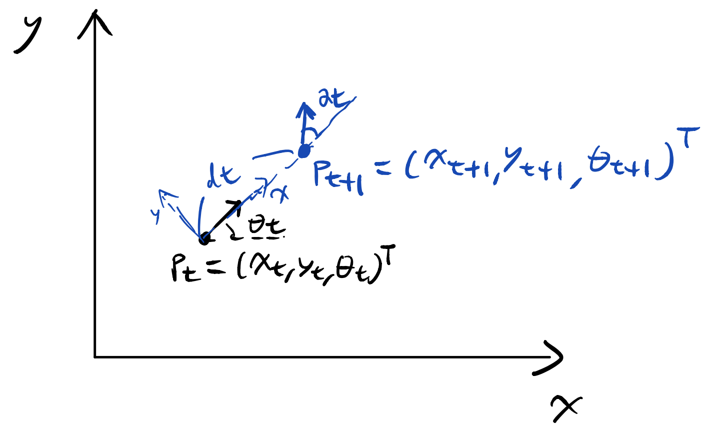
16833 HW2

Jae seok oh (jaeseoko)



y
x
α_t
d_t
θ_t
$P_{t+1} = (x_{t+1}, y_{t+1}, \theta_{t+1})^T$
$P_t = (x_t, y_t, \theta_t)^T$

$x_{t+1} = x_t + d_t \cos\theta_t$

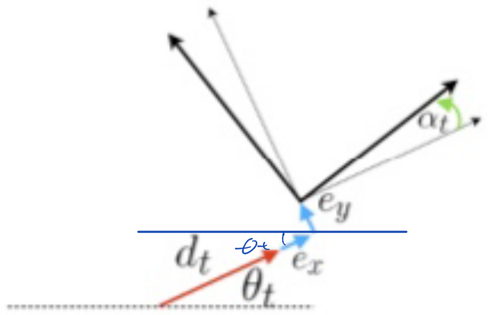
$y_{t+1} = y_t + d_t \sin\theta_t$

$\theta_{t+1} = \theta_t + \alpha_t$

1. Theory

① $P_{t+1} = g(u_t, P_{t-1})$

$u_t = \begin{bmatrix} d_t \\ \alpha_t \end{bmatrix}$



$$\boxed{\overset{\overline{\mu_t}}{\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix}} = \begin{pmatrix} x_t + d_t \cdot \cos\theta_t \\ y_t + d_t \cdot \sin\theta_t \\ \theta_t + \alpha_t \end{pmatrix}}$$

② $\bar{\mu}_{t+1} = g(u_t, \mu_t, \varepsilon_t)$

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t + (d_t + e_x) \cdot \cos\theta_t - e_y \sin\theta_t \\ y_t + (d_t + e_x) \cdot \sin\theta_t + e_y \cos\theta_t \\ \theta_t + \alpha_t + e_\alpha \end{pmatrix}$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T$$

$\downarrow$

$$\bar{\Sigma}_{t+1} = G_t \Sigma_t G_t^T + R_t$$

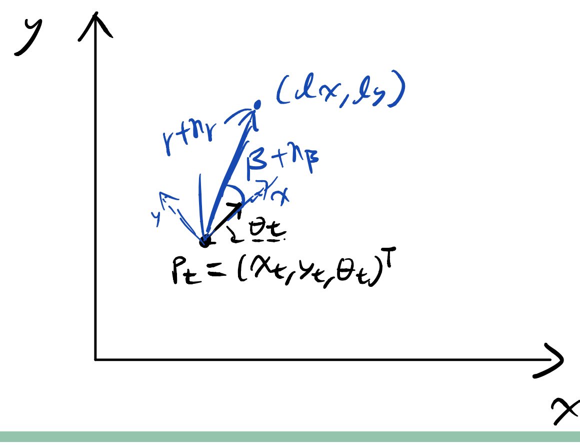
$$G_t = \frac{\partial g}{\partial x_{t-1}} = \begin{bmatrix} 1 & 0 & -d_t \sin\theta_t \\ 0 & 1 & d_t \cos\theta_t \\ 0 & 0 & 1 \end{bmatrix}$$

$$L_t = \frac{\partial g}{\partial e} = \begin{bmatrix} \cos\theta_t & -\sin\theta_t & 0 \\ \sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma_e = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

$$\begin{aligned} \bar{\Sigma}_{t+1} &= G_t \Sigma_t G_t^T + L_t \Sigma_e L_t^T \\ &= G_t \Sigma_t G_t^T + R_t \end{aligned}$$

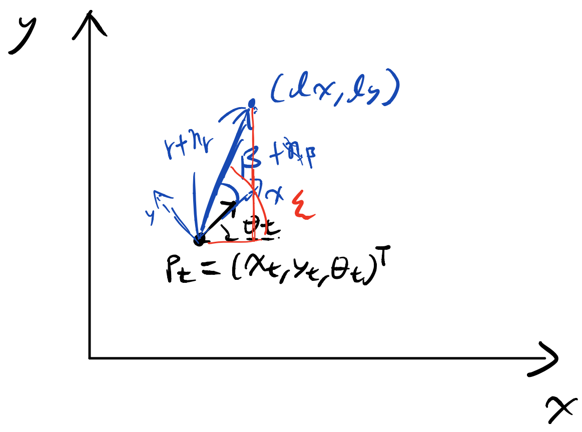
③



$$l_x = x_t + (r+n_r)\cos(\beta+n_\beta+\theta_t)$$
$$l_y = y_t + (r+n_r)\sin(\beta+n_\beta+\theta_t)$$

$$\boxed{\begin{Bmatrix} l_x \\ l_y \end{Bmatrix} = \begin{Bmatrix} x_t + (r+n_r)\cos(\beta+n_\beta+\theta_t) \\ y_t + (r+n_r)\sin(\beta+n_\beta+\theta_t) \end{Bmatrix}}$$

④ $$\begin{Bmatrix} \beta \\ r \end{Bmatrix} = h(P_t, l_x, l_y, n_\beta, n_r)$$



$$(r+n_r) = \sqrt{(l_x-x_t)^2 + (l_y-y_t)}^2$$

$$\xi = (\beta + n_\beta + \theta_t)$$

$$\xi = \tan^{-1}\left(\frac{l_y - y_t}{l_x - x_t}\right)$$

$$\beta + n_\beta = \tan^{-1}\left(\frac{l_y-y_t}{l_x-x_t}\right) - \theta_t$$

$$h = \begin{Bmatrix} \beta \\ r \end{Bmatrix} = \begin{Bmatrix} \text{wrap2pi}\left(\text{atan2}(l_y - y_t, l_x - x_t)\right) - \theta_t \\ \sqrt{(l_x-x_t)^2 + (l_y-y_t)^2} \end{Bmatrix}$$

with noise $\begin{bmatrix} \sigma_\beta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$. $n_\beta \sim N(0, \sigma_\beta^2)$, $n_r \sim N(0, \sigma_r^2)$

⑤

$$H_p = \frac{\partial h}{\partial x_t} = \begin{bmatrix} \frac{\partial h_1}{\partial x_t} & \frac{\partial h_1}{\partial y_t} & \frac{\partial h_1}{\partial \theta_t} \\ \frac{\partial h_2}{\partial x_t} & \frac{\partial h_2}{\partial y_t} & \frac{\partial h_2}{\partial \theta_t} \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{(y_t - l_y)}{(l_x-x_t)^2+(l_y-y_t)^2} & -\frac{(l_x - x)}{(l_x-x_t)^2+(l_y-y_t)^2} & -1 \\ \frac{(x_t - l_x)}{\sqrt{(l_x-x_t)^2+(l_y-y_t)^2}} & \frac{(y_t - l_y)}{\sqrt{(l_x-x_t)^2+(l_y-y_t)^2}} & 0 \end{bmatrix}$$

⑥

$$H_l = \frac{\partial h}{\partial l} = \begin{bmatrix} \frac{\partial h_1}{\partial l_x} & \frac{\partial h_1}{\partial l_y} \\ \frac{\partial h_2}{\partial l_x} & \frac{\partial h_2}{\partial l_y} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{(y_t - l_y)}{(x_t - l_x)^2 + (y_t - l_y)^2} & \frac{(l_x - x_t)}{(x_t - l_x)^2 + (y_t - l_y)^2} \\ \frac{(l_x - x_t)}{\sqrt{(x_t - l_x)^2 + (y_t - l_y)^2}} & \frac{(l_y - y_t)}{\sqrt{(x_t - l_x)^2 + (y_t - l_y)^2}} \end{bmatrix}$$

$\rightarrow$

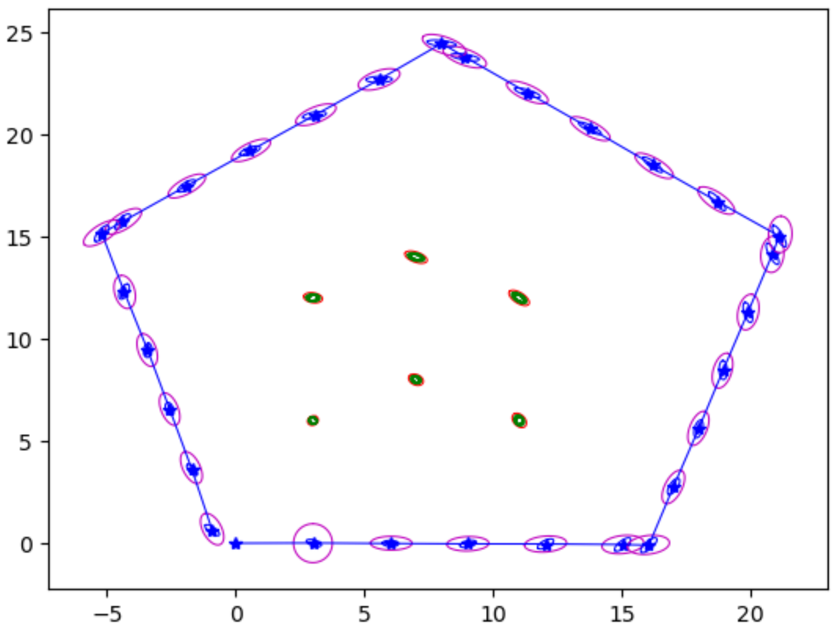
We assume that landmark measurement w.r.t robot is independent to other landmark measurements, thus we only calculate Jacobian of itself.

2. Implementation and evaluation

①

Fixed number of landmarks being observed = 6

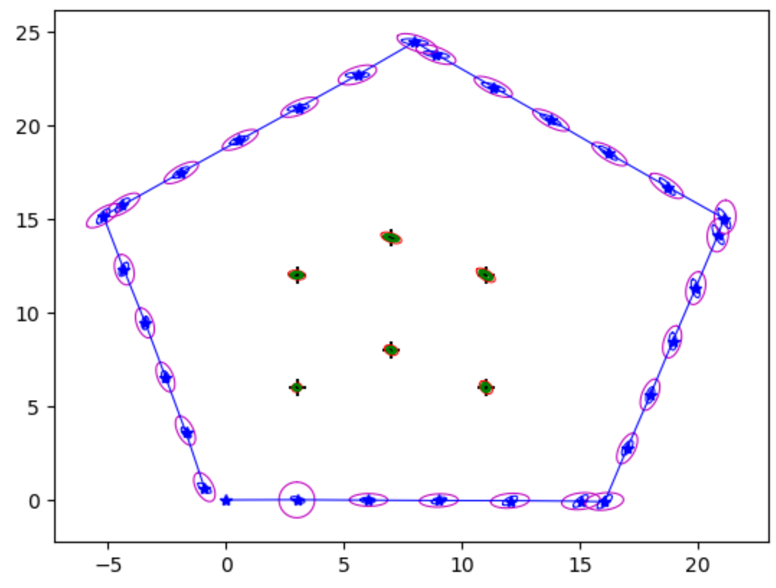
②



③

The EKF algorithm first computes predicted covariance in prediction step. Then in the update step, EKF calculates Kalman gain based on the predicted covariance. The kalman gain is a relative weight to a measurement compared to current state estimate. If it is low due to low uncertainty, EKF will trust prediction more and if the gain is high due to high uncertainty, EKF will put more weight on measurements and correct the belief based on the measurements. The kalman gain updates the covariance and state belief, and EKF assesses how good the prediction is and effectively chooses best balance of how much to follow prediction and measurements.

④



+ : Landmark Ground-truth

Euclidean:	0.00215488, 0.00405229, 0.00255037, 0.00282809, 0.00201858, 0.00399589
Mahalanobis:	0.05673994, 0.07470501, 0.05840717, 0.07296973, 0.03440558, 0.10803126

->

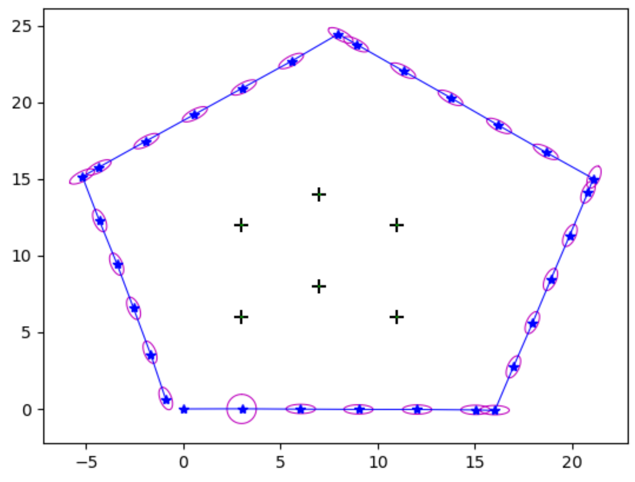
Euclidean distance calculates geometric distance between two points. Mahalanobis distance is a measure of distance between a point and a distribution. It effectively tells us how many standard deviations away the point is from the mean of the distribution. Euclidean distance strictly tells us the mere distance, but Mahalanobis tells how likely it is that the point really belongs to the distribution. For example, we could conclude that it is unlikely that a point belongs to the distribution if Mahalanobis distance is more than 2 standard deviations away even if the euclidian distance is very small. It give us the idea of how well our landmark estimate actually captures the ground truth value.

3.Discussion

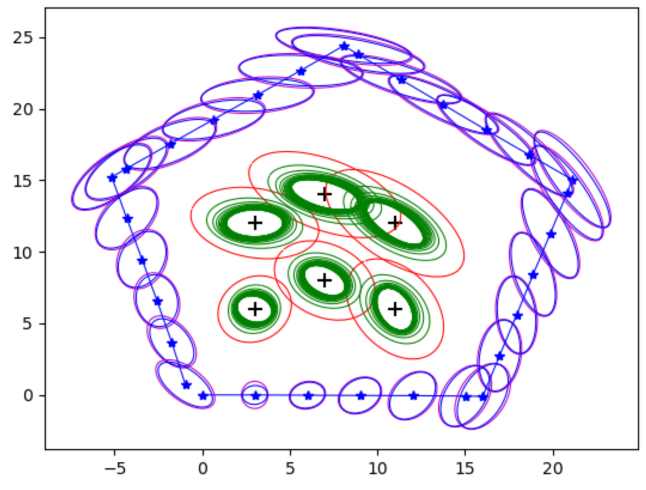
①

The initial landmark covariance matrix is a block-diagonal matrix with other terms as zeros. We initially assumed the cross-covariance between different landmarks and between landmarks and robot pose is zero. They are actually not independent since landmark measurements depend on the pose that measured the landmark, and landmarks are conditionally independent given the robot pose. Thus, we see that our P (state covariance matrix) does have non-zero terms in the off diagonals.

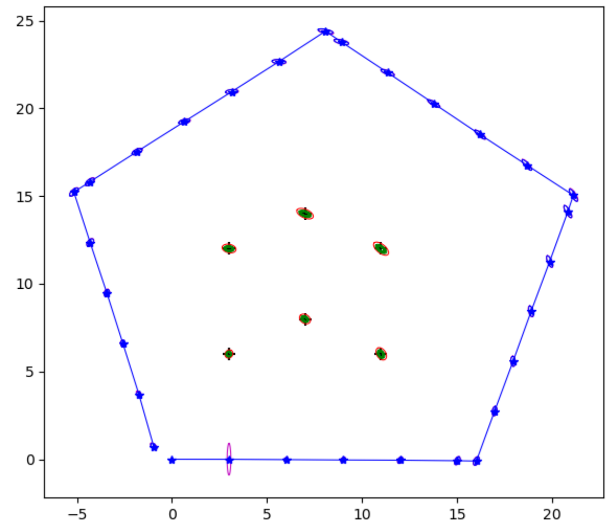
②



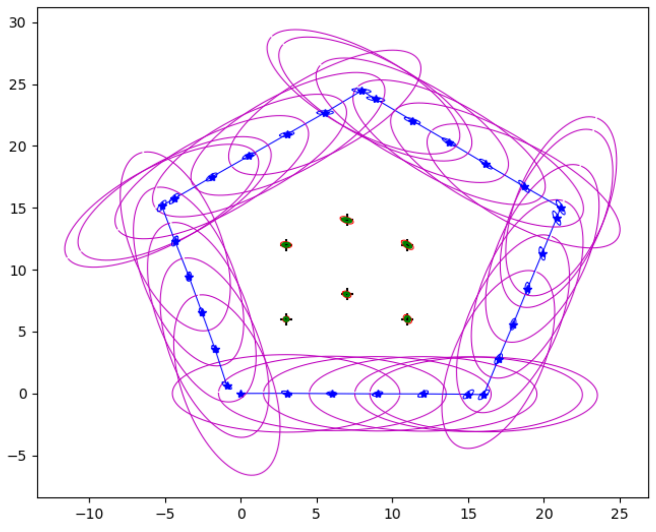
A) $\sigma_\beta, \sigma_r \times 0.1$



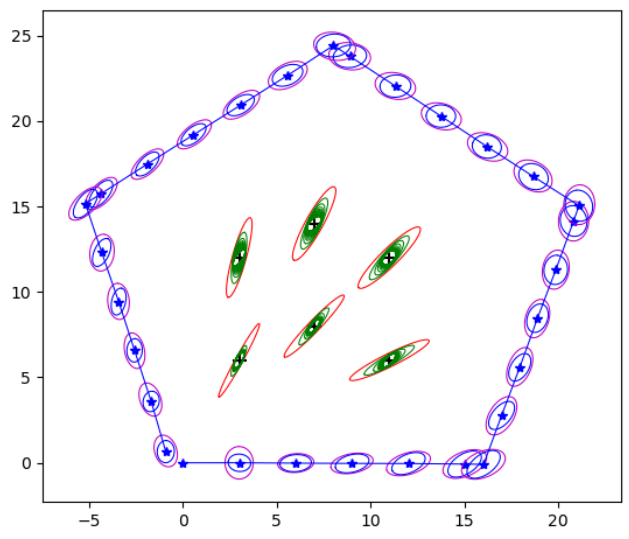
B) $\sigma_\beta, \sigma_r \times 10$



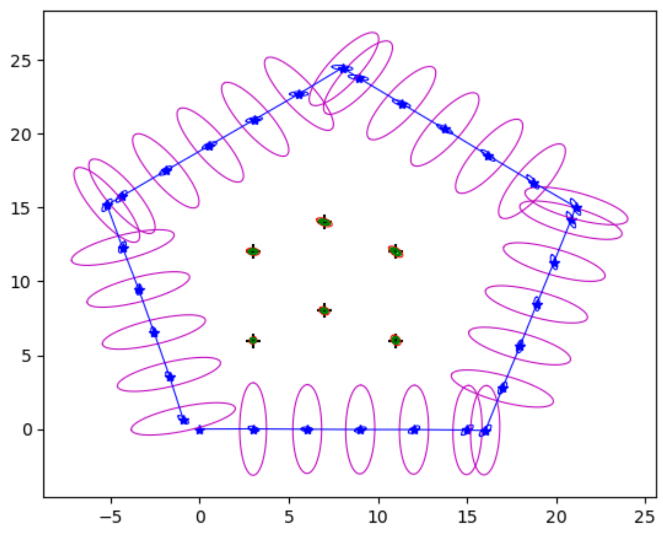
C) $\sigma_x, \sigma_y, \sigma_\alpha \times 0.1$



D) $\sigma_x, \sigma_y, \sigma_\alpha \times 10$



E) $\sigma_r \times 10$



F) $\sigma_y \times 10$

->

In figures A and B, measurement uncertainties were modified. In both figures, we can see that the measurement uncertainties affect both state and landmark covariance since our poses are predicted and updated based on measurements and high uncertainty will increase the covariance in both the poses and landmarks whereas low uncertainty yields very small covariances.

In figures C and D, control uncertainties were modified. Having large uncertainty in control means our prediction will be very bad since it predicts based on the motion. Thus we see big prediction pose covariances in magenta. However, since the Kalman gain will be big in this case, EKF effectively corrects the pose covariance based on the measurements.

In figures E and F, range uncertainty and control in y-direction uncertainty were increased by 10 times to see the effect. We can more intuitively see which variable affects the which axes of covariances. For instance in F, we clearly see that the covariance is spread out along the y-axis of each robot pose.

③

First method that is most intuitive is to set a certain number 'N' of landmark observations to keep. Then, in every iteration if we see more than N landmarks, we only keep the N landmarks that have higher Kalman gains than the rest. This will ensure we get the most useful information while keeping the computation time capped at certain threshold.

Second method is to sparsely select measurements. Similar to subsampling ray casts in particle filter, we can skip over the measurements, considering every other measurements for example. This will reduce the computation time while still mapping the surroundings.

Third method could be to only update with full landmark observations when predicted state covariance grows larger than some set threshold. If it is smaller than the threshold, we can update based only on some set smaller number of landmarks. In setting this small number of landmarks, we should consider prioritizing closer ones since we still want to map the adjacent surroundings decently and to avoid bumping into obstacles by not accounting for near landmarks.