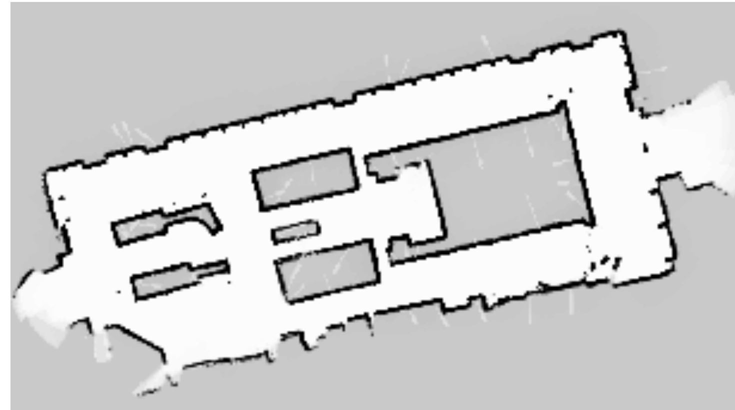
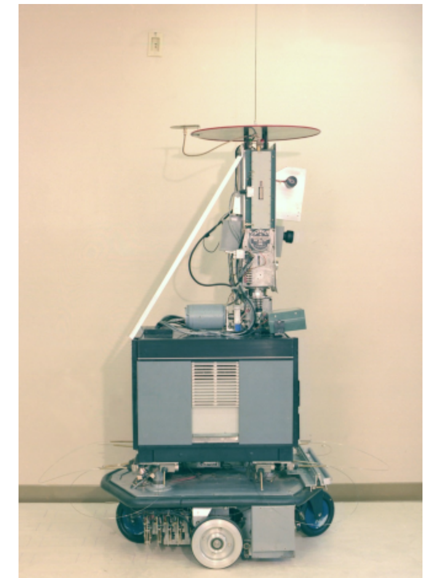
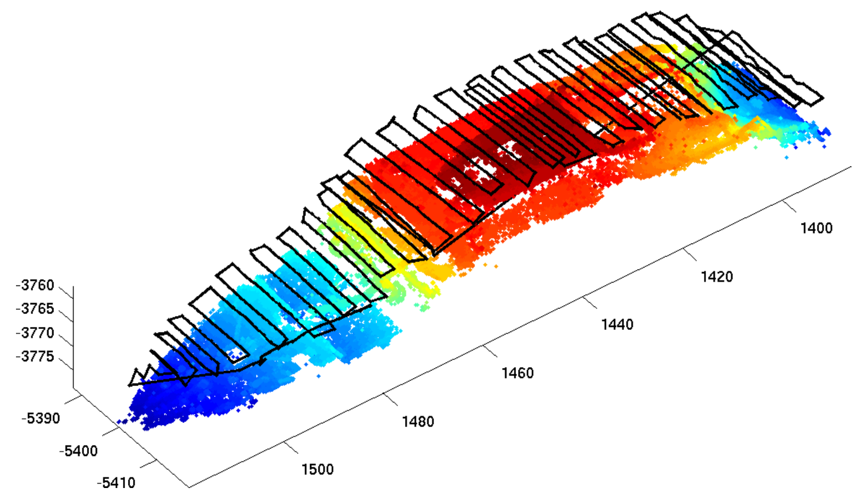
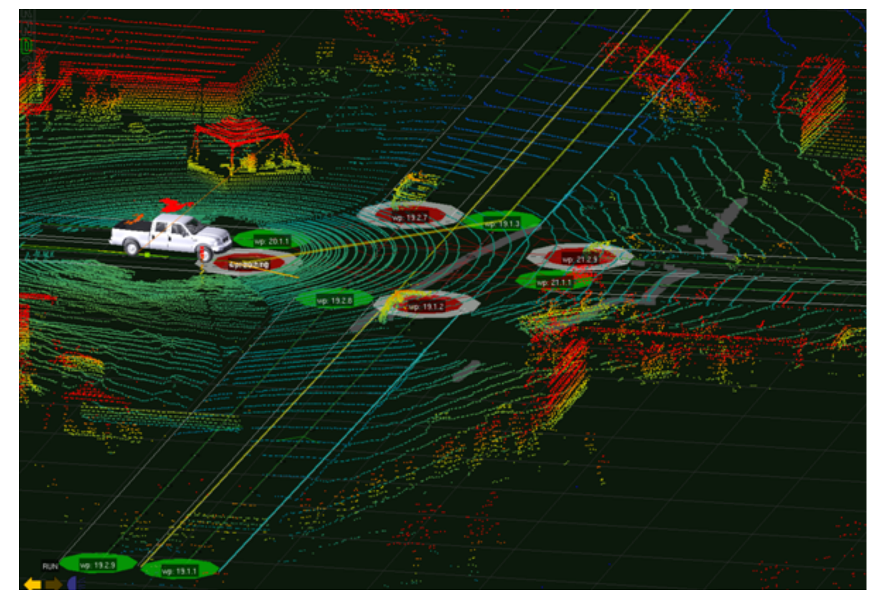








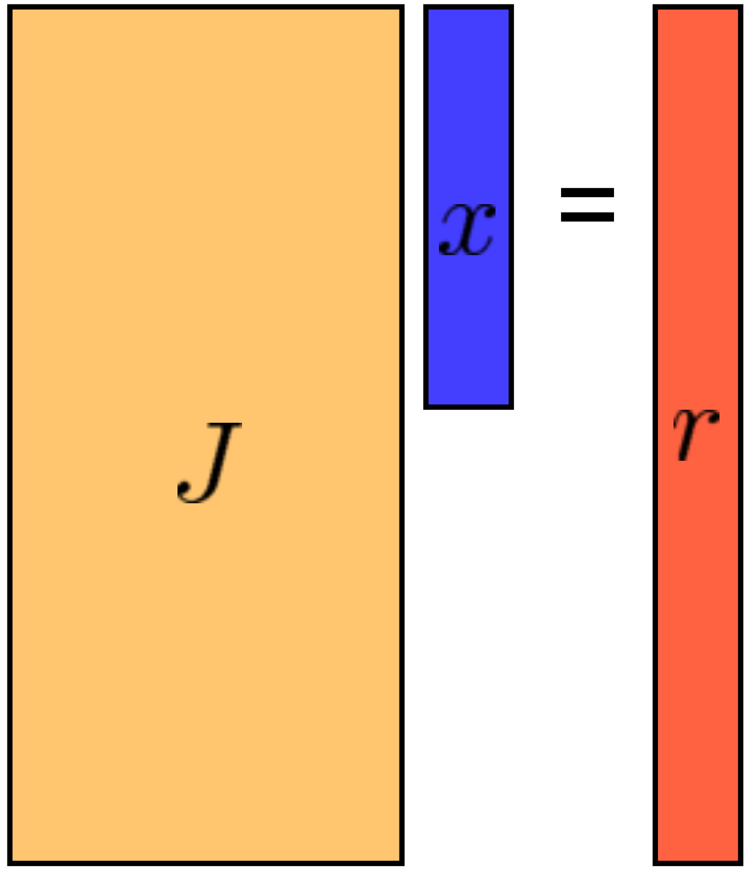
EECS568 Mobile Robotics: Methods and Principles

Prof. Edwin Olson

L04. Least Squares SLAM

Least-squares regression

- We collect simultaneous equations, then solve for x.
  - Or change in x for non-linear regression



$$Ax = b$$

$$A\Delta x = b$$

Non-linear least squares

$$x^2 + 3x + y^3 + y = 4$$
$$\sin(x) + y^2 = 1$$
$$x + 2/y = 4$$

$$f(x) = b$$

$$J|_{x_0}(x - x_0) + f(x_0) = b$$

$$x_0 = \left[\begin{array}{c} 0 \\ 1 \end{array} \right]$$

Solution

$$\begin{bmatrix} 3 & 4 \\ 1 & 2 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 4 \\ 1 & 2 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 1.2 \\ -0.4333 \end{bmatrix}$$

Weighted least-squares (non-probabilistic story)

- What if we want some equations to count more than others?
  - What happens if we repeat an equation?
    - It counts twice
  - Least squares minimizes the squares of the residuals
    - What if we multiply an equation by w?
    - Its residual gets scaled by w!

because error is a vector... and [1.414 1.414] has the same length as 2

  - Why does multiplying one eqn by 1.414 equivalent to repeating it twice?

Weighted least-squares (probabilistic story)

- Where do we get the weights?
  - Consider the *uncertainty* of each equation

$$
\begin{aligned}
x^2 + 3x + y^3 + y + w_1 &= 4 \\
\sin(x) + y^2 + 4w_2 &= 1 \\
x + 2/y + x(w_1 + w_2) &= 1
\end{aligned}
\qquad
\Sigma_w = \left[\begin{array}{cc} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{array} \right]
$$

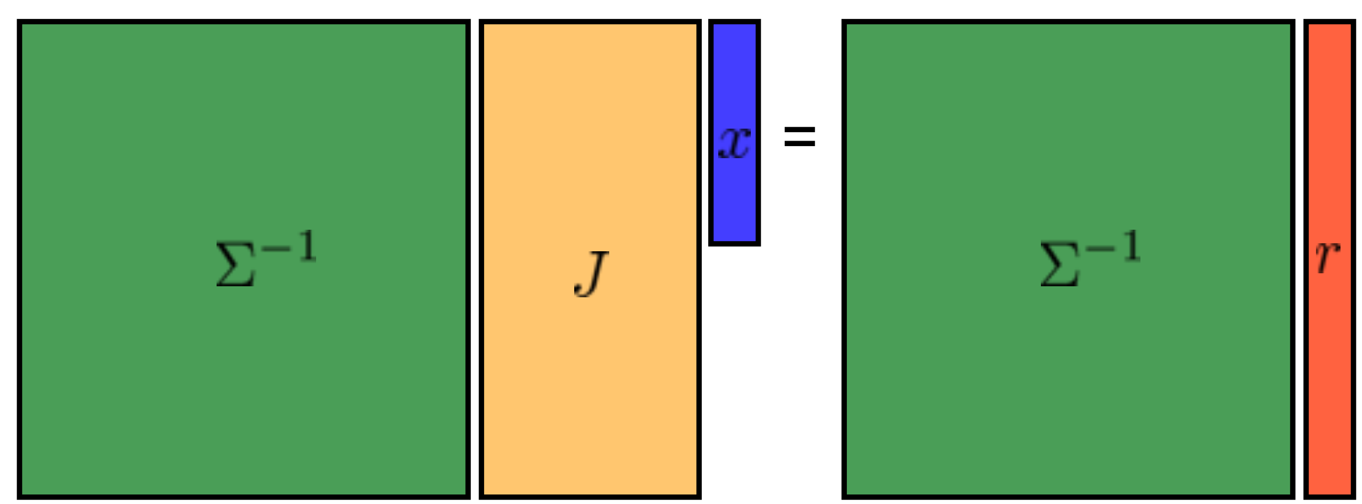
- How should we "scale" each equation?

Weighted least squares

- Consider the uncertainty of each equation
  - What's the covariance of the *set of hyperplanes?*

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 4 \\ x_0 & x_0 \end{bmatrix} \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \\ x_0 & x_0 \end{bmatrix}^T$$

$$\Sigma^{-1} J \Delta x = \Sigma^{-1} b$$



SLAM

- Simultaneous Localization and Mapping
  - Why do we call it this?
- Front ends
  - Sensor processing (not today)
  - Estimation
- Our approach today:
  - Weighted least squares for estimation

State representation

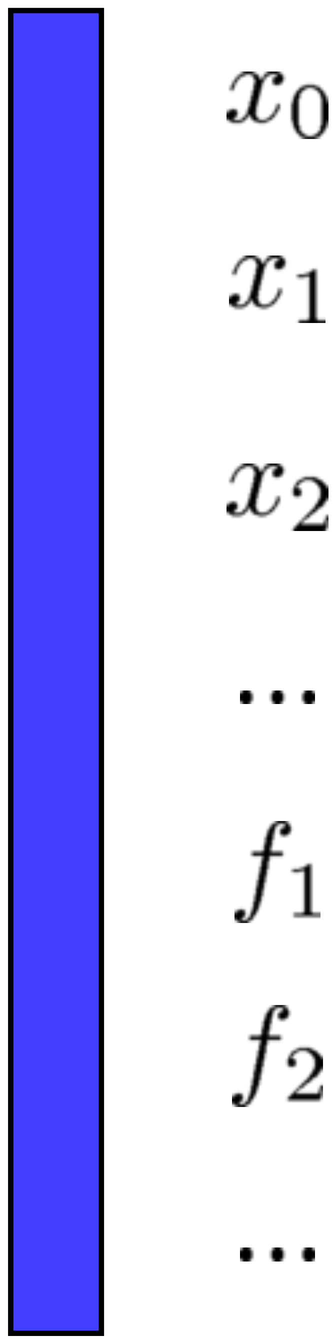
- Robot is operating in the 2D plane; each robot pose represented by:

$$x_i = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- Landmarks (e.g. trees)

$$f_i = \begin{bmatrix} x \\ y \end{bmatrix}$$

State vector



- Of course, actual order of variables in state vector doesn't matter
  - (A, x, and b need to agree on order, of course)

Propagation steps

- Robot is driving down a hallway
  - State added at every time step
  - Equations relate state at time t+1 to time t

$$
\begin{aligned}
x_2 &= d_1 \cos(\theta_1) - d_1 \sin(\theta_1) + x_1 + w_1 \\
y_2 &= d_1 \sin(\theta_1) + d_1 \cos(\theta_1) + y_1 + w_2 \\
\theta_2 &= \theta_1 + \Delta\theta_1 + w_3
\end{aligned}
$$

$$
\begin{aligned}
w_1 &\sim N(0, \sigma_1^2) \\
w_2 &\sim N(0, \sigma_2^2) \\
w_3 &\sim N(0, \sigma_3^2)
\end{aligned}
$$

Collect equations...

$$
\begin{aligned}
x_2 &= d_1 \cos(\theta_1) - d_1 \sin(\theta_1) + x_1 + w_1 \\
y_2 &= d_1 \sin(\theta_1) + d_1 \cos(\theta_1) + y_1 + w_2 \\
\theta_2 &= \theta_1 + \Delta\theta_1 + w_3 \\
x_3 &= d_2 \cos(\theta_2) - d_2 \sin(\theta_2) + x_2 + w_4 \\
y_3 &= d_2 \sin(\theta_2) + d_2 \cos(\theta_2) + y_2 + w_5 \\
\theta_3 &= \theta_2 + \Delta\theta_2 + w_6
\end{aligned}
$$

- And collect terms to write in matrix form.

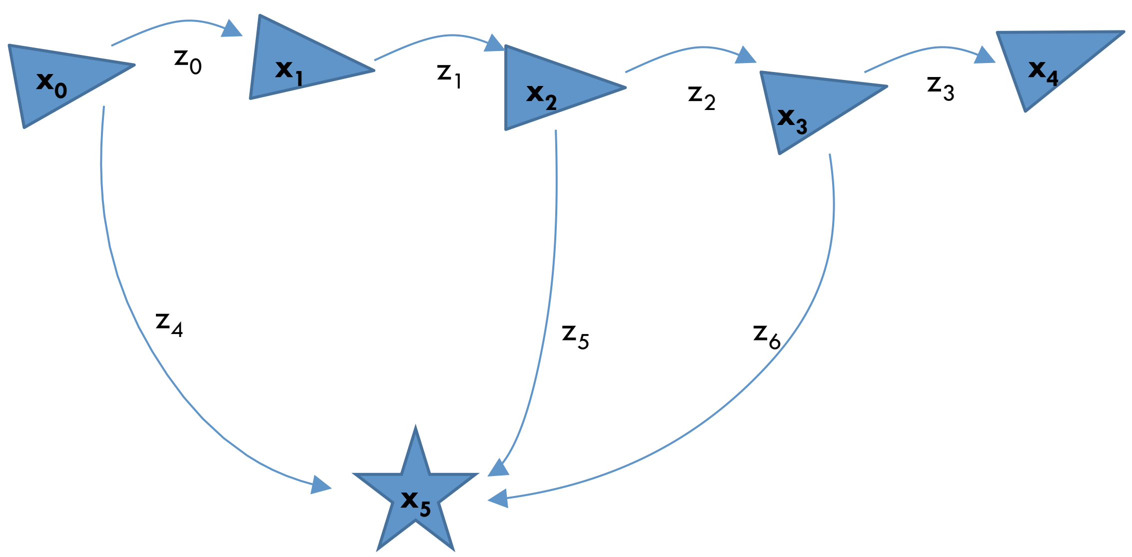
Simple example

- 1D robot

$$x_2 = x_1 + d_1 + w_1$$
$$x_3 = x_2 + d_2 + w_2$$
$$x_4 = x_3 + d_3 + w_3$$
$$x_5 = x_4 + d_4 + w_4$$

$$x_2 - x_1 = d_1 + w_1$$
$$x_3 - x_2 = d_2 + w_2$$
$$x_4 - x_3 = d_3 + w_3$$
$$x_5 - x_4 = d_4 + w_4$$

Loop closures

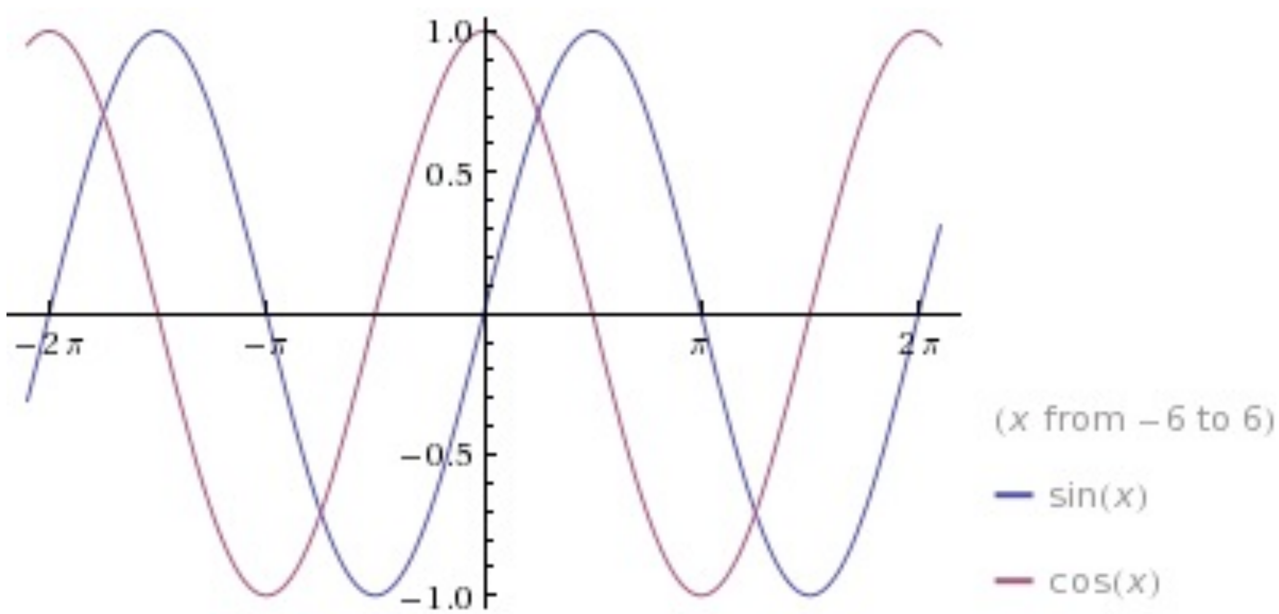


$$z_0 = f_0(x_0, x_1)$$
$$z_1 = f_1(x_1, x_2)$$
$$z_2 = f_2(x_2, x_3)$$
$$z_3 = f_3(x_3, x_4)$$

$$z_4 = f_4(x_0, x_5)$$
$$z_5 = f_5(x_2, x_5)$$
$$z_6 = f_6(x_3, x_5)$$

Effects of linearization

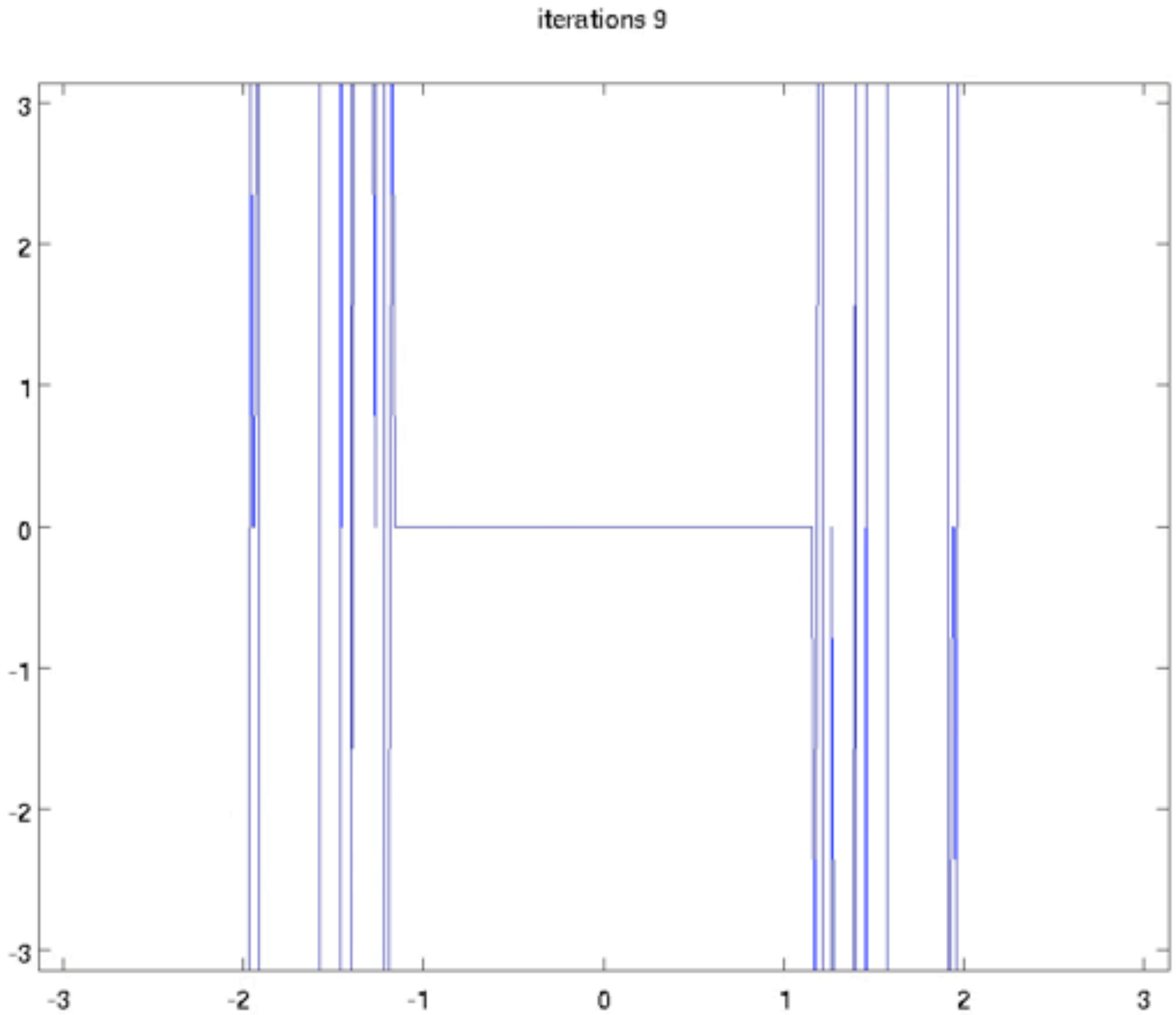
- When does linearization become a problem?
  - Rotations are the common killer [consider sin(x)]:
    - Even at favorable points (where derivative is at an extremum), an error of pi/4 can yield the wrong sign!
    - Rules of thumb
      - a potential problem at 30 degrees
      - potentially catastrophic at 67 degrees



67 degrees

- Consider Newton's method for finding roots of sin(x)
- do {
    x = x - sin(x) / cos(x);
  } until converged
- Region of convergence?
  - +/- 67 degrees

plots: what value do we compute after N iterations of Gauss-Newton where the initial value is x?



Rank Deficiency

- Our problem will always be rank deficient
  - Even if we have many loop closures. Why?

- How do we solve this problem?
  - "Pin" the first pose. (How?)

The whole graph can be rotated around arbitrarily in space.

Pinning can slow convergence in some cases.