2D NLS 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 $\mathbf{X} = [\mathbf{Xr}; \mathbf{Xf}]$, where $\mathbf{Xr} = [\theta_1; \mathbf{xr}_1; ...; \theta_N; \mathbf{xr}_N]$, and $\mathbf{Xf} = [\mathbf{xf}_1; ...; \mathbf{xf}_N]$.

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,

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

Introduction

In this part, we first collect all the measurement data, and then use non-linear least square method to optimize the estimation of the state vector.

The objective is to minimize the estimation error, which contains the odometry error, $\mathbf{e}_{odem}$, the feature measurement error, $\mathbf{e}_{feat}$, the construction measurement error, $\mathbf{e}_{cons}$, and the smooth measurement error, $\mathbf{e}_{smoo}$.

Objective function

$$\begin{aligned}\min F_{obj} = &\sum_i \mathbf{e}_{odem}(i)^T \mathbf{P}_{odem}^{-1} \mathbf{e}_{odem}(i) + \\ &\sum_i \sum_j \mathbf{e}_{feat}(i,j)^T \mathbf{P}feat^{-1} \mathbf{e}_{feat}(i,j)^{-1} + \\ &\sum_i \sum_{j1} \sum_{j2} \mathbf{e}_{cons}(i,j1,j2)^T \mathbf{P}_{cons}^{-1} \mathbf{e}_{cons}(i,j1,j2)^{-1} + \\ &\sum_i \sum_{i+1} \sum_j \mathbf{e}_{smoo}(i,i+1,j)^T \mathbf{P}_{smoo}^{-1} \mathbf{e}_{smoo}(i,i+1,j)^{-1}.\end{aligned}$$

We use Gauss-Newton algorithm to solve the function. In each iteration of the GN algorithm, the incremental $\Delta\mathbf{x}$ can be obtained by solving the normal equation:

$$\mathbf{H}\Delta\mathbf{x} = \mathbf{g},$$

where $\mathbf{H} = \mathbf{J}^T\mathbf{P}^{-1}\mathbf{J}$ and $\mathbf{g} = -\mathbf{J}^T\mathbf{P}^{-1}\mathbf{F}$.

Here, $\mathbf{J}$ is the jacobian of the error model, and the $\mathbf{F}$ is the value of the error. The calculation of them will be given in the following sections.

The initial value of the optimization is obtained by the predict model as the EKF method dose.

Odometry Error

$$\mathbf{e}_{odem}(i) = \mathbf{Xr}_{i+1} - \hat{\mathbf{Xr}}_{i+1|i}.$$

Here, $\hat{\mathbf{Xr}}_{i+1|i} = [\hat{\theta}_{i+1|i}; \hat{\mathbf{xr}}_{i+1|i}]$ can be obtained by

$$\hat{\theta}_{i+1|i} = \hat{\theta}_{i|i} + \omega_i,$$

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

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

Jacobian of the odometry error model

$$\mathbf{J}_{odem}(i, i+1) = \left[\frac{\mathrm{d}\mathbf{e}_{odem}(i)}{\mathrm{d}\hat{\mathbf{xr}}_{i|i}} \; \frac{\mathrm{d}\mathbf{e}_{odem}(i)}{\mathrm{d}\hat{\mathbf{xr}}_{i+1|i+1}}\right] = \begin{bmatrix} -1 & \mathbf{0}_{1,2} & 1 & \mathbf{0}_{1,2} \\ -\mathbf{R}(\hat{\theta}_{i|i})\mathbf{J}\mathbf{v}_i^T & -\mathbf{I}_{2,2} & 0 & \mathbf{I}_{2,2} \end{bmatrix}$$

The corresponding covariance matrix $\mathbf{P}_{odem}(i) = \begin{bmatrix} \sigma^2_{head} & 0 & 0 \\ 0 & \sigma^2_{odem} & 0 \\ 0 & 0 & \sigma^2_{odem} \end{bmatrix}$.

Feature Measurement Error

$$\mathbf{e}_{feat}(i, j) = \mathbf{z}_{feat}(i, j) - \hat{\mathbf{z}}_{feat}(i, j)$$

Feature measurement estimation is given by

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

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

Jacobian of the feature measurement error model

$$\mathbf{J}_{feat}(i, j) = \left[\frac{\mathrm{d}\mathbf{e}_{feat}(i, j)}{\mathrm{d}\mathbf{xr}_{i|i-1}} \; \frac{\mathrm{d}\mathbf{e}_{feat}(i, j)}{\mathrm{d}\mathbf{xf}_{i|i-1,j}}\right] = -\begin{bmatrix} -\mathbf{J}\hat{\mathbf{z}}_{feat}(i, j) & -\mathbf{R}(\hat{\theta}_{i-1|i-1})' & \mathbf{R}(\hat{\theta}_{i-1|i-1})' \end{bmatrix},$$

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

The corresponding covariance matrix $\mathbf{P}_{feat}(i) = \begin{bmatrix} \sigma_{feat}(i)^2 & 0 \\ 0 & \sigma_{feat}(i)^2 \end{bmatrix}$.

Construction Measurement Error

$$\mathbf{e}_{cons}(i, j1, j2) = \mathbf{z}_{cons}(i, j1, j2) - \hat{\mathbf{z}}_{cons}(i, j1, j2)$$

Construction measurement estimation 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

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

Jacobian of the construction measurement error model

$$\mathbf{J}_{cons}(i, j1, j2) = \left[\frac{\mathrm{d}\mathbf{e}_{cons}(i, j1, j2)}{\mathrm{d}\mathbf{xf}_{i,j1}} \; \frac{\mathrm{d}\mathbf{e}_{cons}(i, j1, j2)}{\mathrm{d}\mathbf{xf}_{i|i-1,j2}}\right] = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}. \tag{1}$$

The corresponding covariance matrix $\mathbf{P}_{cons}(i) = \begin{bmatrix} \sigma^2_{cons} & 0 \\ 0 & \sigma^2_{cons} \end{bmatrix}$.

Smooth Measurement Error

$$\mathbf{e}_{smoo}(i, i+1, j) = \mathbf{z}_{smoo}(i, i+1, j) - \hat{\mathbf{z}}_{smoo}(i, i+1, j)$$

The smooth estimation of a feature between two adjacent time steps $i+1$ and i can be obtained by

$$\hat{\mathbf{z}}_{smoo}(i, i+1, j) = \hat{\mathbf{xf}}_{i+1|i+1,j} - \hat{\mathbf{xf}}_{i+1|i,j}.$$

Jacobian of the smooth measurement error model

$$\mathbf{J}_{smoo}(i, i+1, j) = \left[\frac{\mathrm{d}\mathbf{e}_{smoo}(i, i+1, j)}{\mathrm{d}\hat{\mathbf{xf}}_{i|i,j}} \; \frac{\mathrm{d}\mathbf{e}_{smoo}(i, i+1, j)}{\mathrm{d}\hat{\mathbf{xf}}_{i+1|i+1,j}}\right] = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}. \tag{2}$$

The corresponding covariance matrix $\mathbf{P}_{smoo}(i) = \begin{bmatrix} \sigma^2_{smoo} + \sigma^2_{cons} & 0 \\ 0 & \sigma^2_{smoo} + \sigma^2_{cons} \end{bmatrix}$.

Simulation

We set 4 groups of experiments as the EKF method dose, including:

1) straight path with fixed observation noise;

2) straight path with changing observation noise;

3) circle path with fixed observation noise;

4) circle path with changing observation noise.

In the case with fixed observation noise, σ_{feat} is fixed to be 0.05 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 $range = 0.1$, and the maximum value is 0.09 at $range = 2$.

Except $noise_{feat}$, other noises are fixed and the same in all experiments. Concretely, $\sigma_{head} = 0.01$, $\sigma_{odem} = 0.02$, $\sigma_{smoo} = 0.3$, and $\sigma_{cons} = 0.2$.

In each group of experiment, the two methods, predetermined path and the active method, are used respectively. Therefore, we have 8 groups of experiment in total. But in the current note, I just compare the result of using predetermined path with fixed feature measurement noise.

We compare the results of using NLS method with using EKF method in each case in terms of the pose error.

The noises are the same for the two methods in each case.

In the experiment, we set $N = 20$ and $M = 3$.

Comparation of the robot pose error

The comparison of the robot pose error of all cases are shown in Table. 1.

The pose error of the NLS-based method is smaller than the EKF-based one in each case.

Table 1: Robot pose estimation error comparison

	EKF	***NLS***
straight + fixed Noise + pre	0.1629	0.1238
circle + fixed Noise + pre	0.1460	0.0616
straight + changing Noise + pre		
circle + changing Noise + pre		
straight + fixed Noise + active		
circle + fixed Noise + active		
straight + changing Noise + active		
circle + changing Noise + active		

Straight path

In the straight path case, the translation of the features are fixed to be 2m.

When using a predetermined path, the odometry of the robot is also fixed to be 2m.

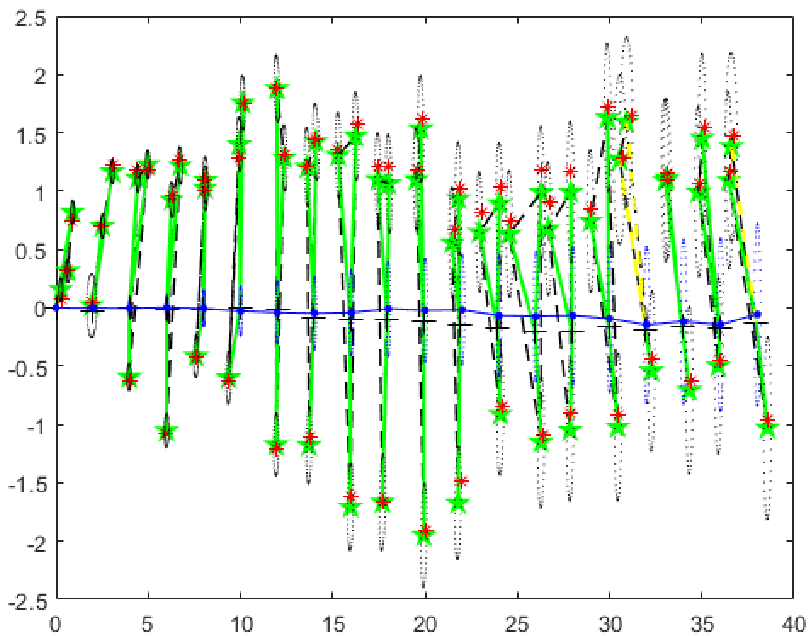
The results are as Fig. 1 shows.

Circle path

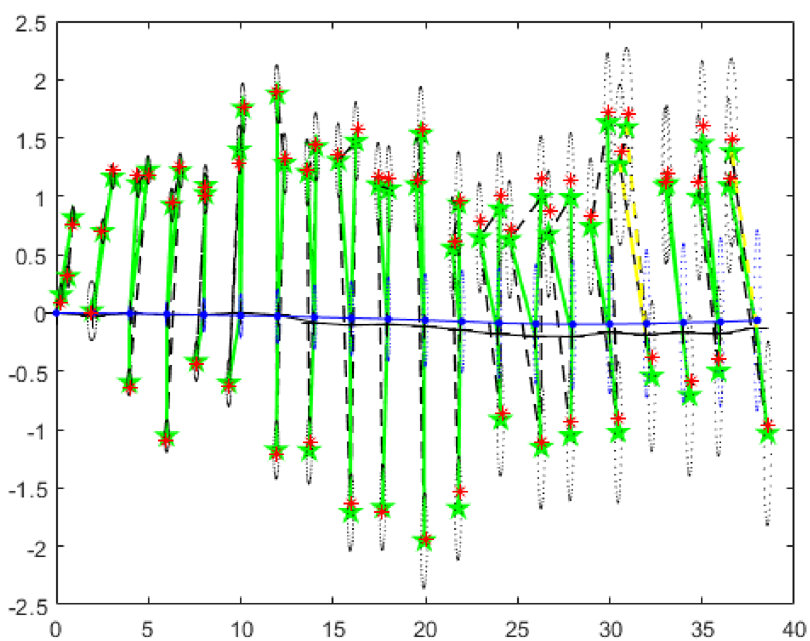
In the 2D case, the features travel along a circle with a fixed travel distance of 2m.

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

The results are as Fig. 2 shows.

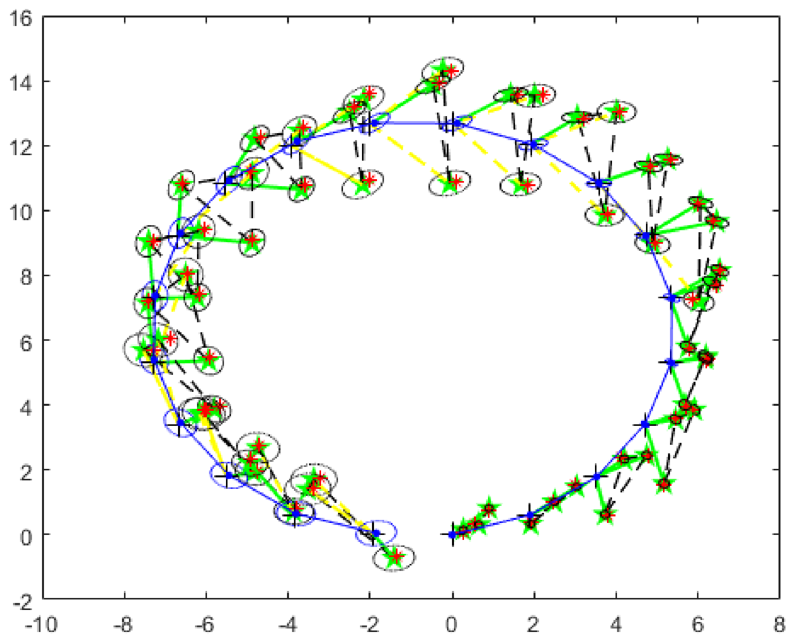


(a) EKF-based method.

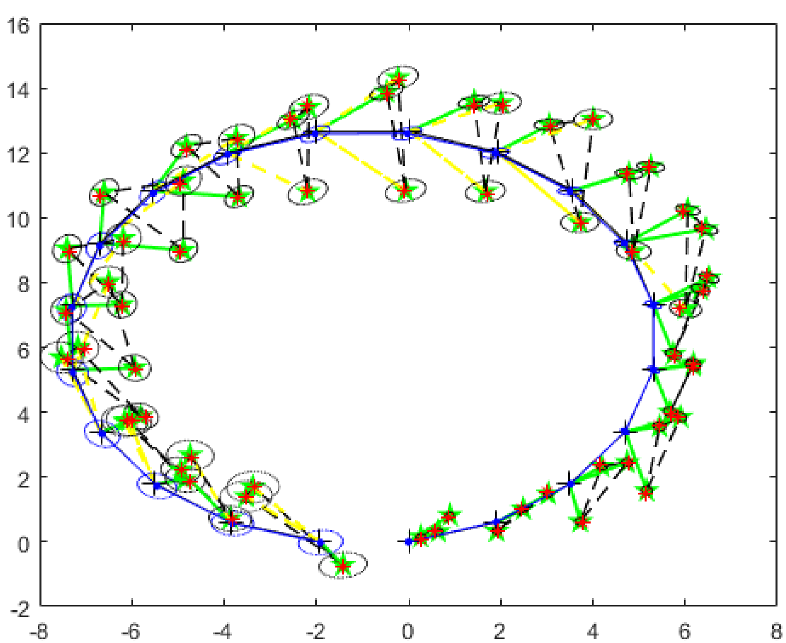


(b) NLS-based method.

Figure 1: Straight path with fixed observation noise using predetermined robot path.



(a) EKF-based method.



(b) NLS-based method.

Figure 2: Circle path with fixed observation noise using predetermined robot path.