1D GraphSLAM Examples

Example One: Suppose a robot is initially at 0. It moves forward for one meter (commanded and/or measured by odometry), and then moves backward for 0.8 meters. However, it then finds itself to be at the original position. So the question is where has the robot been, the initial, after one move and after two moves?

The set of constraints are:

\[
x_0 = 0 \quad \text{initial condition}
\]
\[
x_1 = x_0 + 1 \quad 1 \text{ forward}
\]
\[
x_2 = x_1 - 0.8 \quad 0.8 \text{ backward}
\]
\[
x_2 = x_0 \quad \text{loop closure}
\]

Equivalently,

\[
f_1 = x_0 = 0
\]
\[
f_2 = x_1 - x_0 - 1 = 0
\]
\[
f_3 = x_2 - x_1 + 0.8 = 0
\]
\[
f_4 = x_2 - x_0 = 0
\]

The cost function in this case is:

\[
c = \sum_{i=1}^{4} f_i^2 = x_0^2 + (x_1 - x_0 - 1)^2 + (x_2 - x_1 + 0.8)^2 + (x_2 - x_0)^2
\]
Then we take the partial derivatives of the cost function with respect to the unknown variables \(x_i\), set the partial derivatives to zero, and then solve to find the optimal solution that gives the minimum cost.

\[
\begin{align*}
\frac{\partial c}{\partial x_0} &= 2x_0 - 2(x_1 - x_0 - 1) - 2(x_2 - x_0) = 0 \\
\frac{\partial c}{\partial x_1} &= 2(x_1 - x_0 - 1) - 2(x_2 - x_1 + 0.8) = 0 \\
\frac{\partial c}{\partial x_2} &= 2(x_2 - x_1 + 0.8) + 2(x_2 - x_0) = 0
\end{align*}
\]

Manipulating the above equations we get:

\[
\begin{align*}
3x_0 - x_1 - x_2 &= -1 \\
-x_0 + 2x_1 - x_2 &= 1.8 \\
-x_0 - x_1 + 2x_2 &= -0.8
\end{align*}
\]

Note that by construction, there are exactly as many linear equations as there are unknowns, which we can solve by inverting a matrix and multiplying, i.e.

\[
\begin{bmatrix}
3 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
x_2
\end{bmatrix}
= \begin{bmatrix}
-1.0 \\
1.8 \\
-0.8
\end{bmatrix}
\text{ or } \Omega \mu = \xi
\]

\[
\mu = \Omega^{-1}\xi = \begin{bmatrix}
3 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2
\end{bmatrix}^{-1}
\begin{bmatrix}
-1.0 \\
1.8 \\
-0.8
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & 1 \\
1 & 5/3 & 4/3 \\
1 & 4/3 & 5/3
\end{bmatrix}
\begin{bmatrix}
-1.0 \\
1.8 \\
-0.8
\end{bmatrix}
= \begin{bmatrix}
0 \\
0.93 \\
0.07
\end{bmatrix}
\]

Therefore \(x_0 = 0, x_1 = 0.93, \text{ and } x_2 = 0.07\).

**Example Two:** Suppose a robot is initially at location \(x_0 = 0\) where it observes a landmark at a distance of 2 in the positive direction. Then it moves forward for 1 meter, and observes the same landmark at a distance of only 0.8. The question is: where is the robot and where is the landmark?

The set of constraints are:

\[
\begin{align*}
x_0 &= 0 & \text{ initial condition} \\
l_0 &= x_0 + 2 & \text{ first observation} \\
x_1 &= x_0 + 1 & \text{ 1 forward} \\
l_0 &= x_1 + 0.8 & \text{ second observation}
\end{align*}
\]
Equivalently,

\[ f_1 = x_0 = 0 \]
\[ f_2 = x_1 - x_0 - 1 = 0 \]
\[ f_3 = l_0 - x_0 - 2 = 0 \]
\[ f_4 = l_0 - x_1 - 0.8 = 0 \]

Define our cost function as the sum of the residual errors:

\[ c = \sum_{i=1}^{4} f_i^2 = x_0^2 + (x_1 - x_0 - 1)^2 + (l_0 - x_0 - 2)^2 + (l_0 - x_1 - 0.8)^2 \]

Then we take the partial derivatives of the cost function with respect to the unknown variables \( x_0, x_1 \) and \( l_0 \), set the partial derivatives to zero, and then solve to find the optimal solution that gives the minimum cost.

\[ \frac{\partial c}{\partial x_0} = 2x_0 - 2(x_1 - x_0 - 1) - 2(l_0 - x_0 - 2) = 0 \]
\[ \frac{\partial c}{\partial x_1} = 2(x_1 - x_0 - 1) - 2(l_0 - x_1 - 0.8) = 0 \]
\[ \frac{\partial c}{\partial l_0} = 2(l_0 - x_0 - 2) + 2(l_0 - x_1 - 0.8) = 0 \]

Manipulating the above equations we get:

\[ 3x_0 - x_1 - l_0 = -3 \]
\[ -x_0 + 2x_1 - l_0 = 0.2 \]
\[ -x_0 - x_1 + 2l_0 = 2.8 \]
Expressing the above into the matrix form, we get

\[
\begin{bmatrix}
3 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
l_0
\end{bmatrix}
= \begin{bmatrix}
-3.0 \\
0.2 \\
2.8
\end{bmatrix}
\quad \text{or} \quad \Omega \mu = \xi
\]

\[\mu = \Omega^{-1} \xi = \begin{bmatrix}
3 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2
\end{bmatrix}^{-1}
\begin{bmatrix}
-3.0 \\
0.2 \\
2.8
\end{bmatrix}
= \begin{bmatrix}
0 \\
1.07 \\
1.93
\end{bmatrix}
\]

Therefore \(x_0 = 0, x_1 = 1.07, \text{ and } l_0 = 1.93\).

If for some reason, you believe that the robot odometry is quite accurate, we can give a higher weight, say 10, to \(f_2\), so that the cost function becomes:

\[c = \sum_{i=1}^{4} f_i^2 = x_0^2 + 10(x_1 - x_0 - 1)^2 + (x_2 - x_1 + 0.8)^2 + (x_2 - x_0)^2\]

This translates into solving the following set of linear equations:

\[
\begin{align*}
12x_0 - 10x_1 - l_0 &= -12 \\
-10x_0 + 11x_1 - l_0 &= 9.2 \\
-x_0 - x_1 + 2l_0 &= 2.8
\end{align*}
\]

or in matrix form:

\[
\begin{bmatrix}
12 & -10 & -1 \\
-10 & 11 & -1 \\
-1 & -1 & 2
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
l_0
\end{bmatrix}
= \begin{bmatrix}
-12.0 \\
9.2 \\
2.8
\end{bmatrix}
\]

The solution to the above is \(x_0 = 0, x_1 = 1.01 \text{ and } l_0 = 1.9\). As expected, the constraint on \(x_1\) is more tightly satisfied than that on \(l_0\).