2D EKF based deformable SLAM

Basic variables

Total number of the robot pose N.

Initial robot pose $Pose(1) = [\theta; x; y] = [0; 0; 0]$.

Total number of the feature M.

Feature position $feature$.

Local translation of the features $d_{local\_feature}(i, j)$.

State vector $\mathbf{X} = [\theta; \mathbf{xr}; \mathbf{xf}]$.

Covariance matrix $\mathbf{P}$.

The relative edge in the construction $constriant\_edge$.

Sensor range is from 0.1m to 2m.

Control vector $\mathbf{u} = [\omega; \mathbf{v}]$, where ω is the heading variation, and $\mathbf{v} = [\Delta x; \Delta y]$ is the relative shift.

Noise

Odometry noise $noise_{odem}$.

Smooth noise for the dynamic translation $noise_{smoo}$

Constructure noise for the local deformation $noise_{cons}$

Feature measurement noise $noise_{feat}$

The coresponding variances σ_{head}, σ_{odem}, σ_{smoo}, σ_{cons} and σ_{feat}.

As the value of σ_{feat} is not fixed, which means $\sigma_{feat}(i, j)$ is changing with i and j, we use σ_{Feat} to save the values in each step, that is,

$$\sigma_{Feat}(i) = [\sigma_{feat}(i, j = 1 : M)]$$

Initialize

Initialize the state vector $\mathbf{X}$ and the covariance matrix $\mathbf{P}$.

$$\hat{\mathbf{X}}_{1|1} = [\hat{\theta}_1; \hat{\mathbf{xr}}_{1|1}; \hat{\mathbf{xf}}_{1|1}],$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{0}_{3,3} & \mathbf{0}_{3,2M} \\ \mathbf{0}_{2M,3} & diag(\sigma_{Feat}) \end{bmatrix}$$

where, $\hat{\theta}_{1|1} = 0$, $\hat{\mathbf{xr}}_{1|1} = \mathbf{xr}_{1|1} = [0; 0]$, and $\hat{\mathbf{xf}}_{1|1,j} = feature_{frame}(1, j) + noise_{feat}(1, j)$.

Here, $feature_{frame}(1, j) = feature(1, j) + rotation_d * d_{local\_feature}(1, j) + noise_{cons}(1)$ is the position of features after the local deformation at time step 1.

Predict

Predict the next pose of the robot and the positions of the features according to the dynamic models.

The robot process model and the feature dynamic models of the features are all linear.

Robot process model

Here we use the function $getU(op)$ to obtain the control vector $\mathbf{u}$, where op is the calculation option.

When $op = 1$, the path is predetermined, which means $\mathbf{v}$ is fixed.

When $op = 2$, the path is planned actively based on the greedy method. Here, we considers to minimize the trace of the covariance matrix in the next step.

Then, the predcted next robot pose can be obtained by

$$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1} + \omega_{i-1}$$

$$\hat{\mathbf{xr}}_{i|i-1} = \hat{\mathbf{xr}}_{i-1|i-1} + \mathbf{R}(\hat{\theta}_{i-1|i-1})\mathbf{v}_{i-1},$$

where $\mathbf{R}(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$.

However, there are noises in reality, so the actual robot pose is

$$\theta_i = \theta_{i-1} + \omega_{i-1} + noise_{head}(i)$$

$$\mathbf{xr}_i = \mathbf{xr}_{i-1} + \mathbf{R}(\theta_{i-1|i-1})\mathbf{v}_{i-1} + noise_{odem}(i).$$

Feature dynamic models

The movement of the features includes the global translation $\mathbf{t}$ and the local defoamation $\mathbf{d}_{local\_feature}$.

As we consider the linear case only, there is no rotation for global translation and no rotation for local deformation. That is, $\mathbf{r}_t = \mathbf{I}_{2,2}$ and $\mathbf{r}_d = \mathbf{I}_{2,2}$.

$$\hat{\mathbf{xf}}_{i|i-1,j} = \hat{\mathbf{xf}}_{i-1|i-1,j} + \mathbf{r}_t\mathbf{t}(i) + \mathbf{r}_d\mathbf{d}_{local\_feature}(i,j).$$

Similarly, there are noises in reality, including the global translation noise and the local deformation noise.

For the global translation part,

$$\mathbf{xf}_{i,j} = \mathbf{xf}_{i-1,j} + \mathbf{r}_t\mathbf{t}(i) + \mathbf{r}_d\mathbf{d}_{local\_feature}(i,j) + noise_{smoo}(i,j).$$

For the local deformation part,

$$\mathbf{xf}_{frame}(i,j) = \mathbf{xf}_{i,j} + \mathbf{r}_d\mathbf{d}_{local\_feature}(i,j) + noise_{cons}(i,j).$$

Covariance matrix

$$\mathbf{P}_{i|i-1} = \mathbf{F}_i'\mathbf{P}_{i-1|i-1}\mathbf{F}_i' + \mathbf{G}_i\mathbf{Q}\mathbf{G}_i',$$

Here, the jacobian $\mathbf{F}_i$ and $\mathbf{G}_i$ can be obtained by

$$\mathbf{F}_i = \begin{bmatrix} 1 & \mathbf{0}_{1,2} & \mathbf{0}_{1,2M} \\ \mathbf{R}(\hat{\theta}_{i-1|i-1})\mathbf{J}\mathbf{v}_{i-1}^T & \mathbf{I}_{2,2} & \mathbf{0}_{2,2M} \\ \mathbf{0}_{2M,2} & \mathbf{0}_{2M,2} & \mathbf{I}_{2M,2M} \end{bmatrix}$$

$$\mathbf{G}_i = \begin{bmatrix} 1 & \mathbf{0}_{1,2} & \mathbf{0}_{1,2M} \\ \mathbf{0}_{2,1} & \mathbf{R}(\hat{\theta}_{i-1|i-1}) & \mathbf{0}_{2,2M} \\ \mathbf{0}_{2M,1} & \mathbf{0}_{2M,2} & \mathbf{I}_{2M,2M} \end{bmatrix},$$

and

$$\mathbf{Q} = \begin{bmatrix} \sigma_{odem}^2 * \mathbf{I}_{2,2} & \mathbf{0}_{2,2*M} \\ \mathbf{0}_{2*M,2} & (\sigma_{smoo}^2 + \sigma_{cons}^2) * \mathbf{I}_{2*M,2*M} \end{bmatrix},$$

with $\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

Observe

The observation model contains two parts, the measurement of the features and the measurement of the constructions.

$$\mathbf{Z} = [\mathbf{z}_{feat}; \mathbf{z}_{cons}].$$

Feature measurement is given by

$$\hat{\mathbf{z}}_{feat}(i,j) = \mathbf{R}(\hat{\theta}_{i-1|i-1})\hat{\delta}(i,j),$$

where $\hat{\delta}(i,j) = \hat{\mathbf{xr}}_{i|i-1} - \hat{\mathbf{xf}}_{i|i-1,j}$.

Construction measurement between feature $feature(j_1)$ and $feature(j_2)$ is given by

$$\hat{\mathbf{z}}_{cons}(i,j1,j2) = \hat{\mathbf{f}}_{loc}(i,j_2) - \hat{\mathbf{f}}_{loc}(i,j_1),$$

where $\hat{\mathbf{f}}_{loc}(i,j)$ is the feature's position in the local coordinate, given by

$$\hat{\mathbf{f}}_{loc}(i,j) = \hat{\mathbf{xf}}_{i|i-1,j} - \hat{\mathbf{xf}}_{i|i-1,1}.$$

Correspondingly, the actual observations are given by

$$\mathbf{z}_{feat}(i,j) = \mathbf{R}(\theta_{i-1|i-1})\delta(i,j) + noise_{feat},$$

$$\mathbf{z}_{cons}(i,j1,j2) = \mathbf{f}_{loc}(i,j_2) - \mathbf{f}_{loc}(i,j_1) + noise_{cons}$$

Update

Jacobian of the feature measurement model

$$\mathbf{H}_{feat} = \begin{bmatrix} -\mathbf{J}\hat{\mathbf{z}}_{feat}(i,j) & -\mathbf{R}(\hat{\theta}_{i-1|i-1})' & \mathbf{R}(\hat{\theta}_{i-1|i-1})' \end{bmatrix},$$

where $\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

Jacobian of the construction measurement model

$$\mathbf{H}_{cons} = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}. \tag{1}$$

The covariance matrix can be updated by

$$\mathbf{S} = \mathbf{H} * \mathbf{P_{i+1|i}} * \mathbf{H'} + \mathbf{N},$$

$$\mathbf{K} = \mathbf{P_{i+1|i}} * \mathbf{H'} * \mathbf{S^{-1}},$$

$$\mathbf{P_{i+1|i+1}} = \mathbf{P_{i+1|i}} - \mathbf{KSK'},$$

where $\mathbf{N} = diag(\sigma_{Feat})$.

The state vector can be update by

$$\mathbf{X_{i+1|i+1}} = \mathbf{X_{i+1|i}} + \mathbf{K}(\mathbf{Z} - \hat{\mathbf{Z}})$$

Simulation

We set four groups of experiments, including:

1) straight feature path with fixed observation noise;

2) straight feature path with changing observation noise;

3) circle feature path with fixed observation noise;

4) circle feature path with changing observation noise.

Straight feature path case means that, the features can only move in the x direction in each step, while circle feature path means they can move along both x and y axis along a circle.

In the case with fixed observation noise, σ_{feat} is fixed to be 0.005 in each step.

In the case with changing observation noise, σ_{feat} is changing with distance between the robot and the feature. Concretely, within the sensor range, when the distance gets larger, σ_{feat} gets larger. It is reasonable, because in the real world, the uncertainty is larger when the robot is far from the feature.

Here, the noise model is a simple linear model. The minimum value of σ_{feat} is 0.001 at $range = 0.01$, and the maximum value is 0.009 at $range = 0.2$.

Except $noise_{feat}$, other noises are fixed and the same in all experiments. Concretely, $\sigma_{head} = 0.001$, $\sigma_{odem} = 0.002$, $\sigma_{smoo} = 0.03$, and $\sigma_{cons} = 0.02$.

In our experiment, we have three features, and the initial feature positions are (0.05, 0.05), (0.09, 0.08), (0.03, 0.04) respectively.

We compare the results of using a predetermined path and a path planned by the active SLAM algorithm in each case. For each experiment, the robot moves 20 steps, then we compare the trace of the covariance matrix, $\mathrm{trace}(\mathbf{P})$, and the total number of features that undetected by the robot, num_unseen.

Straight feature path

In this case, the translation of the features are fixed to be 0.2m along the x-axis.

When using a predetermined path, the odometry of the robot is also fixed to be 0.2m along the x-axis.

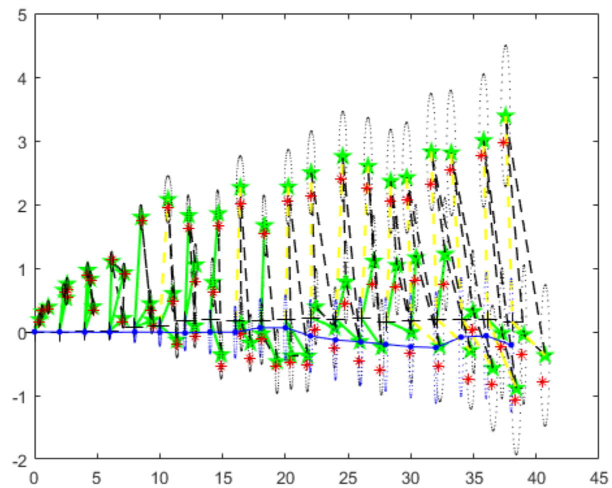
The results are as Fig. 1 and Fig. 2 shows.

Circle feature path

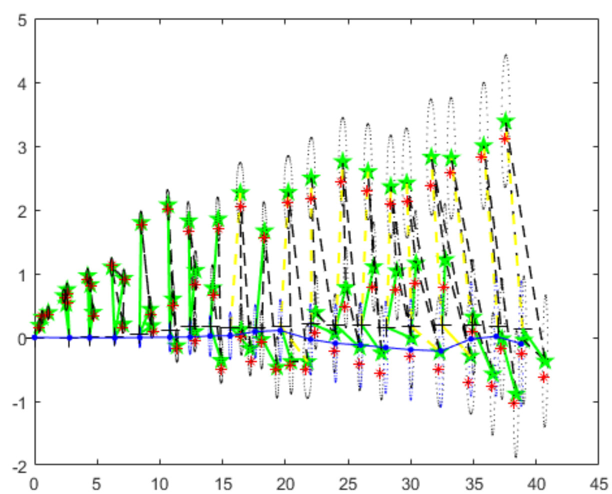
In this case, the features travel along a circle at a fixed travel distance of 0.2m.

Correspondingly, the robot also travels along such a circle as the features when using a predetermined path.

The results are as Fig. 3 and Fig. 4 shows.

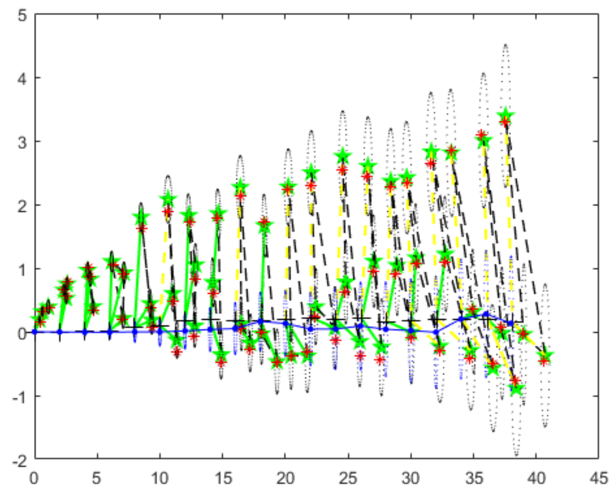


(a) Using a predetermined path (cm). $trace(P) = 0.1061, num\_unseen = 3$.

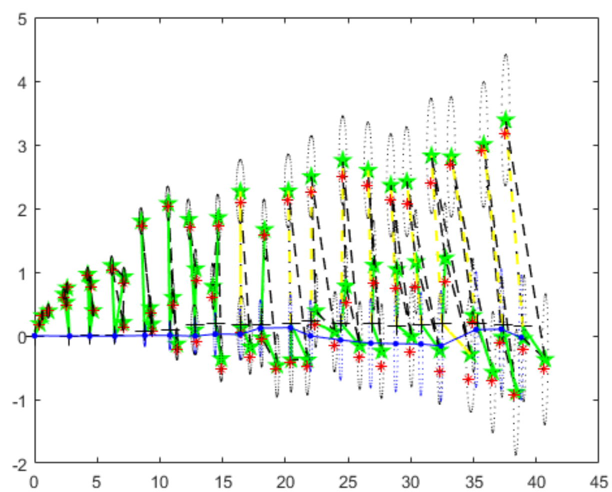


(b) Using active SLAM algorithm (cm). $trace(P) = 0.1061, num\_unseen = 3$.

Figure 1: Straight feature path with fixed observation noise.

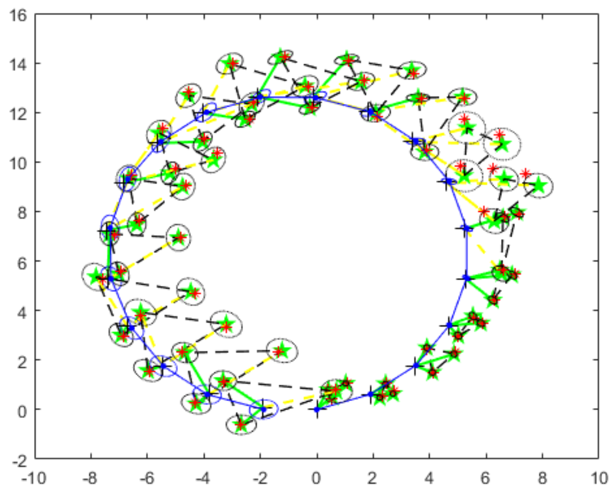


(a) Using a predetermined path (cm). $trace(P) = 0.1124, num\_unseen = 3.$

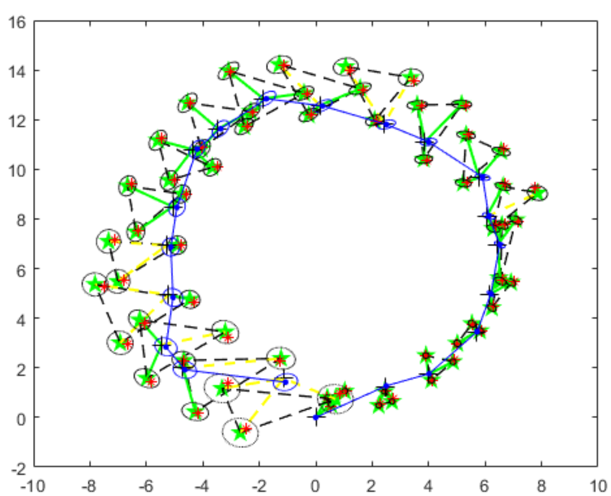


(b) Using active SLAM algorithm (cm). $trace(P) = 0.1124, num\_unseen = 3.$

Figure 2: Straight feature path with changing observation noise.

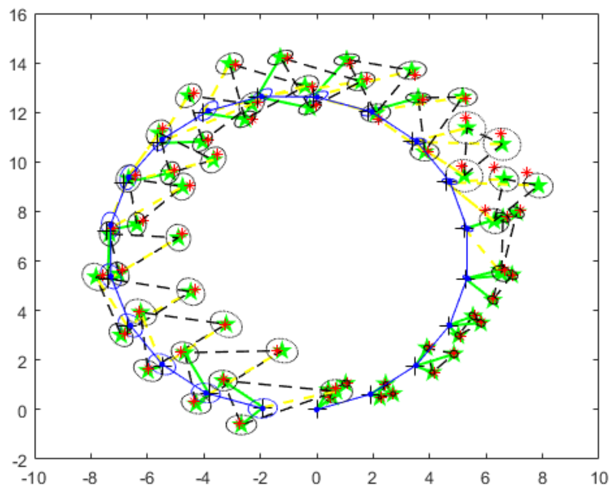


(a) Using a predetermined path (cm). $trace(P) = 0.1061, num\_unseen = 47.$

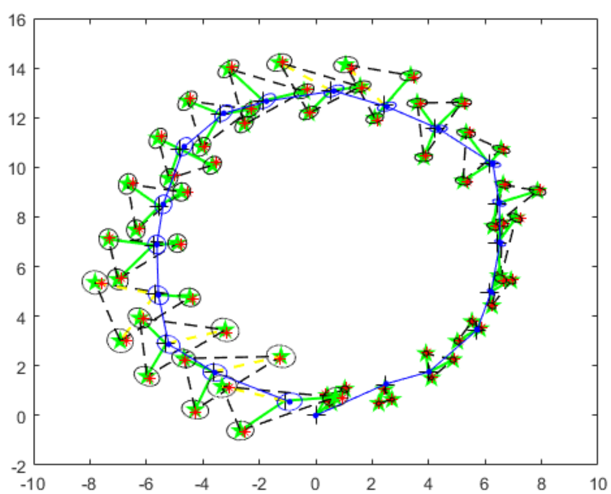


(b) Using active SLAM algorithm (cm). $trace(P) = 0.1063, num\_unseen = 11.$

Figure 3: Circle feature path with fixed observation noise.



(a) Using a predetermined path (cm). $trace(P) = 0.1068, num\_unseen = 47.$



(b) Using active SLAM algorithm (cm). $trace(P) = 0.1034, num\_unseen = 1.$

Figure 4: Circle feature path with changing observation noise.