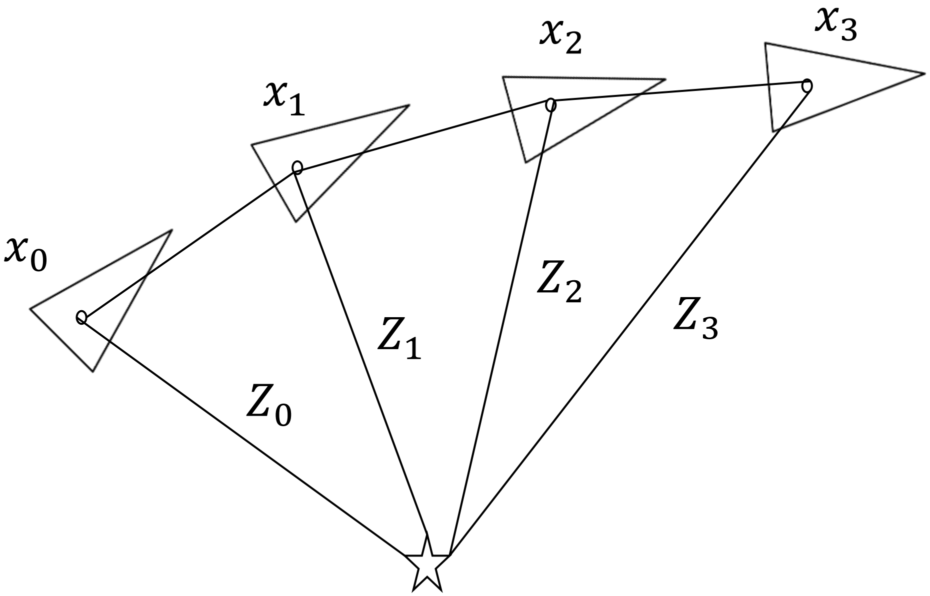
Mobile Robotics

Robot Motion:
SLAM

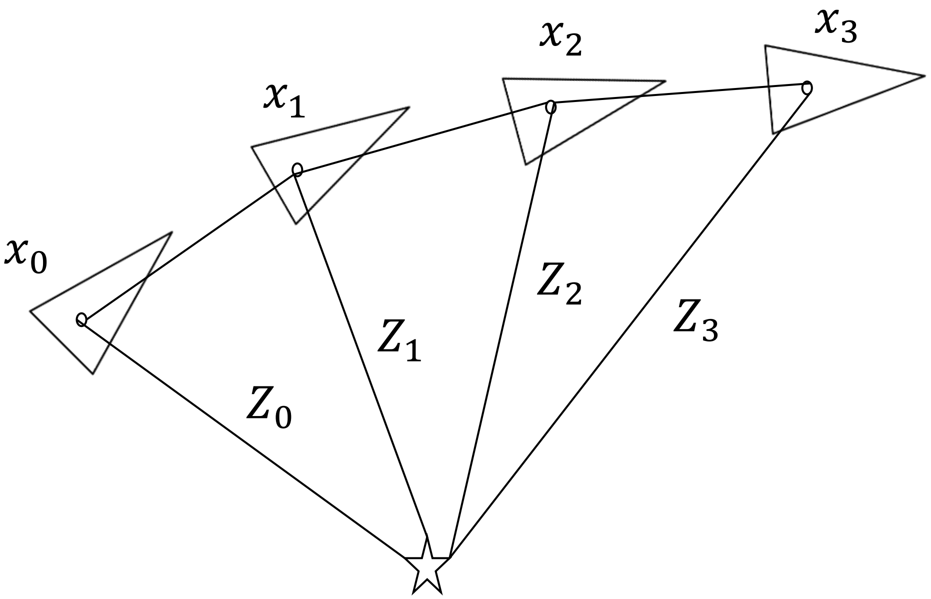
SLAM

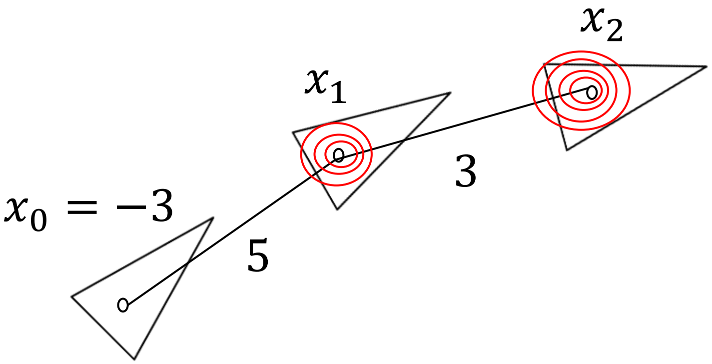
- Simultaneously Localization And Mapping
- When mapping an environment with a mobile robot, uncertainty in robot motion forces us to also perform robot localization.
- Type of Maps: Landmark-based or Grid-based



Graph SLAM

- Construct local constraints
  - Initial Location Constraint
  - Relative Motion Constraints
  - Relative Measurement Constraints.
- Maximize the probabilities of each constraints





$$p(x_1) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{(x_1 - x_0 - 5)^2}{\sigma^2}}$$

$$p(x_2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{(x_2 - x_1 - 3)^2}{\sigma^2}}$$

Implementing constraints

- Assume robot moving along one-dimensional space in which there are some landmarks.
- Assume $x_0 \rightarrow x_1 \rightarrow x_2$ and $x_1 = x_0 + 5$ and $x_2 = x_1 - 4$

$$
\begin{array}{c|ccccc}
 & x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
x_0 & 1 & -1 & & & \\
x_1 & -1 & 1 & & & \\
x_2 & & & & & \\
L_0 & & & & & \\
L_1 & & & & &
\end{array}
\quad
\begin{array}{|c|}
\hline
-5 \\
5 \\
\\
\\
\\
\hline
\end{array}
\quad + \quad
\begin{array}{ccccc}
x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
 & & & & \\
 & 1 & -1 & & \\
 & -1 & 1 & & \\
 & & & & \\
 & & & &
\end{array}
\quad
\begin{array}{|c|}
\hline
\\
4 \\
-4 \\
\\
\\
\hline
\end{array}
\quad \Rightarrow \quad
\begin{array}{ccccc}
x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
1 & -1 & & & \\
-1 & 2 & -1 & & \\
 & -1 & 1 & & \\
 & & & & \\
 & & & &
\end{array}
\quad
\begin{array}{|c|}
\hline
-5 \\
9 \\
-4 \\
\\
\\
\hline
\end{array}
$$

$$x_0 - x_1 = -5$$

$$-x_0 + x_1 = 5$$

$$x_1 - x_2 = 4$$

$$-x_1 + x_2 = -4$$

Adding Landmarks

- Assume robot moving along one-dimensional space in which there are some landmarks.
- Assume $x_0 \rightarrow x_1 \rightarrow x_2$ and $x_1 = x_0 + 5$ and $x_2 = x_1 - 4$
- x_1 to L_0: Distance is 9

$$
\begin{array}{c|c|c|c|c|c|}
 & x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
x_0 & 1 & -1 & & & \\
\hline
x_1 & -1 & 2 & -1 & & \\
\hline
x_2 & & -1 & 1 & & \\
\hline
L_0 & & & & & \\
\hline
L_1 & & & & & \\
\hline
\end{array}
\quad
\begin{array}{|c|}
\hline
-5 \\
\hline
9 \\
\hline
-4 \\
\hline
\\
\hline
\\
\hline
\end{array}
\quad + \quad
\begin{array}{|c|c|c|c|c|}
x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
 & & & & \\
\hline
 & 1 & & -1 & \\
\hline
 & & & & \\
\hline
 & -1 & & 1 & \\
\hline
 & & & & \\
\hline
\end{array}
\quad
\begin{array}{|c|}
\hline
\\
\hline
-9 \\
\hline
\\
\hline
9 \\
\hline
\\
\hline
\end{array}
\quad \Rightarrow \quad
\begin{array}{|c|c|c|c|c|}
x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
1 & -1 & & & \\
\hline
-1 & 3 & -2 & -1 & \\
\hline
 & -1 & 1 & & \\
\hline
 & -1 & & 1 & \\
\hline
 & & & & \\
\hline
\end{array}
\quad
\begin{array}{|c|}
\hline
-5 \\
\hline
0 \\
\hline
-4 \\
\hline
9 \\
\hline
\\
\hline
\end{array}
$$

$$x_1 - L_0 = -9$$

$$-x_1 + L_0 = 9$$

Adding Landmarks

- Assume robot moving along one-dimensional space in which there are some landmarks.
- Assume $x_0 \rightarrow x_1 \rightarrow x_2$ and $x_1 = x_0 + 5$ and $x_2 = x_1 - 4$
- x_1 to L_1: Distance is 12

$$
\begin{array}{c|ccccc|}
 & x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
x_0 & 1 & -1 & & & \\
x_1 & -1 & 2 & -1 & & \\
x_2 & & -1 & 1 & & \\
L_0 & & & & & \\
L_1 & & & & & \\
\hline
\end{array}
\quad
\begin{array}{|c|}
\hline
-5 \\ 9 \\ -4 \\ \\ \\
\hline
\end{array}
\quad + \quad
\begin{array}{|ccccc|}
x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
 & & & & \\
 & 1 & & & -1 \\
 & & & & \\
 & & & & \\
 & -1 & & & 1 \\
\hline
\end{array}
\quad
\begin{array}{|c|}
\hline
 \\ -12 \\ \\ 12 \\ \\
\hline
\end{array}
\quad \Rightarrow \quad
\begin{array}{|ccccc|}
x_0 & x_1 & x_2 & L_0 & L_1 \\
\hline
1 & -1 & & & \\
-1 & 3 & -2 & -1 & \\
 & -1 & 1 & & \\
 & -1 & & 1 & \\
 & & & & \\
\hline
\end{array}
\quad
\begin{array}{|c|}
\hline
-5 \\ 0 \\ -4 \\ 9 \\ \\
\hline
\end{array}
$$

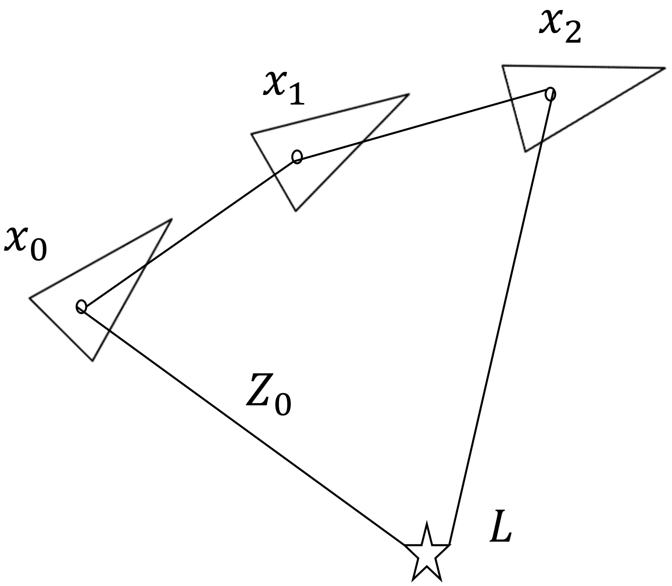
$$x_1 - L_1 = 12$$

$$-x_1 + L_1 = 12$$

Graph SLAM Questions

- True or False?
  - Graph SLAM is all about local constraints
  - They require multiplications
  - They require additions
- How to represent initial constraint $x_0 = 0$?
- Which matrix cells need to be modified? If we have

	x_0	x_1	x_2	L	
x_0					
x_1					
x_2					
L					



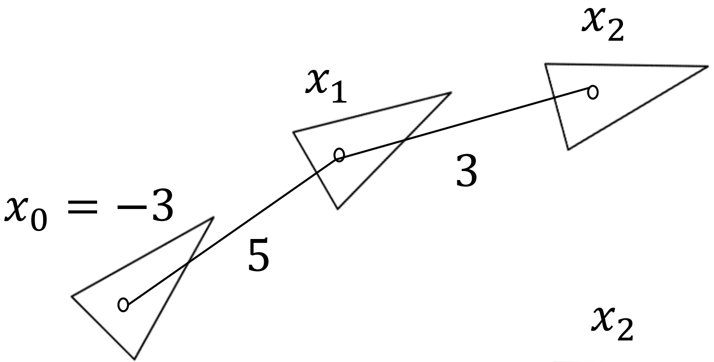
	x_0	x_1	x_2	L	
x_0					
x_1					
x_2					
L					

Omega, Xi, and Mu

- Let Ω be the matrix and ξ be the vector, then $\mu = \Omega^{-1}\xi$ gives the robot and landmark locations
- Example 1: compute $x_0 \; x_1 \; x_2$

$$x_0 = -3$$
$$x_1 = x_0 + 5$$
$$x_2 = x_1 + 3$$

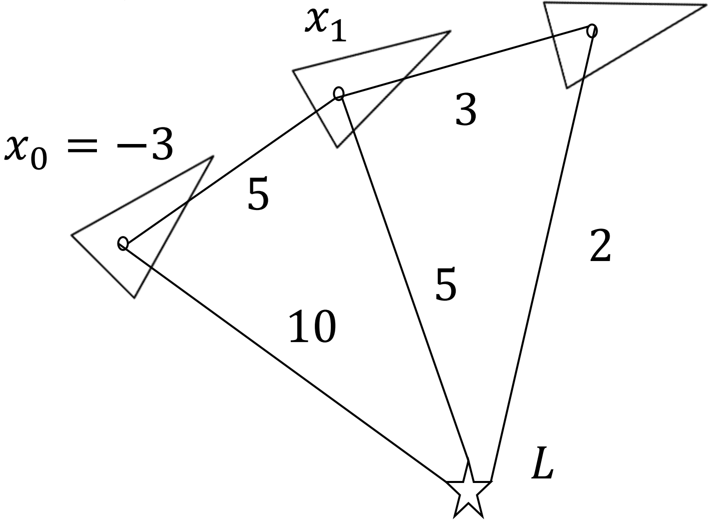
$$\begin{array}{c|c|c|c|} & x_0 & x_1 & x_2 \\ \hline x_0 & & & \\ \hline x_1 & & \Omega & \\ \hline x_2 & & & \\ \hline \end{array} \qquad \begin{array}{|c|} \hline \\ \hline \xi \\ \hline \\ \hline \end{array}$$



- Example 2: compute L

$$L = x_0 + 10$$
$$L = x_1 + 5$$
$$L = x_2 + 2$$

$$\begin{array}{c|c|c|c|c|} & x_0 & x_1 & x_2 & L \\ \hline x_0 & & & & \\ \hline x_1 & & \Omega & & \\ \hline x_2 & & & & \\ \hline L & & & & \\ \hline \end{array} \qquad \begin{array}{|c|} \hline \\ \hline \xi \\ \hline \\ \hline \\ \hline \end{array}$$



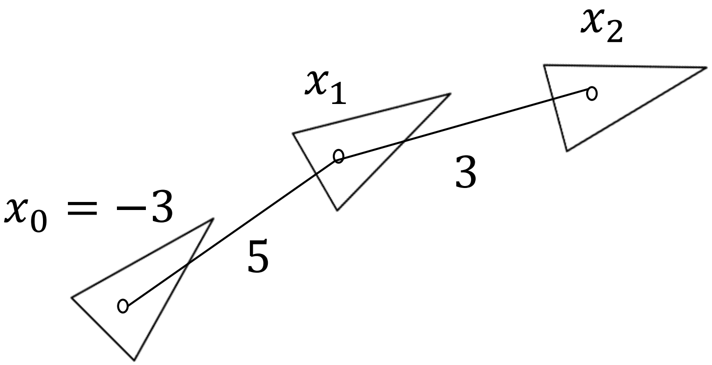
PA6A: Compute $\mu = \Omega^{-1}\xi$

- Write a function, doit, that takes as its input an initial robot position, move1, and move2.
- It should compute the Omega and Xi matrices, and
- It should return the mu vector

- Example 1: compute $x_0\ x_1\ x_2$

$$x_0 = -3$$
$$x_1 = x_0 + 5$$
$$x_2 = x_1 + 3$$

$$\begin{array}{c|c|c|c|} & x_0 & x_1 & x_2 \\ \hline x_0 & & & \\ \hline x_1 & & \Omega & \\ \hline x_2 & & & \\ \hline \end{array} \qquad \begin{array}{|c|} \hline \\ \hline \xi \\ \hline \\ \hline \end{array}$$

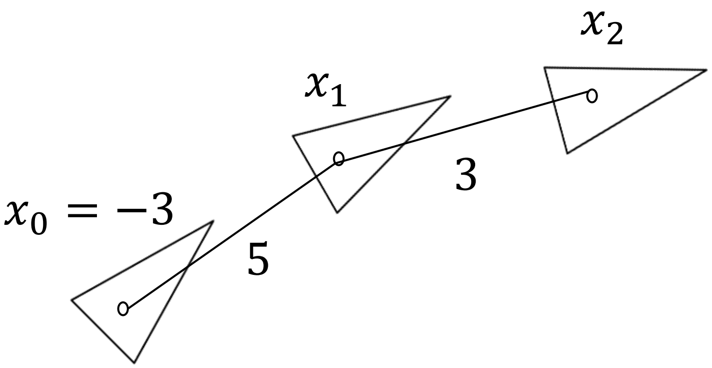


PA6A: Compute $\mu = \Omega^{-1}\xi$

- Example 1: compute x_0 x_1 x_2

$$x_0 = -3$$
$$x_1 = x_0 + 5$$
$$x_2 = x_1 + 3$$

$$\begin{array}{c|c|c|c|} & x_0 & x_1 & x_2 \\ \hline x_0 & & & \\ \hline x_1 & & \Omega & \\ \hline x_2 & & & \\ \hline \end{array} \qquad \begin{array}{|c|} \hline \\ \hline \xi \\ \hline \\ \hline \end{array}$$



$$\Omega = \begin{array}{c|c|c|c|} & x_0 & x_1 & x_2 \\ \hline x_0 & 1 & & \\ \hline x_1 & & & \\ \hline x_2 & & & \\ \hline \end{array} + \begin{array}{c|c|c|c|} & x_0 & x_1 & x_2 \\ \hline x_0 & 1 & -1 & \\ \hline x_1 & -1 & 1 & \\ \hline x_2 & & & \\ \hline \end{array} + \begin{array}{c|c|c|c|} & x_0 & x_1 & x_2 \\ \hline x_0 & & & \\ \hline x_1 & & 1 & -1 \\ \hline x_2 & & -1 & 1 \\ \hline \end{array}$$

$$\xi = \begin{array}{|c|} \hline -3 \\ \hline \\ \hline \\ \hline \end{array} + \begin{array}{|c|} \hline -5 \\ \hline 5 \\ \hline \\ \hline \end{array} + \begin{array}{|c|} \hline \\ \hline -3 \\ \hline 3 \\ \hline \end{array}$$

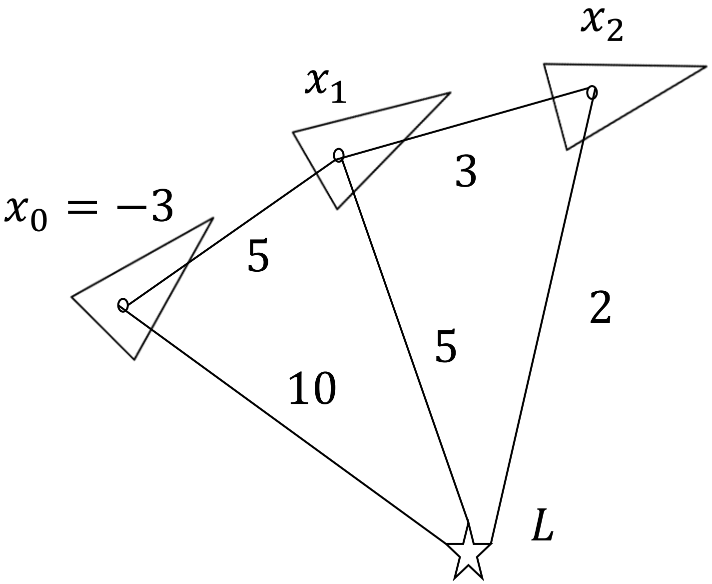
PA6B: Expand

- Modify your doit function to incorporate 3 distance measurements to a landmark (z_0, z_1, z_2).
- You should use the provided expand function to allow your Omega and Xi matrices to accommodate the new information.
- Each landmark measurement should modify 4 values in your Omega matrix and 2 in your Xi vector.

- Example 2: compute L

$$L = x_0 + 10$$
$$L = x_1 + 5$$
$$L = x_2 + 2$$

$$\begin{array}{c|c|c|c|c|}
 & x_0 & x_1 & x_2 & L \\ \hline
x_0 & & & & \\ \hline
x_1 & & \Omega & & \\ \hline
x_2 & & & & \\ \hline
L & & & & \\ \hline
\end{array}
\qquad
\begin{array}{|c|}
\hline
 \\ \hline
\xi \\ \hline
 \\ \hline
 \\ \hline
\end{array}$$



PA6B: Expand

- Example 2: compute L

$$L = x_0 + 10$$
$$L = x_1 + 5$$
$$L = x_2 + 2$$

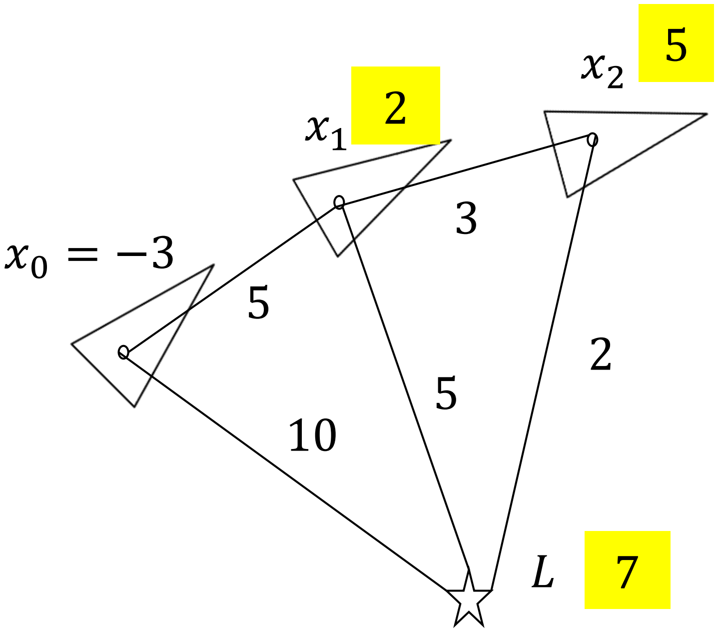
$$\begin{array}{c|c|c|c|c|}
 & x_0 & x_1 & x_2 & L \\ \hline
x_0 & & & & \\ \hline
x_1 & & \Omega & & \\ \hline
x_2 & & & & \\ \hline
L & & & & \\ \hline
\end{array}
\qquad
\begin{array}{|c|} \hline
\\ \hline
\xi \\ \hline
\\ \hline
\\ \hline
\end{array}$$

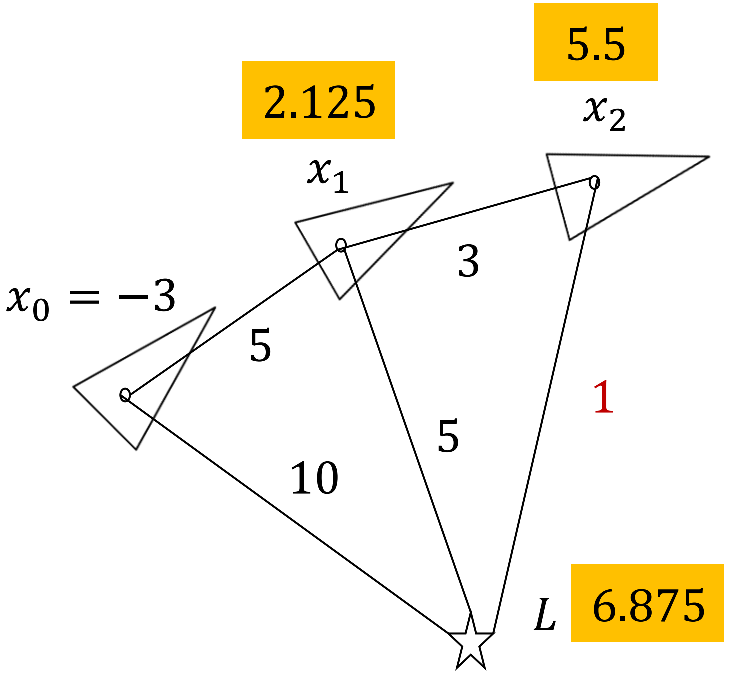
$$\Omega = \begin{array}{c|c|c|c|c|}
 & x_0 & x_1 & x_2 & L \\ \hline
x_0 & 2 & -1 & 0 & \\ \hline
x_1 & -1 & 2 & -1 & \\ \hline
x_2 & 0 & -1 & 1 & \\ \hline
L & & & & \\ \hline
\end{array}
+
\begin{array}{|c|c|c|c|}
x_0 & x_1 & x_2 & L \\ \hline
1 & & & -1 \\ \hline
 & & & \\ \hline
 & & & \\ \hline
-1 & & & 1 \\ \hline
\end{array}
+
\begin{array}{|c|c|c|c|}
x_0 & x_1 & x_2 & L \\ \hline
 & & & \\ \hline
 & 1 & & -1 \\ \hline
 & & & \\ \hline
 & -1 & & 1 \\ \hline
\end{array}
+
\begin{array}{|c|c|c|c|}
x_0 & x_1 & x_2 & L \\ \hline
 & & & \\ \hline
 & & & \\ \hline
 & & 1 & -1 \\ \hline
 & & -1 & 1 \\ \hline
\end{array}$$

$$\xi = \begin{array}{|c|} \hline
-8 \\ \hline
2 \\ \hline
3 \\ \hline
\\ \hline
\end{array}
+
\begin{array}{|c|} \hline
-10 \\ \hline
\\ \hline
\\ \hline
10 \\ \hline
\end{array}
+
\begin{array}{|c|} \hline
\\ \hline
-5 \\ \hline
\\ \hline
5 \\ \hline
\end{array}
+
\begin{array}{|c|} \hline
\\ \hline
\\ \hline
-2 \\ \hline
2 \\ \hline
\end{array}$$

Introducing Noise

- If the last measurement (from x_2 to L) down to 1, what will happen?
  - x_0 has no change
  - x_1 becomes larger
  - x_2 becomes larger
  - L becomes smaller



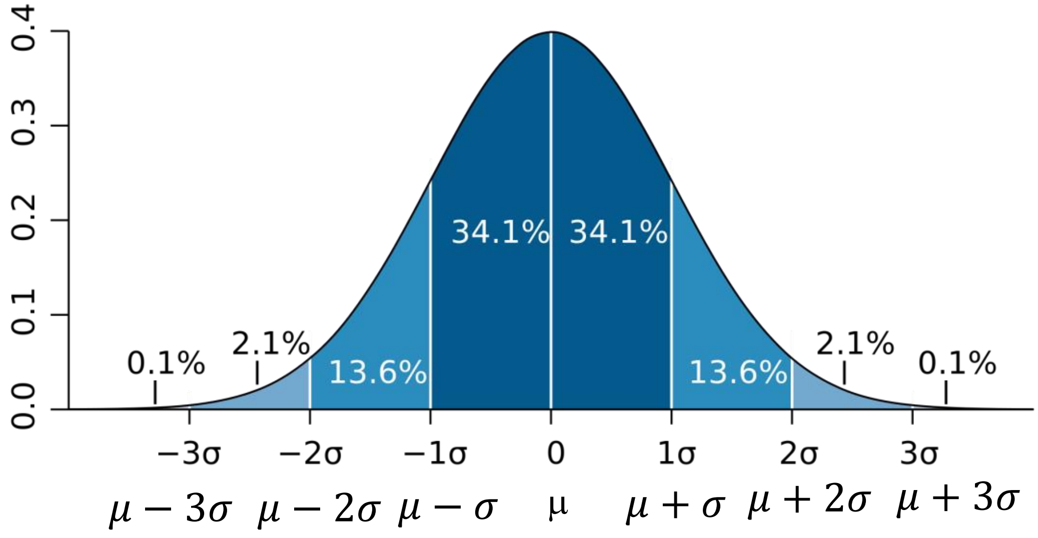


Recall: Gaussian (Normal) Distribution: Univariate

Univariate

$X \sim N(\mu, \sigma^2)$:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

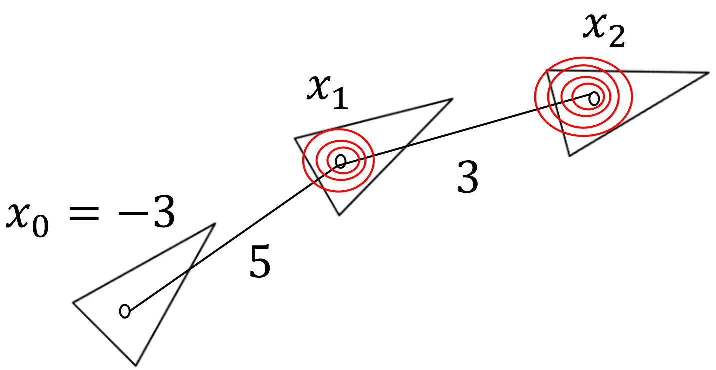


- μ: Mean
- σ^2: Variance
- σ: Standard Deviation

$$\mu = E(X)$$

$$\sigma^2 = E((X - \mu)^2)$$

Confident Measurements



$$p(x_1) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{(x_1 - x_0 - 5)^2}{\sigma^2}}$$

$$p(x_2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{(x_2 - x_1 - 3)^2}{\sigma^2}}$$

- We want to maximize $p(x_1)p(x_2)$
- This is equivalent to minimize $\frac{(x_1 - x_0 - 5)^2}{\sigma^2} + \frac{(x_2 - x_1 - 3)^2}{\sigma^2}$
- So, we want

$$\frac{1}{\sigma} x_1 - \frac{1}{\sigma} x_0 = \frac{1}{\sigma} 5 \qquad \frac{1}{\sigma} x_2 - \frac{1}{\sigma} x_1 = \frac{1}{\sigma} 3$$

- Note that the smaller σ is, the more confident you are!

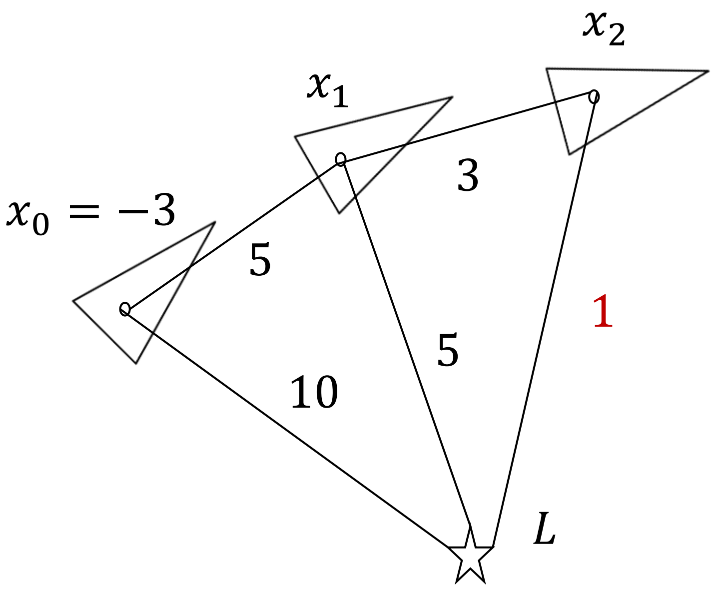
PA6C: Confident Measurements

- Modify the previous code to adjust for a highly confident last measurement.
- Do this by adding a factor of 5 ($\sigma = 0.2$) into your Omega and Xi matrices as shown below.

$$5x_2 - 5L = -5 \times 1$$

$$-5x_2 + 5L = 5 \times 1$$

$$\begin{array}{c|c|c|c|c|}
 & x_0 & x_1 & x_2 & L \\
\hline
x_0 & & & & \\
\hline
x_1 & & & & \\
\hline
x_2 & & & 5 & -5 \\
\hline
L & & & -5 & 5 \\
\hline
\end{array}
\qquad
\begin{array}{|c|}
\hline
 \\
\hline
 \\
\hline
-5 \\
\hline
5 \\
\hline
\end{array}$$

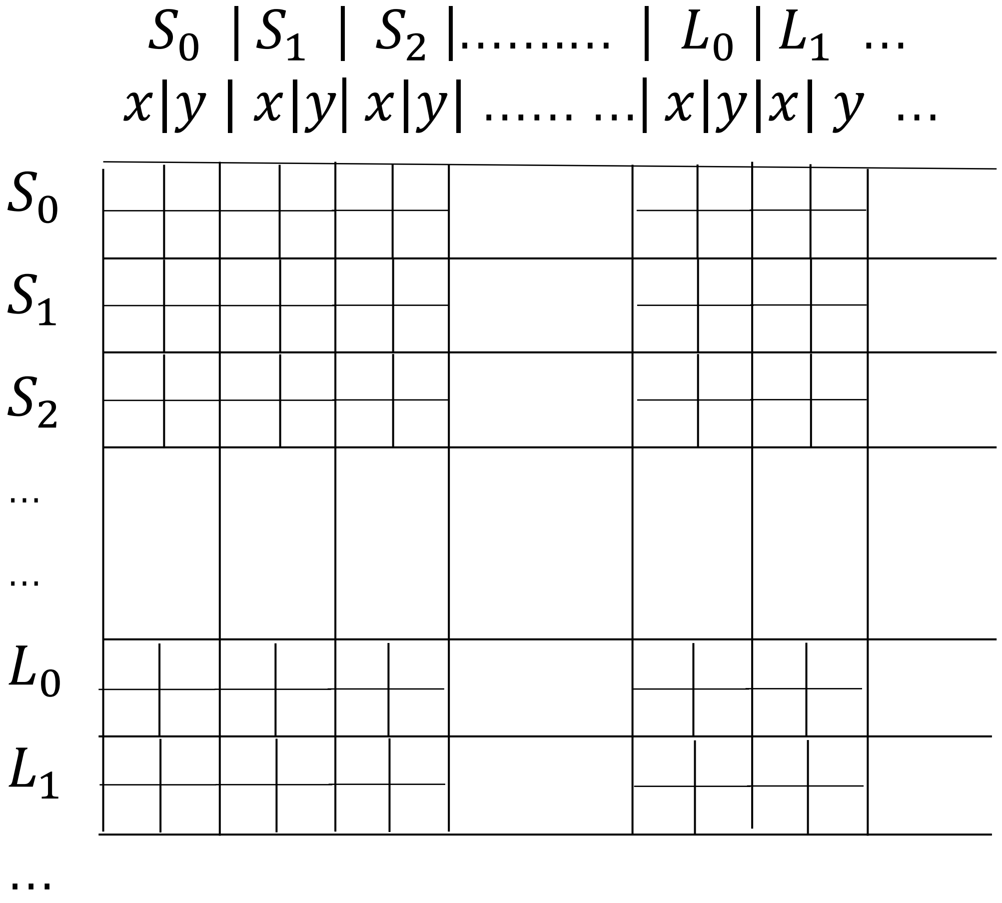


PP6A: 2D Graph SLAM (1)

- In this project you will implement SLAM in a 2-dimensional world. Please define a function, slam, which takes three parameters as input and returns the vector mu.
- Suppose there are N robot poses and M landmarks
- Omega Matrix will have dimension

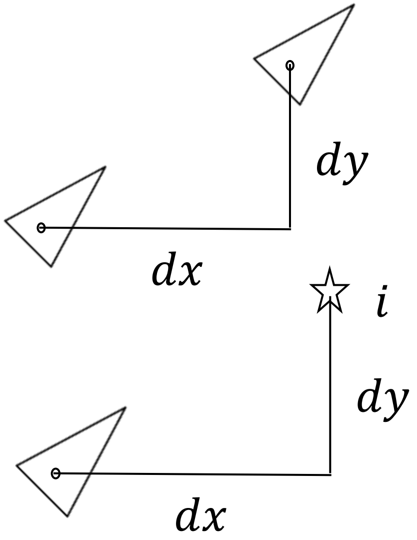
  $2(N + M) \times 2(N + M)$
- Xi Matrix will have dimension

  $2(N + M) \times 1$



PP6A: 2D Graph SLAM (2)

- Robot moving in 2D:
  - Move along x-axis and along y-axis (dx, dy).
- Measurement in 2D
  - Distance to x-axis and to y-axis (dx, dy)

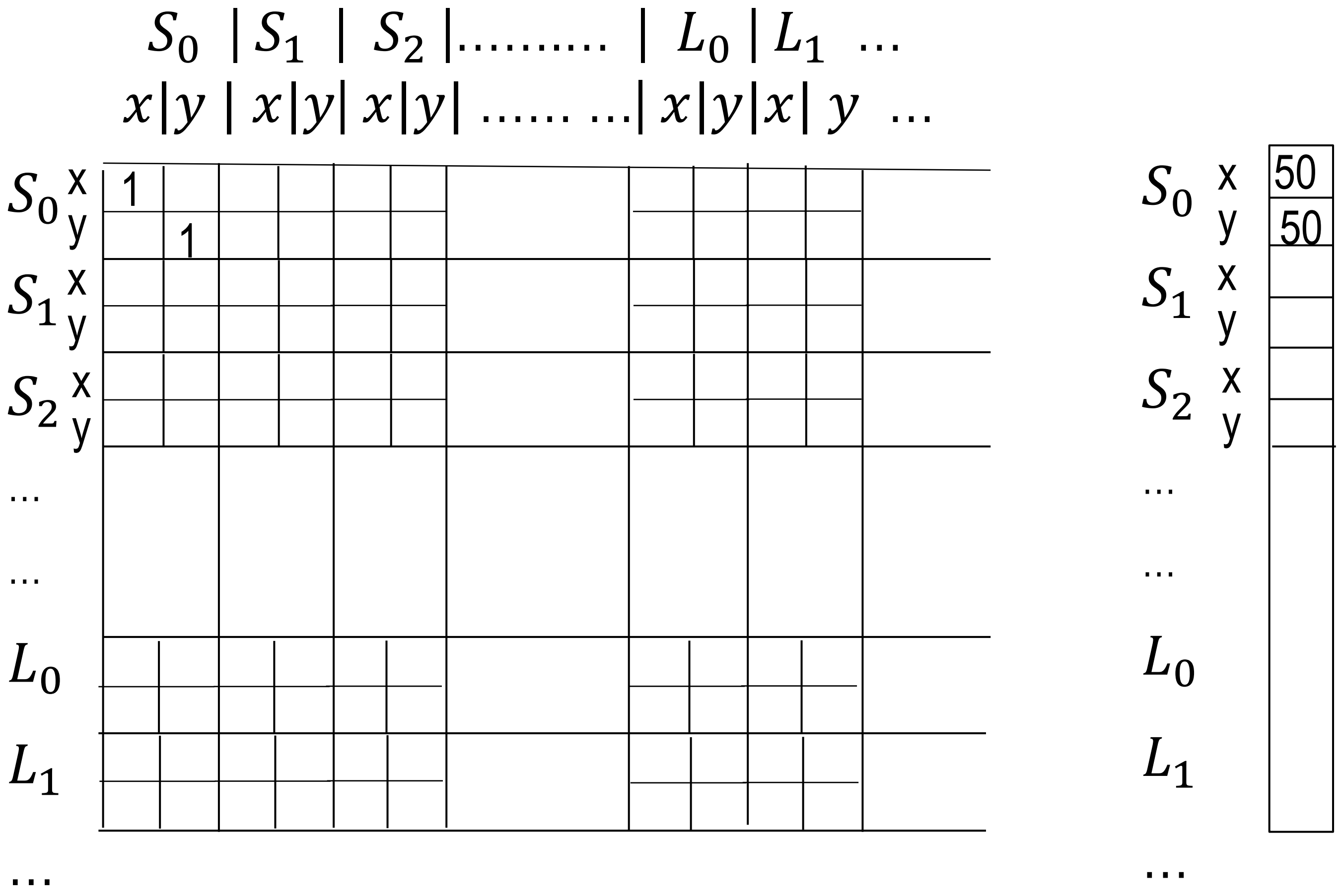


- Initial position of robot is at the center of 2D world

- The three inputs to SLAM function
  - data: a list of [Z, [dx, dy]], where Z is measurements, [dx, dy] is next move
    - Z is a list of [i, dx, dy], where dx, dy is the measure to the i's landmark
  - N: number of robot poses. Note robot moves N-1 times.
  - Num_landmarks

PP6A: 2D Graph SLAM (3)

- Update Omega and Xi.
  - Initial constraint

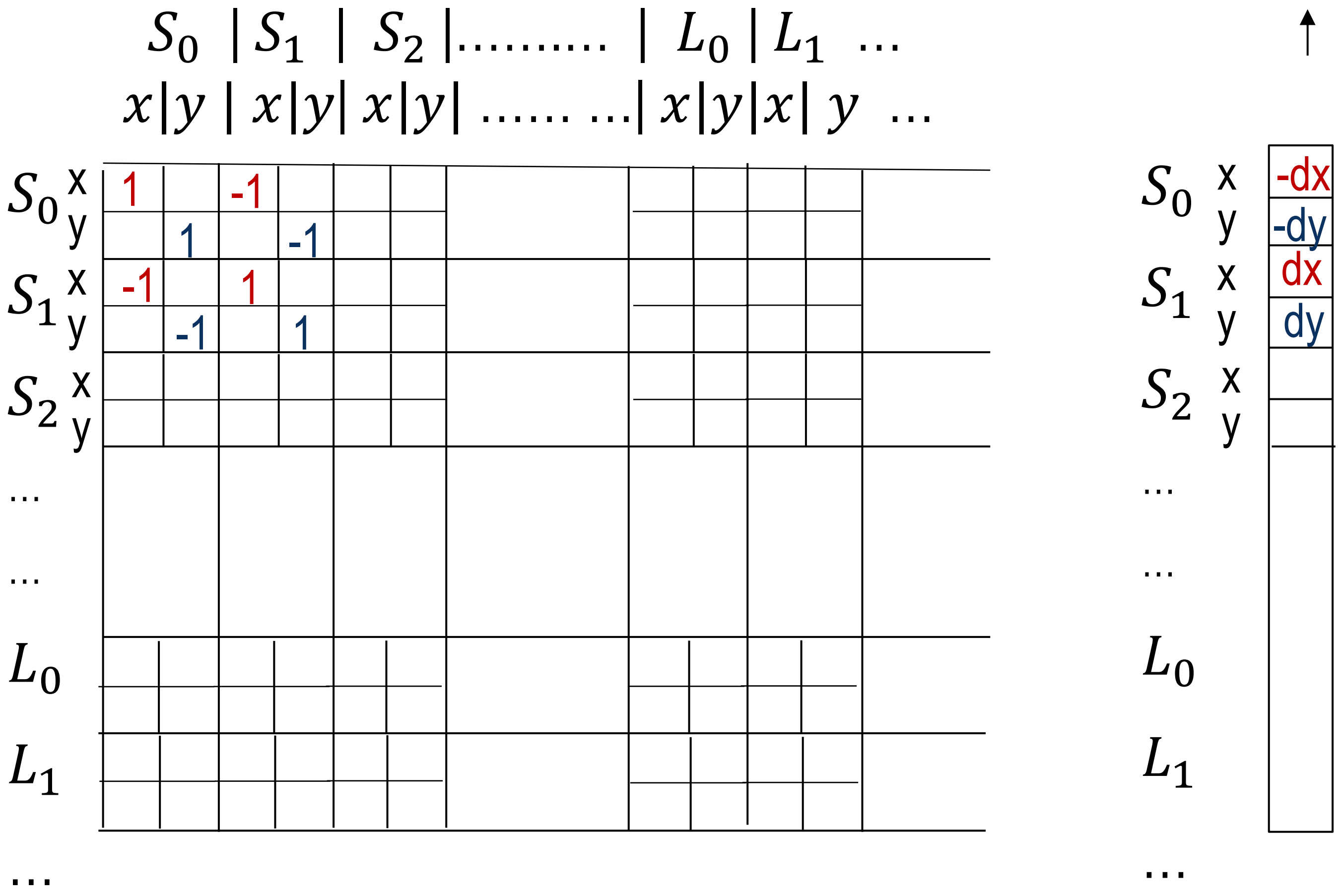


PP6A: 2D Graph SLAM (4)

- Update Omega and Xi.
  - Based on the robot motion

$S_0 \longrightarrow S_1$

dx = data[0][1][0]
dy = data[0][1][1]

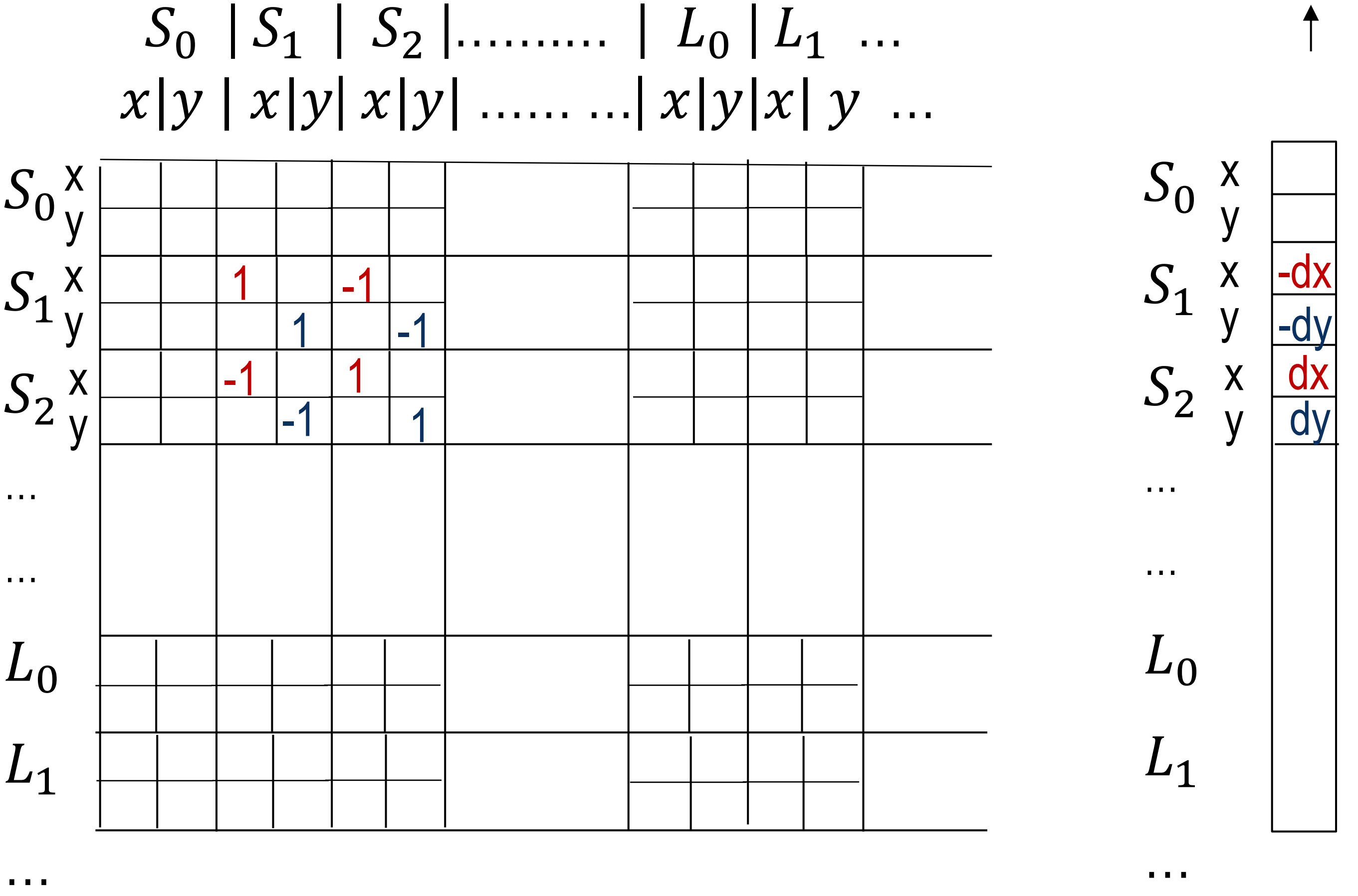


PP6A: 2D Graph SLAM (5)

- Update Omega and Xi.
  - Based on the robot motion

$S_1 \longrightarrow S_2$

dx = data[1][1][0]
dy = data[1][1][1]



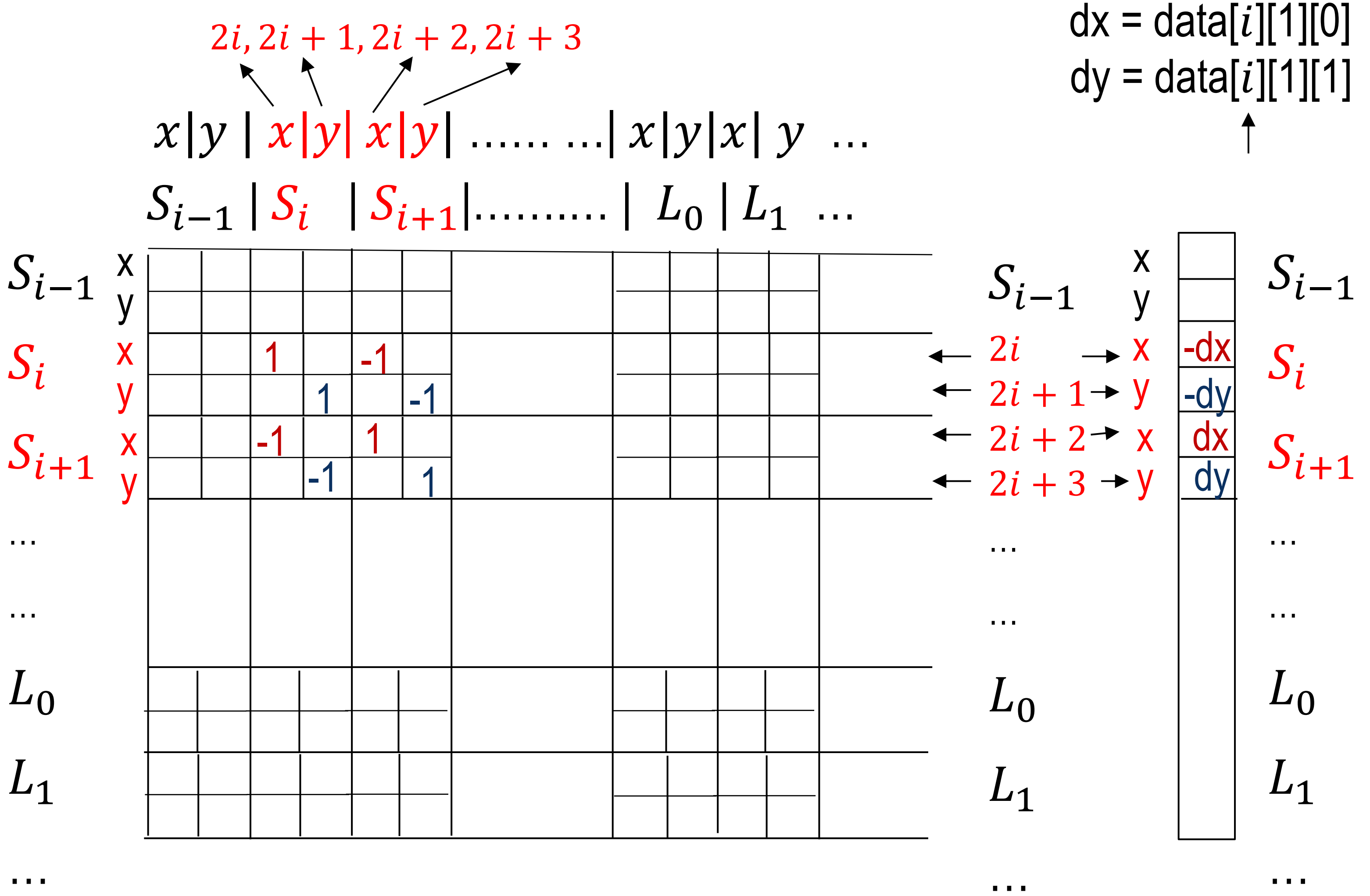
PP6A: 2D Graph SLAM (6)

- Update Omega and Xi: Based on the robot motion

$S_i \longrightarrow S_{i+1}$

dx = data[i][1][0]

dy = data[i][1][1]



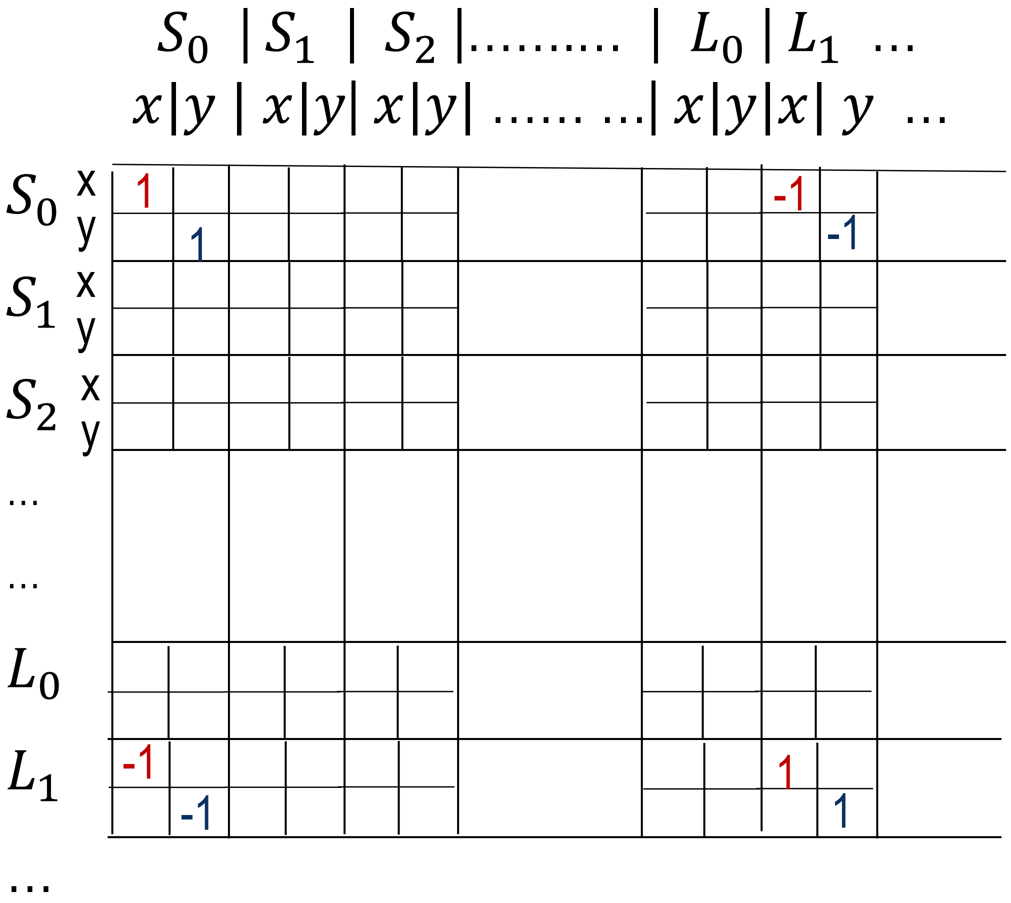
PP6A: 2D Graph SLAM (7)

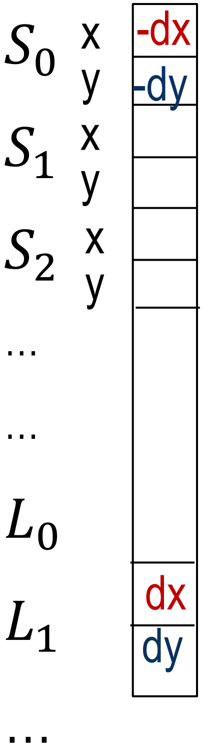
- Update Omega and Xi.
  - Based on the measurement

suppose id = 1

$S_0 \longrightarrow Z$

```
for measure in data[0][0]
    id = measure[0]
    dx = measure[1]
    dy = measure[2]
```





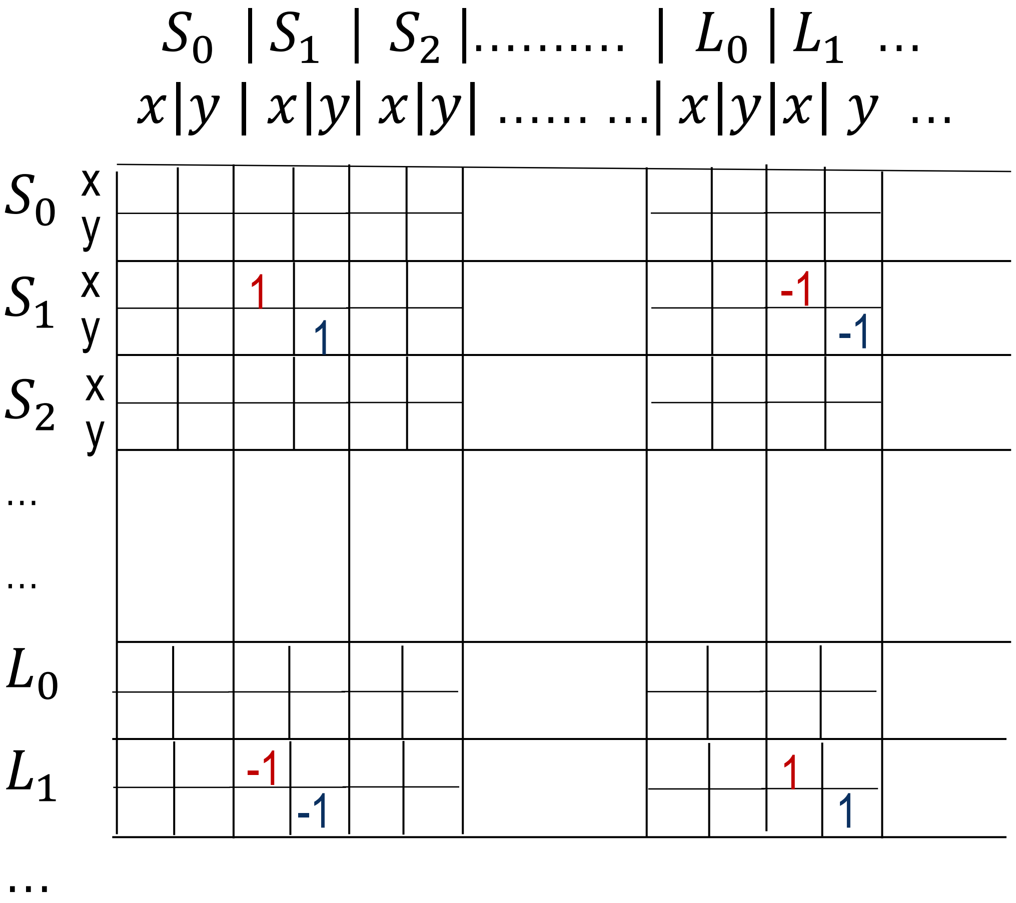
PP6A: 2D Graph SLAM (8)

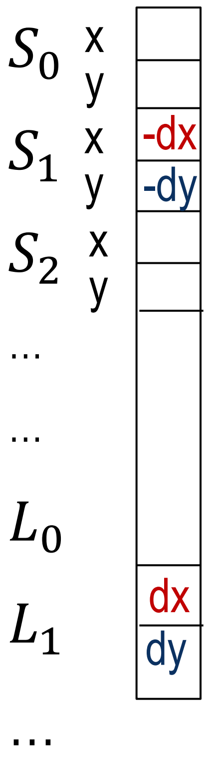
- Update Omega and Xi.
  - Based on the measurement

suppose id = 1

$S_1 \longrightarrow Z$

```
for measure in data[1][0]
    id = measure[0]
    dx = measure[1]
    dy = measure[2]
```



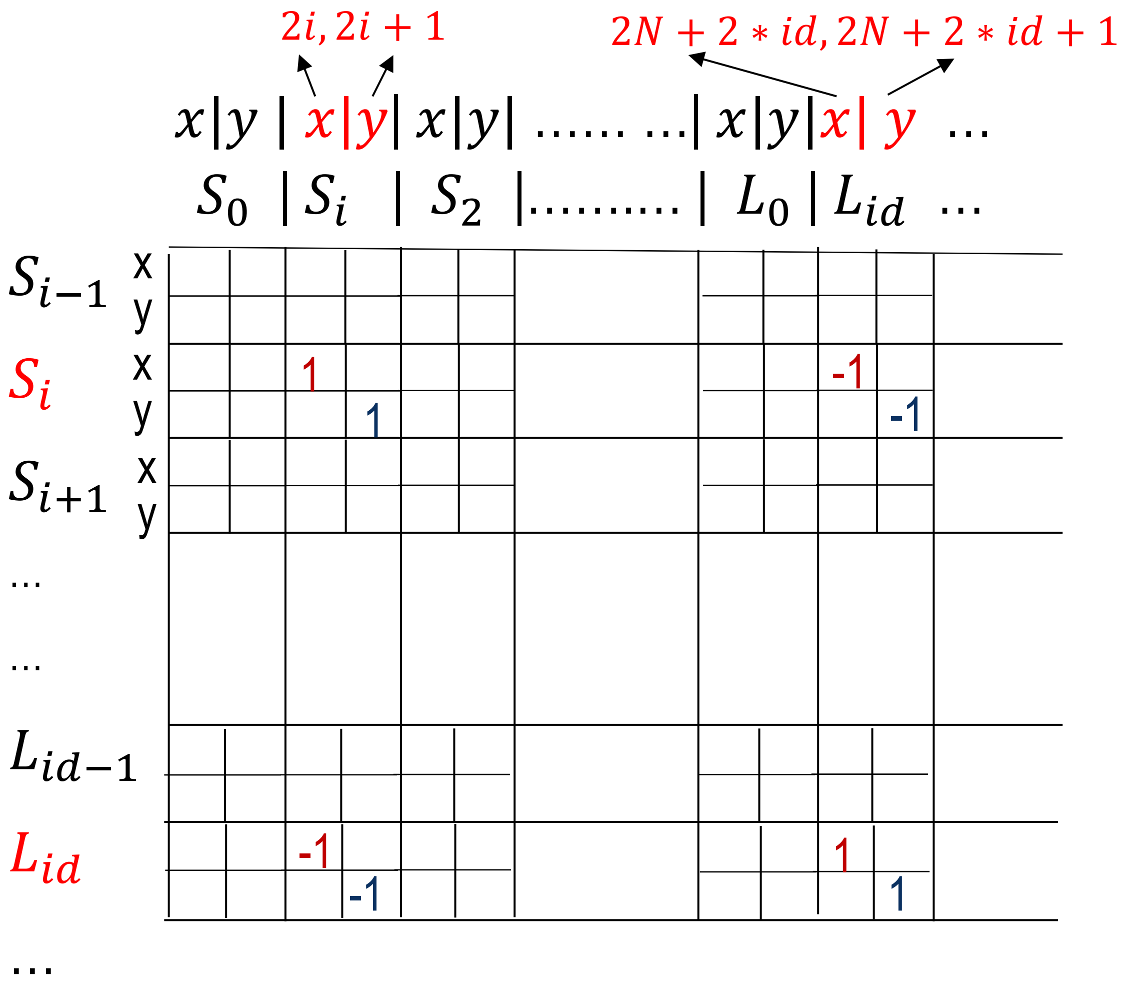


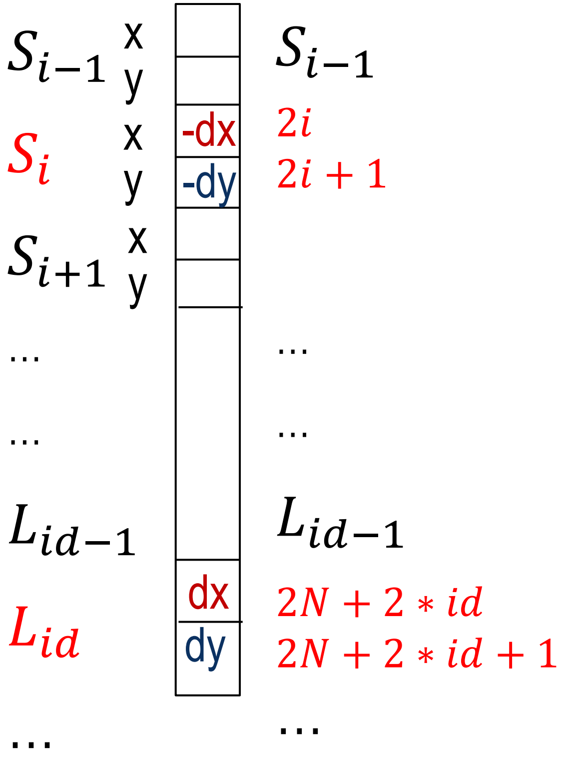
PP6A: 2D Graph SLAM (9)

- Update Omega and Xi: Based on the measurement

$S_i \longrightarrow Z$

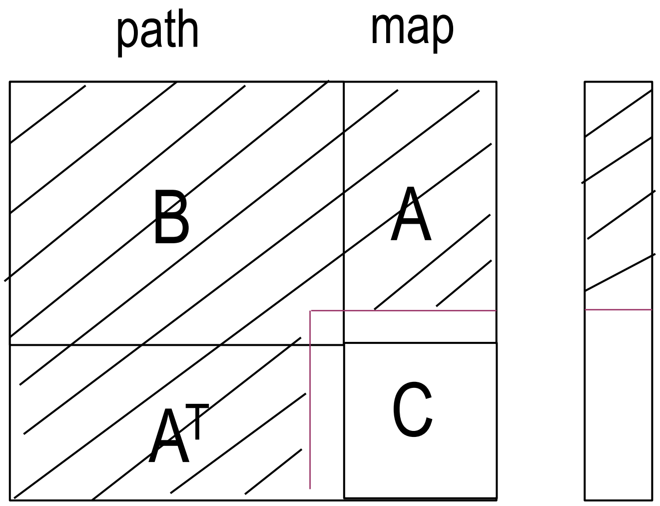
```
for measure in data[i][0]
    id = measure[0]
    dx = measure[1]
    dy = measure[2]
```

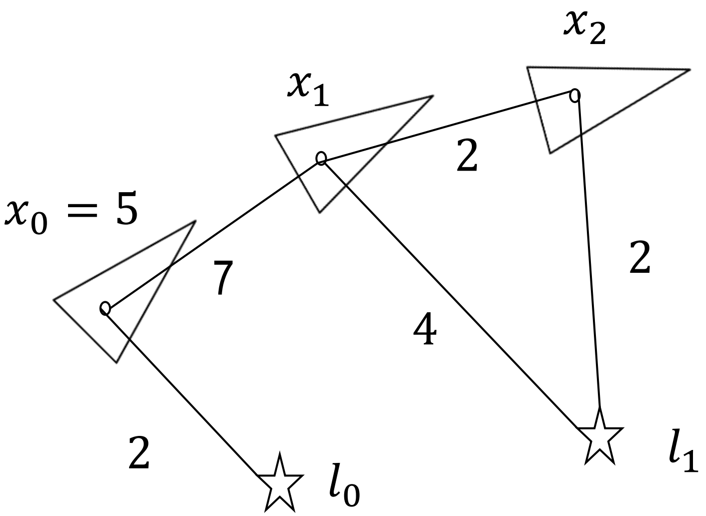




Online SLAM

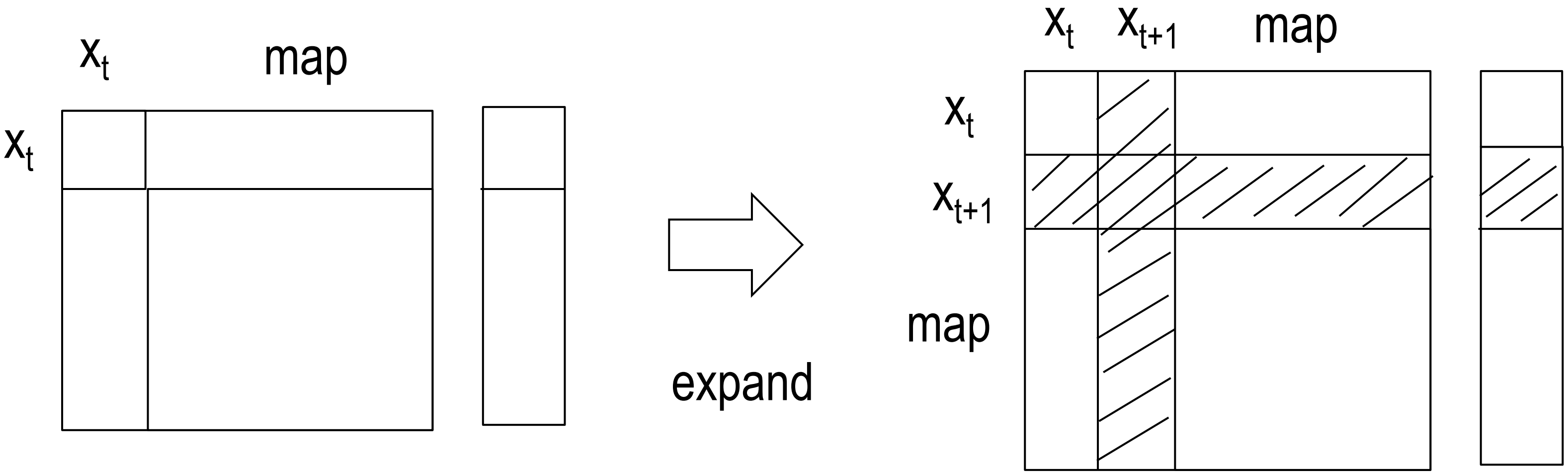
- Idea: Keep only the last robot pose in Omega matrix

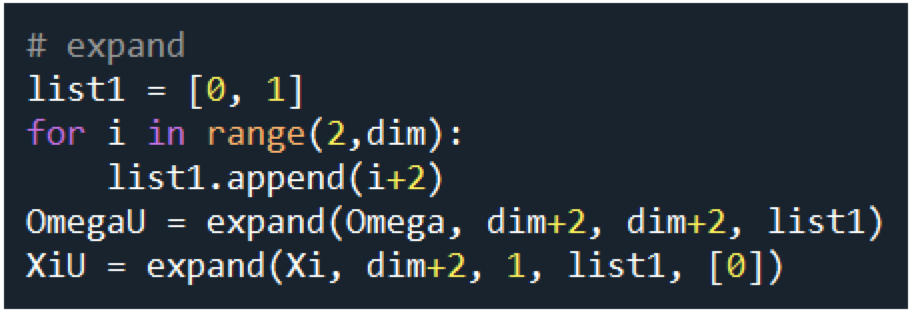




Online SLAM Algorithm (1)

- When robot moves from x_t to x_{t+1},
  1. Expand Omega and Xi and then update them.

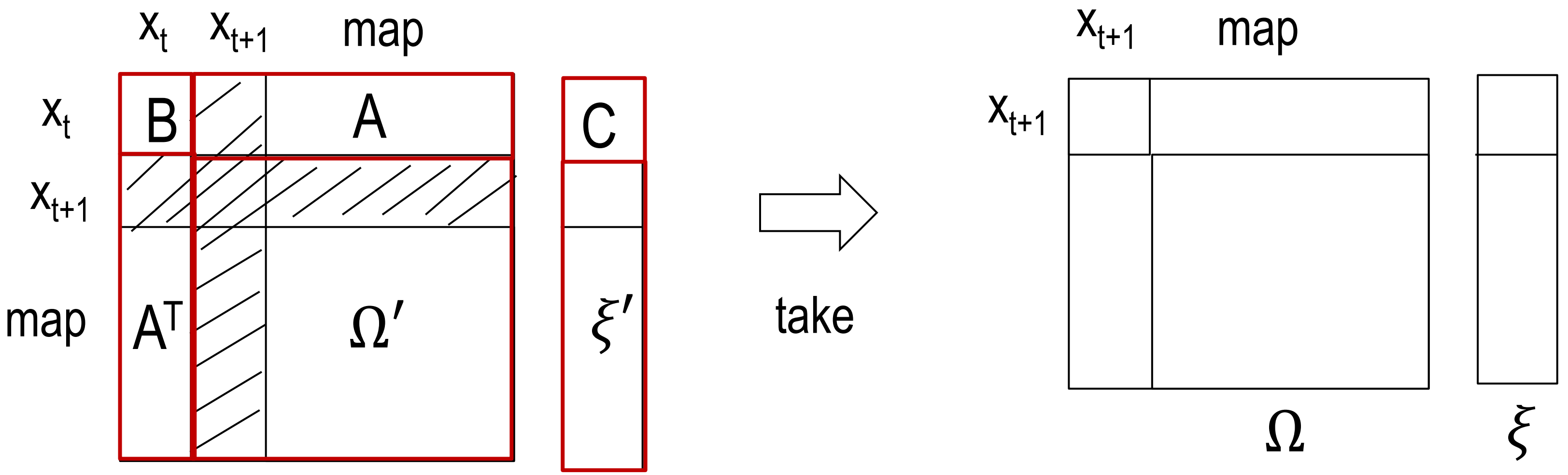


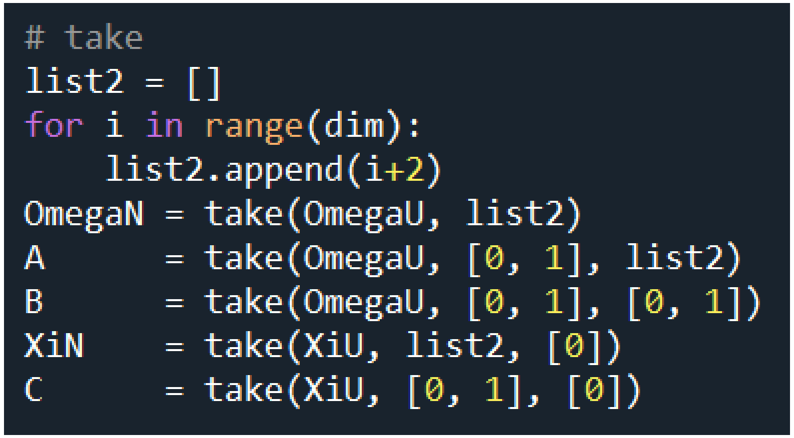


```
# expand
list1 = [0, 1]
for i in range(2,dim):
    list1.append(i+2)
OmegaU = expand(Omega, dim+2, dim+2, list1)
XiU = expand(Xi, dim+2, 1, list1, [0])
```

Online SLAM Algorithm (2)

- When robot moves from x_t to x_{t+1},
  2. Take from updated Omega and Xi and then calculate new Omega and Xi

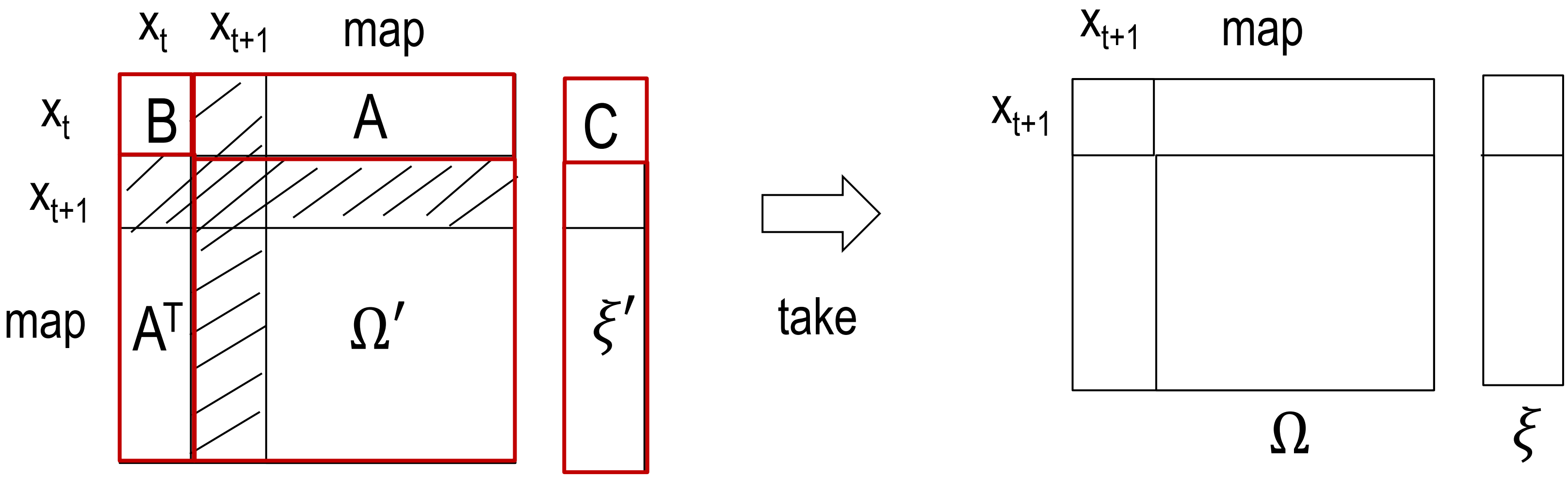




```python
# take
list2 = []
for i in range(dim):
    list2.append(i+2)
OmegaN = take(OmegaU, list2)
A      = take(OmegaU, [0, 1], list2)
B      = take(OmegaU, [0, 1], [0, 1])
XiN    = take(XiU, list2, [0])
C      = take(XiU, [0, 1], [0])
```

Online SLAM Algorithm (3)

- When robot moves from x_t to x_{t+1},
  2. Take from updated Omega and Xi and then calculate new Omega and Xi



$$\begin{pmatrix} B & A \\ A^T & \Omega' \end{pmatrix} \begin{pmatrix} x_t \\ X \end{pmatrix} = \begin{pmatrix} C \\ \xi' \end{pmatrix} \quad \Rightarrow \quad \begin{aligned} \Omega &= \Omega' - A^T B^{-1} A \\ \xi &= \xi' - A^T B^{-1} C \end{aligned}$$

$$\Omega X = \xi$$

PP6B: Online 2D Graph SLAM (1)

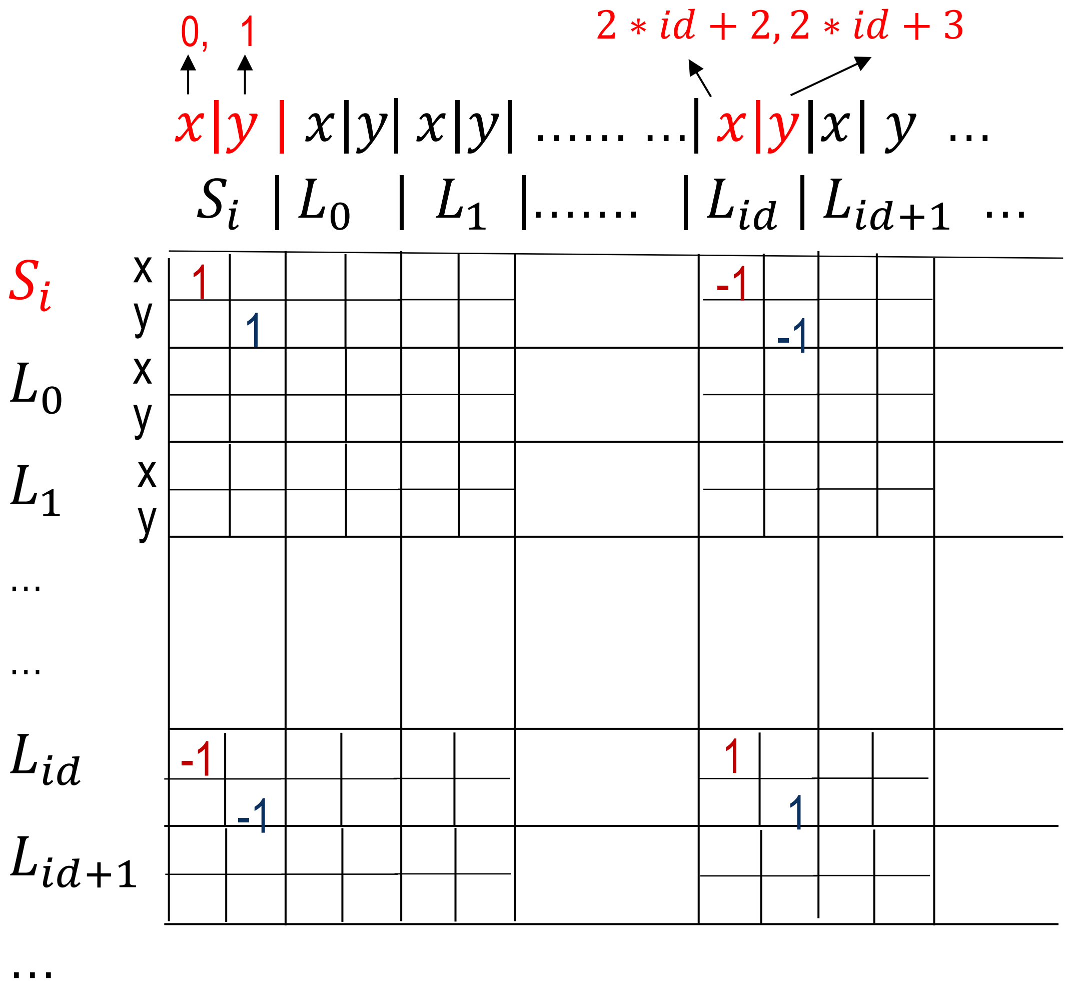
- In this project you will implement a more manageable version of graph SLAM in 2-dimensional world.
- Please define a function, online_slam, that takes five inputs and return two matrices, mu and the final Omega
- The five inputs to online-slam function
  - data: a list of [Z, [dx, dy]], where Z is measurements, [dx, dy] is next move
    - Z is a list of [i, dx, dy], where dx, dy is the measure to the i's landmark
  - N: number of robot poses. Note robot moves N-1 times.
  - Num_landmarks: number of landmarks
  - motion_noise: The noise associated with motion. The update strength for motion should be 1.0 / motion_noise
  - measurement_noise: The noise associated with measurement. The update strength for measurement should be 1.0 / measurement_noise.

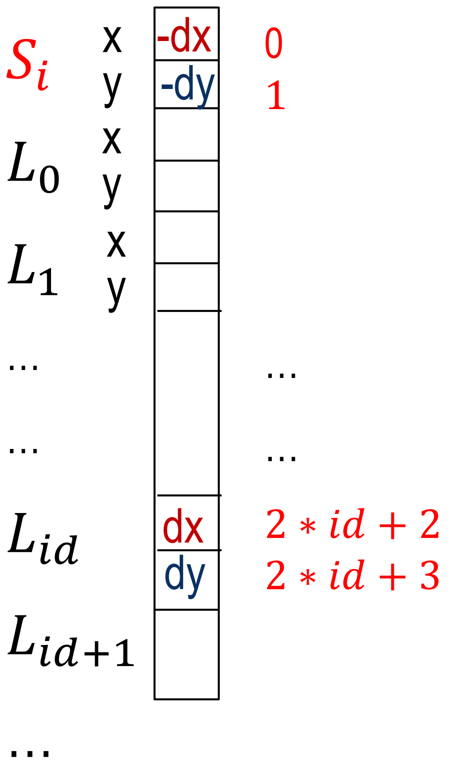
PP6B: Online 2D Graph SLAM (2)

1. Update Omega and Xi: Based on the measurement

$S_i \longrightarrow Z$

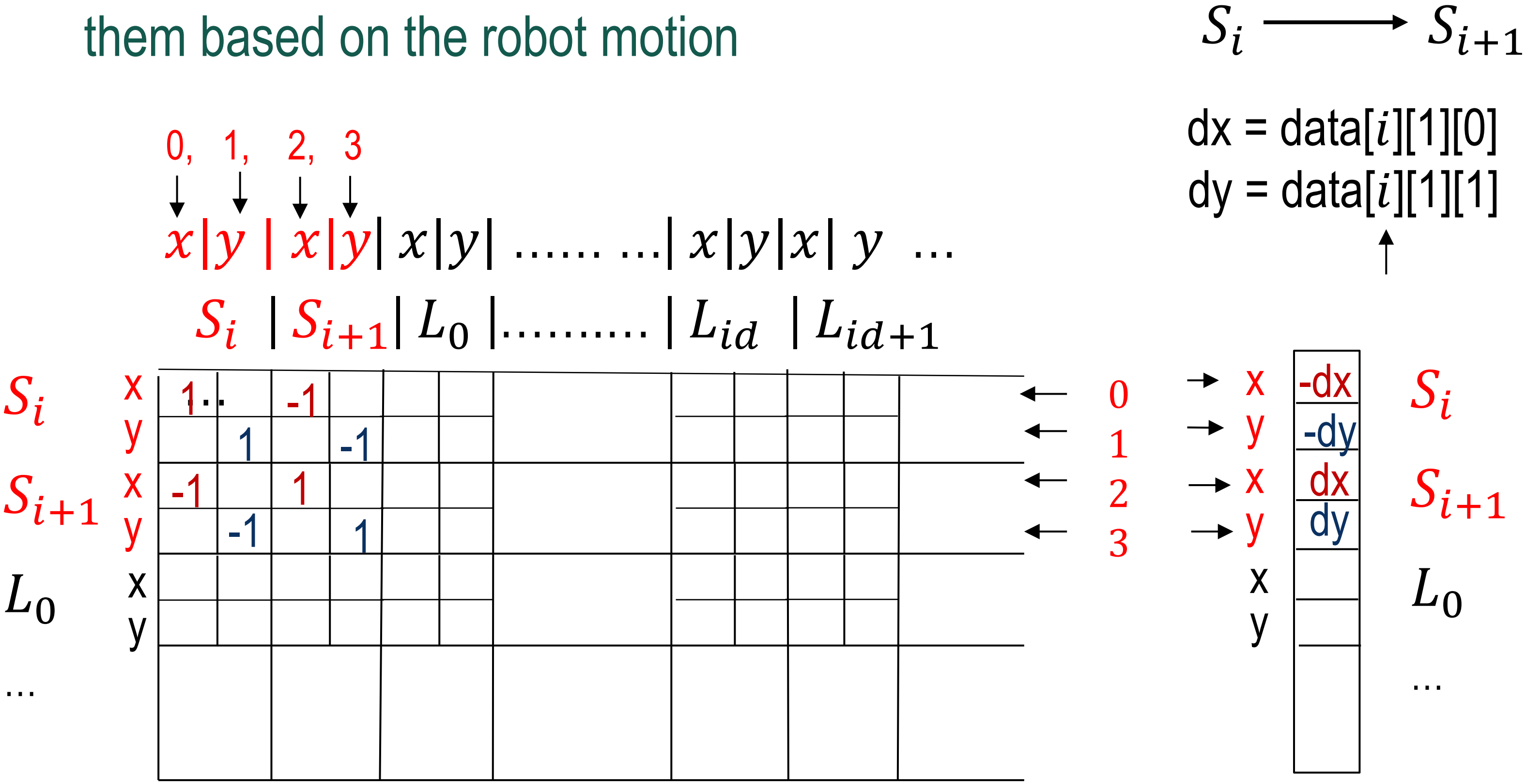
```
for measure in data[$i$][0]
    id = measure[0]
    dx = measure[1]
    dy = measure[2]
```





PP6B: Online 2D Graph SLAM (3)

2. Expand Omega and Xi and then update them based on the robot motion



$S_i \longrightarrow S_{i+1}$

dx = data[i][1][0]
dy = data[i][1][1]

0, 1, 2, 3

$x|y\ |\ x|y|\ x|y|\ \ldots\ldots\ \ldots|\ x|y|x|\ y\ \ \ldots$

$S_i\ \ |S_{i+1}|\ L_0\ |\ldots\ldots\ldots|\ L_{id}\ \ |L_{id+1}$

$$
\begin{array}{cc|cccc|cc|c|c}
 & & x & y & x & y & x & y & \ldots & \\
S_i & x & 1.. & & -1 & & & & & 0 \\
 & y & & 1 & & -1 & & & & 1 \\
S_{i+1} & x & -1 & & 1 & & & & & 2 \\
 & y & & -1 & & 1 & & & & 3 \\
L_0 & x & & & & & & & & \\
 & y & & & & & & & & \\
\ldots & & & & & & & & &
\end{array}
\qquad
\begin{array}{c|c|c}
x & -dx & S_i \\
y & -dy & \\
x & dx & S_{i+1} \\
y & dy & \\
x & & L_0 \\
y & & \\
 & & \ldots
\end{array}
$$

3. Take from updated Omega and Xi and then calculate new Omega and Xi