

## Expected Results

Figure 1 depicts the result that you should obtain after convergence for each dataset. Additionally, the initial and the final error for each dataset should approximately take the values in the table below.

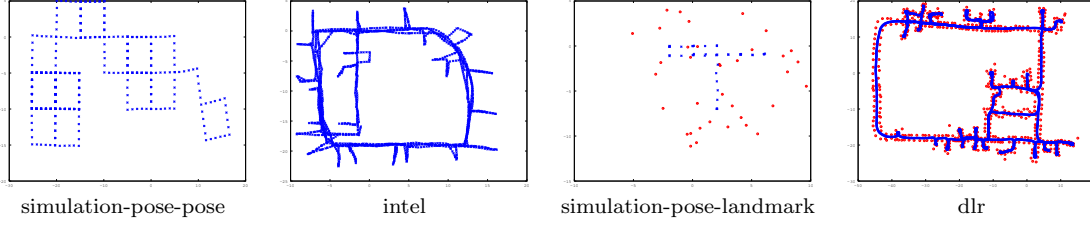


Figure 1: Result for each dataset.

dataset	initial error	final error
simulation-pose-pose.dat	138862234	8269
intel.dat	1795139	360
simulation-pose-landmark.dat	3030	474
dlr.dat	369655336	56860

## Hints

- The key elements in the graph datastructure  $\mathbf{g}$  are the node poses/locations stored in  $\mathbf{g.x}$  as well as the edge information  $\mathbf{g.edges}$  consisting of the measurement (**measurement**) between the two nodes and the corresponding information matrix (**information**).
- The state vector ( $\mathbf{g.x}$ ) consists of a concatenation of robot poses (and possibly landmark positions), i.e.,  $\mathbf{x}_{1:N}$  with

- pose of the robot:  $\mathbf{x}_i = (x_i \ y_i \ \theta_i)^T$

Hint: Use the function  $\mathbf{v2t}(\cdot)$  and  $\mathbf{t2v}(\cdot)$  that transform a vector into a transformation matrix ( $\mathbf{v2t}(\cdot)$ ) and vice versa ( $\mathbf{t2v}(\cdot)$ ). The function  $\mathbf{inv}(\cdot)$  inverts a transformation.

$$\mathbf{v2t}(\mathbf{x}_i) = \begin{pmatrix} R_i & \mathbf{t}_i \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) & x_i \\ \sin(\theta_i) & \cos(\theta_i) & y_i \\ 0 & 0 & 1 \end{pmatrix} = X_i \quad \mathbf{t2v}(X_i) = \mathbf{x}_i$$

- position of a landmark:  $\mathbf{x}_l = (x_l \ y_l)^T$

- We consider the following error functions:

- pose-pose constraint:  $\mathbf{e}_{ij} = \mathbf{t2v}(Z_{ij}^{-1}(X_i^{-1}X_j))$ , where  $Z_{ij} = \mathbf{v2t}(\mathbf{z}_{ij})$  is the transformation matrix of the measurement  $\mathbf{z}_{ij}^T = (\mathbf{t}_{ij}^T, \theta_{ij})$ .
- pose-landmark constraint:  $\mathbf{e}_{il}(\mathbf{x}) = X_i^{-1}\mathbf{x}_l - \mathbf{z}_{il}$

# Further Information on the Error Functions and Jacobians for 2D Graph-Based SLAM

In the following we will provide the definitions and the derivations for the Jacobians to implement 2D graph-based SLAM.

## Error Functions and Jacobians for a Pose-Pose Constraint

The basic entities of the poses of the robot and the constraint are defined as

$$\mathbf{x}_i^\top = (\mathbf{t}_i^\top, \theta_i) = (x_i, y_i, \theta_i) \quad (1)$$

$$\mathbf{x}_j^\top = (\mathbf{t}_j^\top, \theta_j) = (x_j, y_j, \theta_j) \quad (2)$$

$$\mathbf{z}_{ij}^\top = (\mathbf{t}_{ij}^\top, \theta_{ij}) = (x_{ij}, y_{ij}, \theta_{ij}) \quad (3)$$

where  $\mathbf{t}_i$  and  $\mathbf{t}_{ij}$  are 2D vectors and  $\theta_i$  and  $\theta_{ij}$  are rotation angles which are normalized to  $[-\pi, \pi)$ .

### Error Function

The error function is

$$\mathbf{e}_{ij}(\mathbf{x}) = \text{t2v}(Z_{ij}^{-1}(X_i^{-1}X_j)) \quad (4)$$

### Jacobian

To derive the Jacobian, it is advantageous to start from:

$$\mathbf{e}_{ij}(\mathbf{x}) = \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \begin{pmatrix} R_{ij}^\top(R_i^\top(\mathbf{t}_j - \mathbf{t}_i) - \mathbf{t}_{ij}) \\ \theta_j - \theta_i - \theta_{ij} \end{pmatrix}, \quad (5)$$

where  $R_i$  and  $R_{ij}$  are the  $2 \times 2$  rotation matrices of  $\theta_i$  and  $\theta_{ij}$  with the following structure

$$R_i = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{pmatrix} \quad R_{ij} = \begin{pmatrix} \cos(\theta_{ij}) & -\sin(\theta_{ij}) \\ \sin(\theta_{ij}) & \cos(\theta_{ij}) \end{pmatrix}. \quad (6)$$

The Jacobians of the error function become

$$\mathbf{A}_{ij} = \frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_i} = \begin{pmatrix} -R_{ij}^\top R_i^\top & R_{ij}^\top \frac{\partial R_i^\top}{\partial \theta_i}(\mathbf{t}_j - \mathbf{t}_i) \\ \mathbf{0} & -1 \end{pmatrix} \quad (7)$$

$$\mathbf{B}_{ij} = \frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_j} = \begin{pmatrix} R_{ij}^\top R_i^\top & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}. \quad (8)$$

And for  $\frac{\partial R_i^\top}{\partial \theta_i}$  we obtain

$$\frac{\partial R_i^\top}{\partial \theta_i} = \begin{pmatrix} -\sin(\theta_i) & \cos(\theta_i) \\ -\cos(\theta_i) & -\sin(\theta_i) \end{pmatrix}. \quad (9)$$

## Error Functions and Jacobians for a Pose-Landmark Constraint

The basic entities of the poses of the robot and the constraint are defined as

$$\mathbf{x}_i^\top = (\mathbf{t}_i^\top, \theta_i) \quad (10)$$

$$\mathbf{x}_l^\top = (x_l, y_l)^\top \quad (11)$$

$$\mathbf{z}_{il} = (x_{il}, y_{il})^\top, \quad (12)$$

where  $\mathbf{t}_i$  is a 2D vector,  $\theta_i$  is a rotation angle which is normalized to  $[-\pi, \pi)$ , and  $(x_{il}, y_{il})^\top$  is the measurement of  $\mathbf{x}_l$  in the local frame of  $\mathbf{x}_i$ .

### Error Function

The error function is

$$\mathbf{e}_{il}(\mathbf{x}) = X_i^{-1}\mathbf{x}_l - \mathbf{z}_{il} \quad (13)$$

### Jacobian

To derive the Jacobian, it is advantageous to start from:

$$\mathbf{e}_{il}(\mathbf{x}) = R_i^\top(\mathbf{x}_l - \mathbf{t}_i) - \mathbf{z}_{il}, \quad (14)$$

where  $R_i$  is the  $2 \times 2$  rotation matrices of  $\theta_i$  with the following structure

$$R_i = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{pmatrix} \quad (15)$$

The Jacobians of the error function becomes

$$\mathbf{A}_{il} = \frac{\partial \mathbf{e}_{il}(\mathbf{x})}{\partial \mathbf{x}_i} = \left( -R_i^\top \quad \frac{\partial R_i^\top}{\partial \theta_i}(\mathbf{x}_l - \mathbf{t}_i) \right) \quad (16)$$

$$\mathbf{B}_{il} = \frac{\partial \mathbf{e}_{il}(\mathbf{x})}{\partial \mathbf{x}_j} = R_i^\top \quad (17)$$

And for  $\frac{\partial R_i^\top}{\partial \theta_i}$  we obtain

$$\frac{\partial R_i^\top}{\partial \theta_i} = \begin{pmatrix} -\sin(\theta_i) & \cos(\theta_i) \\ -\cos(\theta_i) & -\sin(\theta_i) \end{pmatrix}. \quad (18)$$