1 Theory

(a) The position of the robot at time $t+1$ is given by

$$p_{t+1} = p_t + \begin{bmatrix} d_t * \cos(\theta_t) \\ d_t * \sin(\theta_t) \\ \alpha_t \end{bmatrix}$$

(b) From (a), we have

$$p_{t+1} = \begin{bmatrix} x_t + d_t * \cos(\theta_t) \\ y_t + d_t * \sin(\theta_t) \\ \theta_t + \alpha_t \end{bmatrix}$$

Thus, the Jacobian of the pose at time $t+1$ is given by,

$$G_{t+1} = \begin{bmatrix} 1 & 0 & -d_t \cos(\theta_t) \\ 0 & 1 & d_t \sin(\theta_t) \\ 0 & 0 & 1 \end{bmatrix}$$

The noise parameters of the system are given by

$$R_{t+1} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix}$$

From the update equation of EKF, we have,

$$\begin{aligned} \Sigma_{t+1} &= G_{t+1}\Sigma_t G_{t+1}^T + R_{t+1} \\ &= \begin{bmatrix} 1 & 0 & -d_t \cos(\theta_t) \\ 0 & 1 & d_t \sin(\theta_t) \\ 0 & 0 & 1 \end{bmatrix} \Sigma_t \begin{bmatrix} 1 & 0 & -d_t \cos(\theta_t) \\ 0 & 1 & d_t \sin(\theta_t) \\ 0 & 0 & 1 \end{bmatrix}^T + \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix} \end{aligned}$$

Now given that the uncertainty for at time t is given by $\mathcal{N}(0, \Sigma_t)$, we have the uncertainty at time $t+1$ is given by

$$\mathcal{N}(0, \Sigma_{t+1}) = \begin{bmatrix} 1 & 0 & -d_t \cos(\theta_t) \\ 0 & 1 & d_t \sin(\theta_t) \\ 0 & 0 & 1 \end{bmatrix} \mathcal{N}(0, \Sigma_t) \begin{bmatrix} 1 & 0 & -d_t \cos(\theta_t) \\ 0 & 1 & d_t \sin(\theta_t) \\ 0 & 0 & 1 \end{bmatrix}^T + \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix}$$

Thus,

$$\mathcal{N}(0, \Sigma_{t+1}) = G_{t+1}\mathcal{N}(0, \Sigma_t)G_{t+1}^T + R_{t+1}$$

(c) Given the state estimate p_t and some measurement $z = \begin{pmatrix} \beta \\ r \end{pmatrix}$, we have the global position of the landmarks give by

$$\begin{aligned} \bar{L} &= \begin{bmatrix} l_x \\ l_y \end{bmatrix} \\ &= \begin{bmatrix} x_t + r * \cos(\theta_t + \beta) \\ y_t + r * \sin(\theta_t + \beta) \end{bmatrix} \end{aligned}$$

Now given the noise in the measurements $\eta_\beta \sim \mathcal{N}(0, \sigma_\beta^2)$ and $\eta_r \sim \mathcal{N}(0, \sigma_r^2)$, we can define the noise matrix for the measurements as

$$R = \begin{bmatrix} \sigma_\beta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$$

We compute the value $A = \frac{\partial \bar{L}}{\partial z}$ and we get

$$A = \begin{bmatrix} -r * \sin(\theta_t + \beta) & \cos(\theta_t + \beta) \\ r * \cos(\theta_t + \beta) & \sin(\theta_t + \beta) \end{bmatrix}$$

Then, the covariance of the location estimates can be given by $\bar{L}_{cov} = ARA^T$. Thus, the location estimates $l = (l_x, l_y)$ can be given by the mean $\bar{L}$ and covariance $\bar{L}_{cov}$.

(d) At time t, we have $p_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}$ and we have $\bar{L} = \begin{bmatrix} l_x \\ l_y \end{bmatrix}$ defined. We define two quantities $\delta_x = l_x - x_t$ and $\delta_y = l_y - y_t$. We will define the quantity $q = \delta_x^2 + \delta_y^2$. From this, we can predict the bearing and range as

$$\begin{aligned} h(p) &= \begin{bmatrix} \beta \\ r \end{bmatrix} \\ &= \begin{bmatrix} warpToPi(atan2(\delta_y, \delta_x) - \theta_t) \\ \sqrt{q} \end{bmatrix} \end{aligned}$$

with noise R as

$$R = \begin{bmatrix} \sigma_\beta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$$

(e) Computing the Jacobian H_p with respect to the pose, we get,

$$
\begin{aligned}
H_p &= \begin{bmatrix} \frac{\partial h_1}{\partial x_t} & \frac{\partial h_1}{\partial y_t} & \frac{\partial h_1}{\partial \theta_t} \\ \frac{\partial h_2}{\partial x_t} & \frac{\partial h_2}{\partial y_t} & \frac{\partial h_2}{\partial \theta_t} \end{bmatrix} \\
&= \frac{1}{q} \begin{bmatrix} \delta_y & -\delta_x & -q \\ -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 \end{bmatrix}
\end{aligned}
$$

(f) Computing the Jacobian H_l with respect to the landmarks, we get,

$$
\begin{aligned}
H_l &= \begin{bmatrix} \frac{\partial h_1}{\partial l_x} & \frac{\partial h_1}{\partial l_y} \\ \frac{\partial h_2}{\partial l_x} & \frac{\partial h_2}{\partial l_y} \end{bmatrix} \\
&= \frac{1}{q} \begin{bmatrix} -\delta_y & \delta_x \\ \sqrt{q}\delta_x & \sqrt{q}\delta_y \end{bmatrix}
\end{aligned}
$$

For each landmark, we make the assumption that its measurement with respect to the robot is independent of any other landmark and hence, we do not need to calculate the measurement Jacobian with respect to other landmarks except for itself.

2 Implementation

(a) The number of fixed landmarks being observed over the entire sequence is `6`.

(b) The final image showing the robot positions and the robot trajectory is shown in Figure 1.

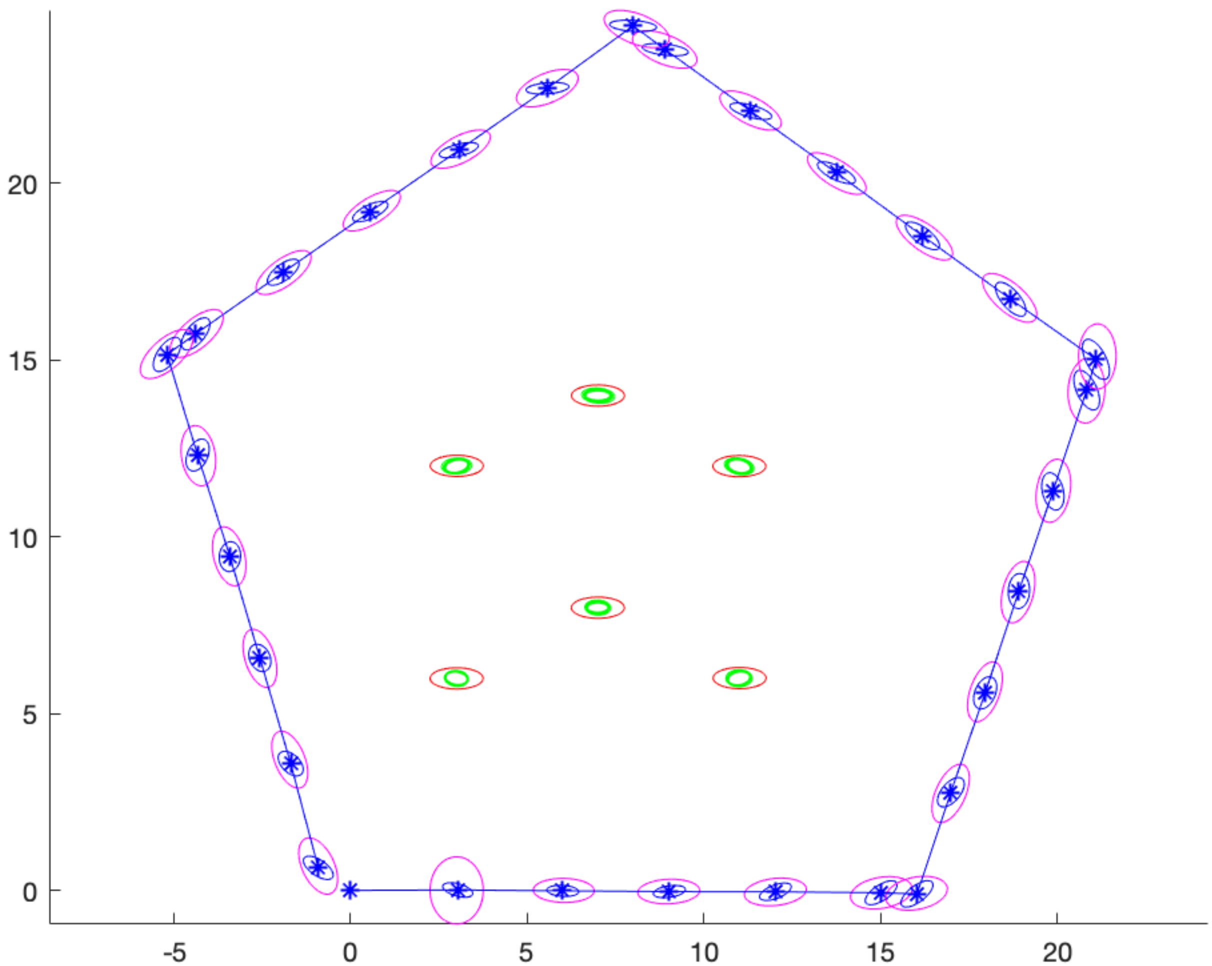

Figure 1: The blue lines show the robot trajectory. The blue '*' points represent a position of the robot over the sequence of observations without orientation. The uncertainties of the positions of both the robot and the landmarks are shown as 3-sigma error ellipses.

(c) The EKF algorithm uses the predicted covariance to compute the quantity K, the Kalman gain - which is a measurement that gives relative weight to the measurements and the current state estimate. When the gain is higher, the algorithm places more weight on the most recent measurements and follows them more closely. When the gain is low, the algorithm inherently follows the model predictions more closely. The Kalman gain is used to update the predicted covariance to give the actual covariance of the system. Let's say that the predicted uncertainty is low, which means that the Kalman gain is effectively low and hence the algorithm realizes it is performing well and hence it just computes the predictions and follows them. However, when the prediction uncertainty is high, the gain is inherently higher and hence the algorithm needs to update its uncertainty to account for the latest measurements and update its predictions accordingly. The EKF algorithm uses the entire state of the system to keep track of how good the predictions and the uncertainty (diagonals of the covariance matrix) are.

(d) The final results with ground truth landmark positions are shown below in Figure 2. The Euclidean and the Mahalanobis distance of each of the 6 landmarks are shown in Table 1.

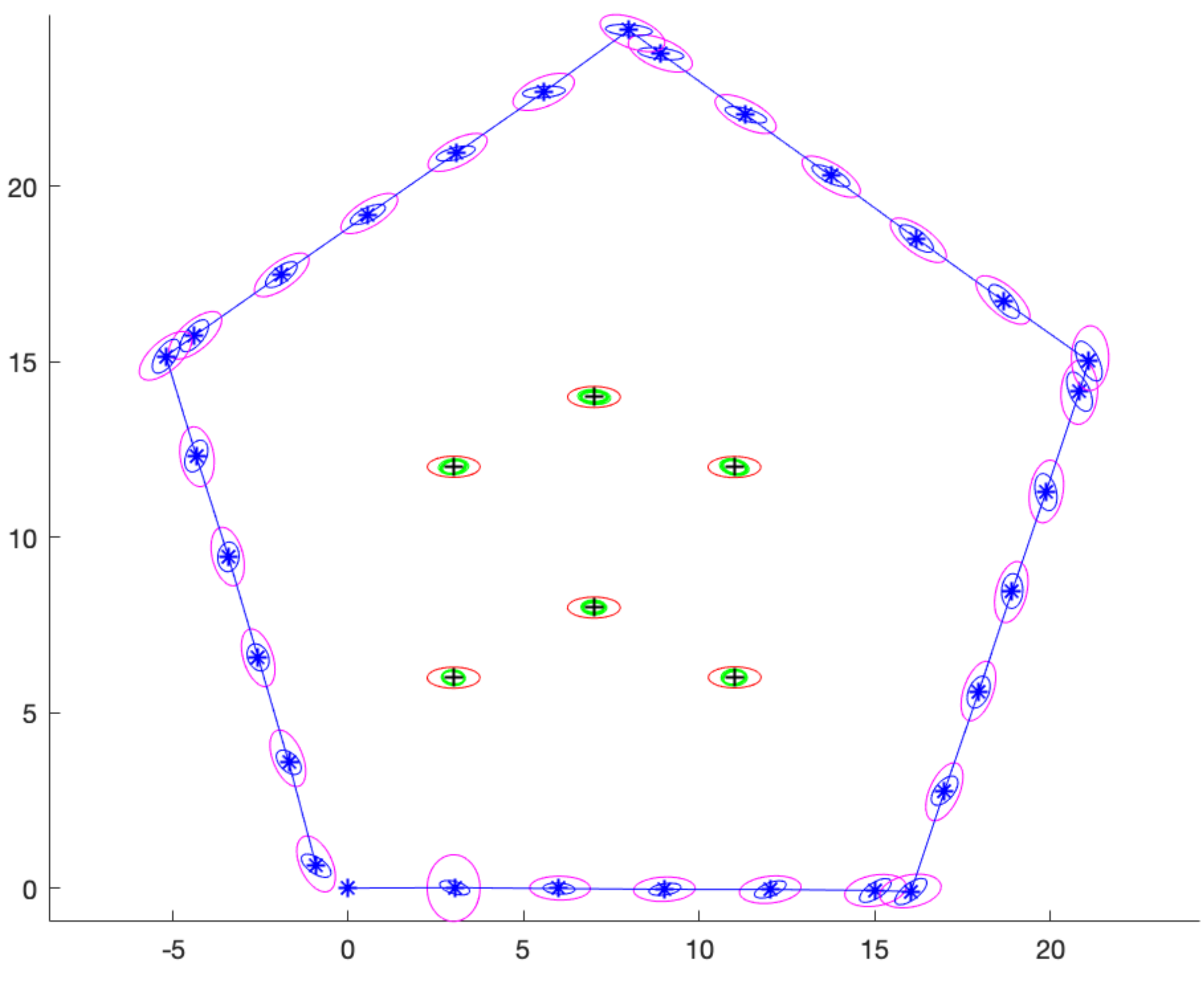

Figure 2: The black '+' represent the ground truth locations of the landmarks

The Euclidean distance computes the geometric distance in the spatial dimension. We can see from the table above that the Euclidean distances are very small - meaning that in the absolute 2D space (x, y), we are very close to the original ground truth. The Mahalanobis distance measures the distance between a point P and a distribution D - essentially measuring how the number of standard deviations away P is from the mean of D. The distance is zero if P is the mean of D. Since the numbers in the table above

Landmark #	Euclidean Distance	Mahalanobis Distance
1	0.0055	0.0005
2	0.0036	0.0002
3	0.0061	0.0006
4	0.0043	0.0004
5	0.0052	0.0005
6	0.0013	0.0001

Table 1: Euclidean and Mahalanobis distances for each of the 6 landmarks.

are very small, it means that we have a very good representation of the probability distribution of the landmarks (their μ, Σ).

The ground truth point for each of the landmarks are inside the smallest ellipse. This has two derivatives. The first that the updates that the algorithm performed are correct and that the ground truth can be sampled from the estimated distribution with very high accuracy.

3 Discussions

(a) The initial zero off-diagonal terms in the state covariance matrix become non-zero as we apply updates. Each prediction and update involved projection control and measurement updates into the state space and incorporating those uncertainties into our state covariance matrix, with the effect of creating uncertainty of each state element relative to other elements.

To initialize the state covariance matrix, we assume that each state element's variance is independent of all other state elements (diagonal covariance matrix). This is actually not true, as we form the initial estimates based on measurements from the same position, but EKF adapts quickly enough to nullify this assumption.

(b) I performed 2 experiments with the different parameters. First, I kept $\sigma_x, \sigma_y, \sigma_\alpha$ fixed and varied σ_β, σ_r. In the variations, I multiplied both parameters by 10x. The results for the above two experiments are shown in Figure 3. In the second experiment, I kept σ_β, σ_r fixed and multiplied $\sigma_x, \sigma_y, \sigma_\alpha$ by 5x. The results for the above two experiments are shown in Figure 4.

From Figure 3, we can see that when we increase the noise in measurement values, the state estimate means appear to be track similar to the previous runs (predefined values), however the covariance matrices are significantly larger, implying that the localization problem has become more uncertain while the mapping capabilities remain almost similar (with some increased uncertainty).

From Figure 4, we can see that increasing the noise in pose values, the overall covariance matrices for both the landmarks and the positions of the robot are very large, resulting in very high uncertainty in both localization and mapping from the robot.

(c) One approach for the exploration problem would be to initialize an arbitrarily large number of landmarks. For example, if there are known N landmarks, initialize with 3N landmarks instead and keep updating any new incoming landmarks in the same matrix - and use sparse matrix techniques to ensure the speed up in computation.

A second approach would be to keep the number of landmarks fixed over time and when we see any new

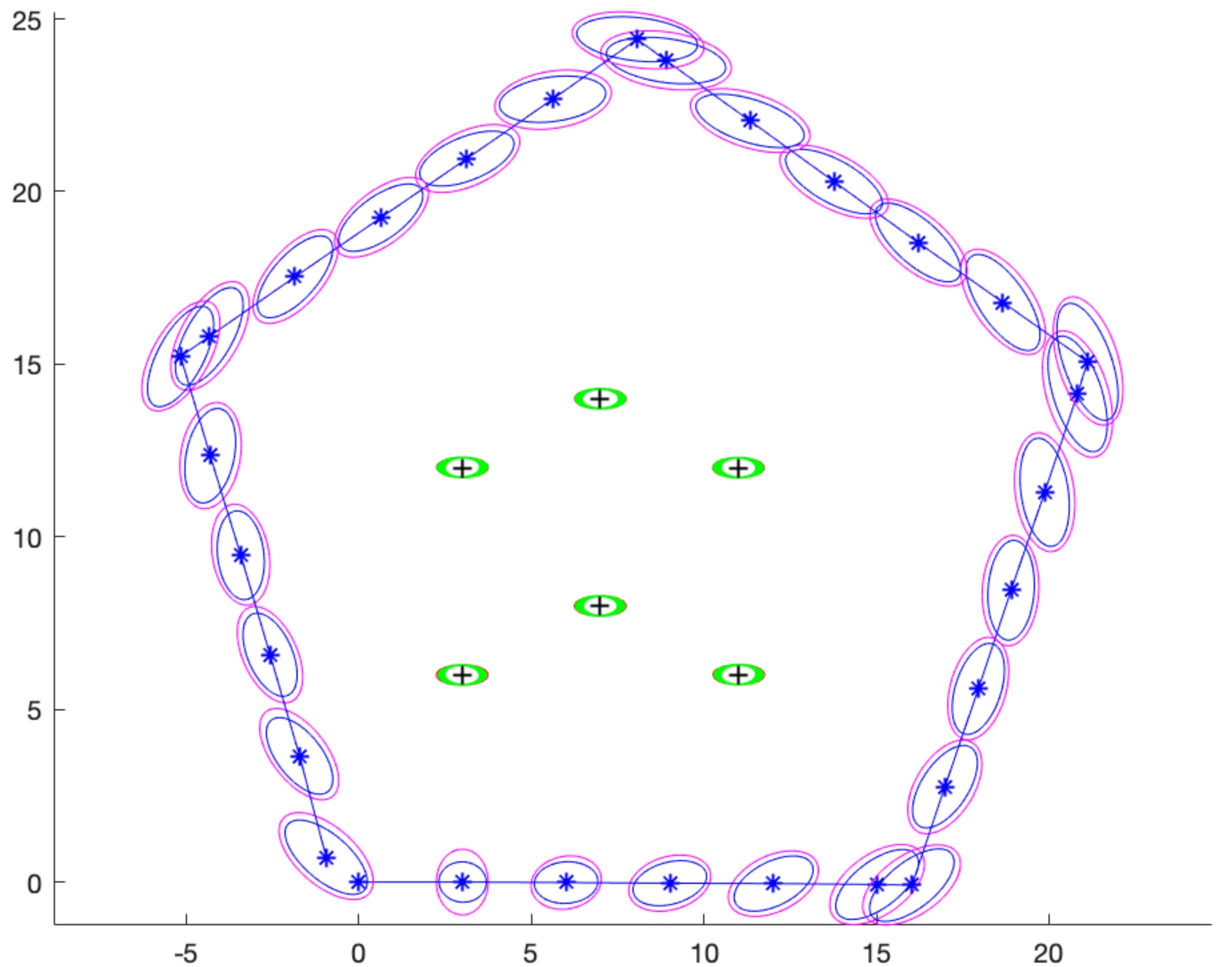

Figure 3: σ_β, σ_r * 10x

landmarks, remove the landmarks with the least Kalman gain in the previous iteration and add the new landmarks - this will require some re-initialization for the covariance of new landmarks.

A third approach - when there are maps with a very large number of landmarks, we can cluster closely related landmarks and replace them with a centroid landmark - this can reduce the computation.

4 References

[1] Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

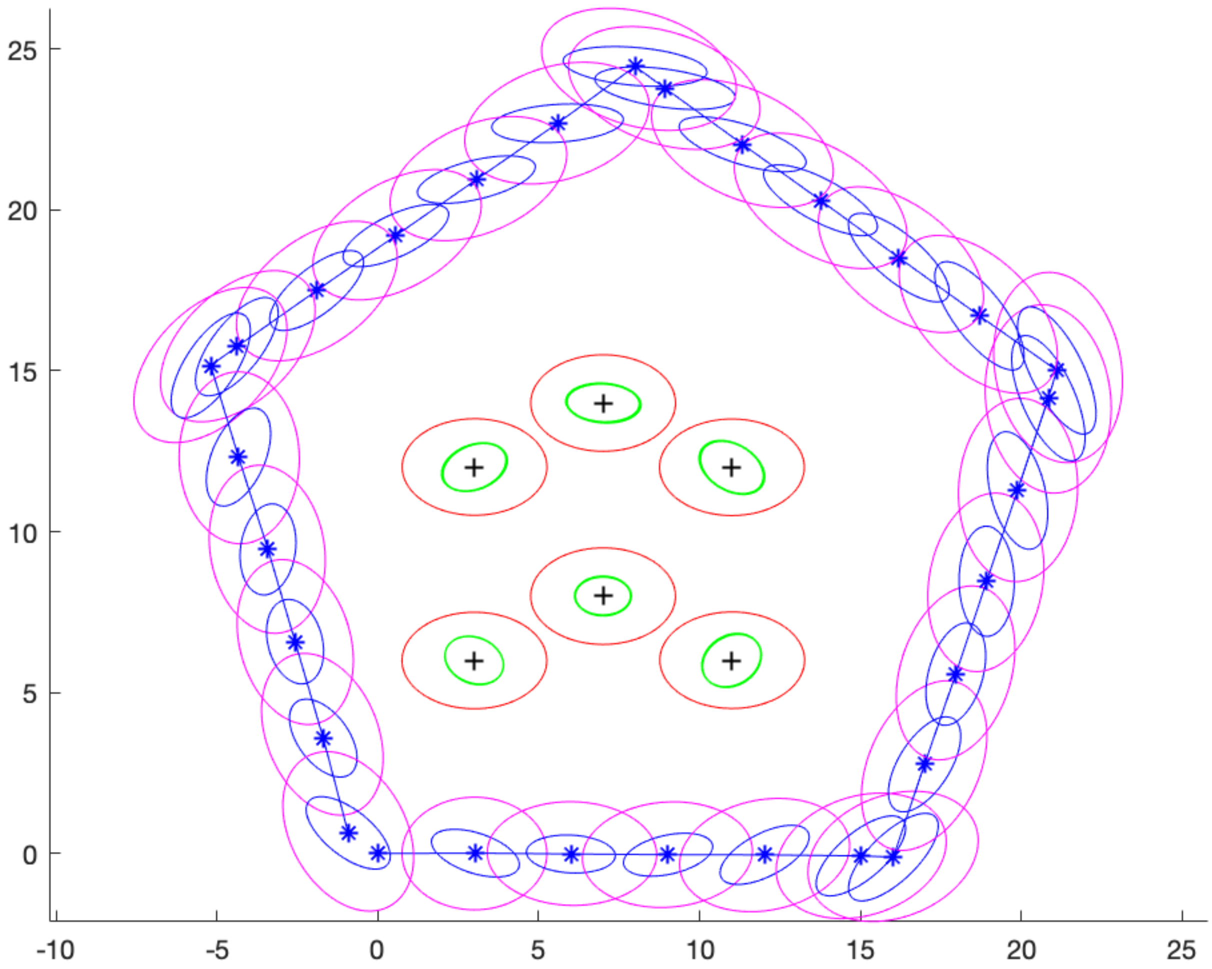

Figure 4: $\sigma_x, \sigma_y, \sigma_\alpha$ * 5x