + T(k)\ sen\ \theta_k \\ \theta_k + \Delta\theta \end{bmatrix}$$

Its Jacobian:

$$\nabla f = \begin{bmatrix} 1 & 0 & -T(k)\ sen\ \theta_k \\ 0 & 1 & T(k)\ cos\ \theta_k \\ 0 & 0 & 1 \end{bmatrix}$$

Example: Motion Function

With the command $\underline{u}(k)$ the new position of the robot is:

$$\underbrace{\underline{x}(k+1)} = f(\underline{x}(k),\ \underline{u}(k)) + \underline{\gamma}(k)$$

where $\underline{\gamma}(k)$ is the noise that is added when the move is made. The values of the covariance matrix of this noise, $Q(k)$, are found experimentally by externally measuring the real position of the robot by finding the aggregated noise $\underline{\gamma}(k)$.

$$Q(k) = \begin{bmatrix} \sigma^2_{\gamma_x}(k) & 0 & 0 \\ 0 & \sigma^2_{\gamma_y}(k) & 0 \\ 0 & 0 & \sigma^2_{\gamma_\theta}(k) \end{bmatrix}$$

Example: Position Estimate

With the same movement command $\underline{u}(k)$ the estimate of this position is calculated:

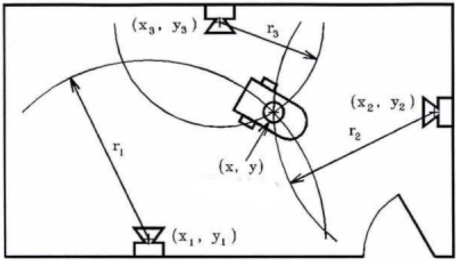
$$\hat{\underline{x}}(k+1|k) = f\left(\hat{\underline{x}}(k|k), \underline{u}(k)\right)$$

The covariance $P(k+1|k)$ is:

$$P(k+1|k) = \nabla f P(k|k) \nabla f^T + Q(k)$$

Measurement Model

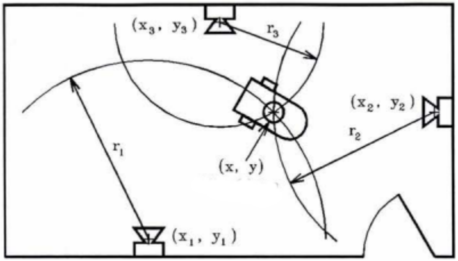
First we will see the case where the sensor readings are used to find the position of the robot. There is a system that finds landmarks in the environment, finding the distance r at which they are with respect to the position of the robot. The distances r_i are used to form three circles centered on the position of the landmarks (x_i, y_i) and finding their intersection the position of the robot (x, y) can be found.



Example: Sensor Reading

Then, the equations of each of the circles can be expressed as:

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2$$
$$(x - x_2)^2 + (y - y_2)^2 = r_2^2$$
$$(x - x_3)^2 + (y - y_3)^2 = r_3^2$$



Example: Sensor Reading

After developing the binomials, and subtracting them, we obtain:

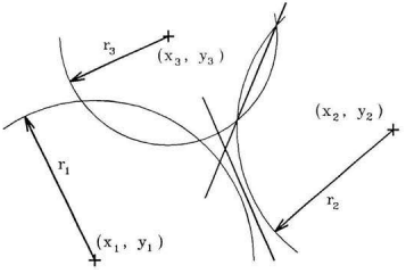
$$2x(x_2 - x_1) + x_1^2 - x_2^2 + 2y(y_2 - y_1) + y_1^2 - y_2^2 = r_1^2 - r_2^2$$

$$2x(x_2 - x_3) + x_3^2 - x_2^2 + 2y(y_2 - y_3) + y_3^2 - y_2^2 = r_3^2 - r_2^2$$

These expressions in the Cartesian plane represent two lines. Because the measured distances have so many inherent errors in the system due to noise, in general the three circles do not intersect at one point, but in the best case only the intersection of sets of two circles can be ensured. The lines obtained corresponded to those that pass through the two intersection points of the involved circles, or, perpendicular to the line that passes through their centers and that pass through the intermediate space between them.

Example: Sensor Reading

The following figure illustrates this result:



Example: Sensor Reading

Then the robot position calculated by these landmarks $\underline{z}_1(k+1)$ is as follows:

$$x = \frac{(y_2-y_1)(r_2^2-r_3^2-x_2^2+x_3^2-y_2^2+y_3^2)-(y_3-y_2)(r_1^2-r_2^2-x_1^2+x_2^2-y_1^2+y_2^2)}{2(x_3-x_2)(y_2-y_1)-2(x_2-x_1)(y_3-y_2)}$$

$$y = \frac{(x_2-x_1)(r_2^2-r_3^2-x_2^2+x_3^2-y_2^2+y_3^2)-(x_3-x_2)(r_1^2-r_2^2-x_1^2+x_2^2-y_1^2+y_2^2)}{2(x_2-x_1)(y_3-y_2)-2(x_3-x_2)(y_2-y_1)}$$

To find the angle θ an IMU (inertial measurement unit) is used.

Example: Robot Pose Using Sensors

So the pose of the robot using sensors is:

$$\underline{z}_j(k+1) = \; h_{map}\left(\underline{x}(k+1),\; M_j\right) + \underline{\omega}_j(k+1)$$

$$= (x, y, \theta)$$

Where $\underline{\omega}(k+1)$ is the noise that is added when the robot position is found with the sensors. The values of the covariance matrix of this noise, $R(k+1)$, are found experimentally by externally measuring the real position of the robot by finding the aggregate noise $\underline{\omega}(k+1)$.

$$R(k+1) = \begin{bmatrix} \sigma^2_{\omega_x} & 0 & 0 \\ 0 & \sigma^2_{\omega_y} & 0 \\ 0 & 0 & \sigma^2_{\omega_\theta} \end{bmatrix}$$

Example: Robot Pose Using Sensors

Therefore the function:

$$\underline{h}_1 = h_{map}\left(\underline{x}(k+1),\ M_1\right) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

The Jacobian of the matrix $\underline{h}_1$ is the identity matrix I:

$$\nabla \underline{h}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example: Measurement Prediction

With the sensing simulator, the estimated positions of the robot are predicted, based on simulating the readings of the new estimated position of the robot $\underline{\hat{x}}(k+1|k)$ after simulating its movement of the position $\hat{x}(k)$.

$$\underline{\hat{z}}_1(k+1) = \ h_{map}\left(\underline{\hat{x}}(k+1|k),\ M_1\right)$$

What is predicted is compared with what is observed:

$$\underline{v}_{11}(k+1) = [\underline{z}_1\ (k+1)\ -\ \underline{\hat{z}}_1(k+1)]$$

$$= [\underline{z}_1\ (k+1)\ -\ h_{map}\left(\underline{\hat{x}}(k+1|k),\ M_1\right)]$$

Example: Measurement Prediction

The vector is formed:

$$\underline{\mathbf{v}}(k+1) = \{\underline{v}_{11}\}$$

The uncertainty of this error or covariance is:

$$S_{11}(k+1) = E\left[\underline{v}_{11}(k+1)\underline{v}_{11}^{T}(k+1)\right]$$

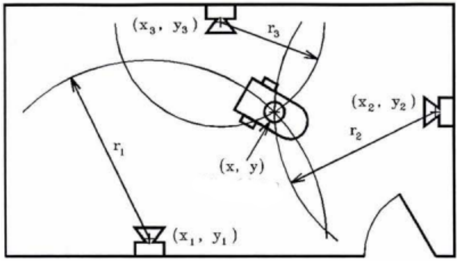
This covariance can be found with:

$$S_{11}(k+1) = \nabla \underline{h}_1 P(k+1|k) \nabla \underline{h}_1^{T} + R(k+1)$$

where $R\ (k+1)$ is the variance noise $\underline{\omega}(k+1)$

Measurement Model

For the case where sensor readings are used to find the position of the landmarks. There is a system that finds landmarks in the environment, finding the distance r_i and the angle α_i at which they are with respect to the position of the robot.



Example: Sensors Reading

For this case the function $\underline{h}_i$ is:

$$\underline{h}_i = h_{map}\left(\underline{x}(k+1),\ M_i\right) = \begin{bmatrix} r_i(k+1) \\ \alpha_i(k+1) \end{bmatrix} = \begin{bmatrix} \sqrt{((x-x_i)^2 + (y-y_i)^2)} \\ atan\left(\frac{(y-y_i)}{(x-x_i)}\right) \end{bmatrix}$$

and its Jacobian is:

$$\nabla \underline{h}_i = \begin{bmatrix} \frac{(x-x_i)}{\sqrt{((x-x_i)^2+(y-y_i)^2)}} & \frac{(y-y_i)}{\sqrt{((x-x_i)^2+(y-y_i)^2)}} & 0 \\ \frac{-(y-y_i)}{1+\left(\frac{(y-y_i)}{(x-x_i)}\right)^2 (x-x_i)^2} & \frac{1}{1+\left(\frac{(y-y_i)}{(x-x_i)}\right)^2 (x-x_i)} & -1 \end{bmatrix}$$

Example: Sensors Reading

With the sensing simulator, it is predicted which are the positions of the marks in the environment, from simulating the readings of the new estimated position of the robot $\hat{\underline{x}}(k+1|k)$ after simulating its move from position $\hat{x}(k)$.

$$\hat{\underline{z}}_i(k+1) = \ h_{map}\left(\hat{\underline{x}}(k+1|k),\ M_i\right)$$

In the same way as before, the predicted is compared with the observed:

$$\underline{v}_{ij}(k+1) = \left[\underline{z}_i\ (k+1)\ -\ \hat{\underline{z}}_j(k+1)\right]$$

$$= \left[\underline{z}_i\ (k+1)\ -\ h_{map}\left(\hat{\underline{x}}(k+1|k),\ M_j\right)\right]$$

Example: Measurement Prediction

The vector is formed:

$$\underline{\mathbf{v}}(k+1) = \{\underline{v}_{ij}\}$$

As was done in the other case, the uncertainty of this error or covariance is:

$$S_{ij}(k+1) = E\left[\underline{v}_{ij}(k+1)\underline{v}_{ij}^{T}(k+1)\right]$$

This covariance can be found with:

$$S_{ij}(k+1) = \nabla\underline{h}_i P(k+1|k)\nabla\underline{h}_i^{T} + R(k+1)$$

where $R\ (k+1)$ is the variance of the noise $\underline{\omega}(k+1)$

Jacobian of the sensing of simulation

$$\nabla \underline{h}_i = \left[\begin{matrix} \frac{(\hat{x}(k+1|k)-x_i)}{\sqrt{((\hat{x}(k+1|k)-x_i)^2+(\hat{y}(k+1|k)-y_i)^2)}} & \frac{(\hat{y}(k+1|k)-y_i)}{\sqrt{((\hat{x}(k+1|k)-x_i)^2+(\hat{y}(k+1|k)-y_i)^2)}} & 0 \\ \frac{-(\hat{y}(k+1|k)-y_i)}{1+\left(\frac{(\hat{y}(k+1|k)-y_i)}{(\hat{x}(k+1|k)-x_i)}\right)^2(\hat{x}(k+1|k)-x_i)^2} & \frac{1}{1+\left(\frac{(\hat{y}(k+1|k)-y_i)}{(\hat{x}(k+1|k)-x_i)}\right)^2(\hat{x}(k+1|k)-x_i)} & -1 \end{matrix} \right.$$

Example: Kalman Gain

Finally the following position is predicted by finding the Kalman gain:

$$\mathbf{K}(k+1) = P(k+1|k)\nabla h_i^T \cdot \left[\nabla h_i P(k+1|k)\nabla h_i^T + R(k+1)\right]^{-1}$$

$$K(k+1) = \begin{bmatrix} k_{xx} & k_{xy} & k_{x\theta} \\ k_{yx} & k_{yy} & k_{y\theta} \\ k_{\theta x} & k_{\theta y} & k_{\theta\theta} \end{bmatrix}$$

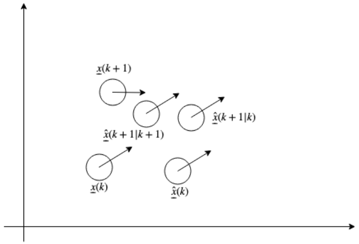
And the covariance associated with this position is:

$$P(k+1|k+1) = P(k+1|k) + \mathbf{K}(k+1)S(k+1)\mathbf{K}^T(k+1)$$

Example: Kalman Gain

Therefore the estimation of the next robot position is:

$$\underline{\hat{x}}(k+1|k+1) = \underline{\hat{x}}(k+1|k) + \mathbf{K}(k+1)\underline{\mathbf{v}}(k+1)$$



MATLAB Implementation

In the following GitHub repository, there is a MATLAB implementation using MXL modules explaining the EKF, with instructions on how to use this software:
https://github.com/RobotJustina/MRS_EKF_Matlab.git