COMP0130: Robotic Vision and Navigation

Lecture 6A: Going Nonlinear and Big with Kalman Filters

UCL ENGINEERING
Change the world

Structure

- Motivation
- Definitions of success for a Kalman filter
- Nonlinear example
- EKF-SLAM
- Scalability
- Performance

Going Nonlinear …

- BeadSLAM is a simple linear test case, but it’s not reflective of any real SLAM system
- Real SLAM systems have *nonlinear models* and are often *very high dimensional*
- We’ll now look at how to use a Kalman Filter in these cases

The SLAM Big Equation (SBE)

- The expression for the probabilistic formulation is

$$p\left(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0\right)$$

Current pose of the platform

Map

Set of all observations

$\mathbf{Z}_{0:k} = \{\mathbf{z}_0, \dots, \mathbf{z}_k\}$

Set of all control inputs

$\mathbf{U}_{0:k} = \{\mathbf{u}_0, \dots, \mathbf{u}_k\}$

Initial conditions

Reducing the Notation

- This is too much to write so, we'll use simplified notation

$$\mathbf{s}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{m}^1 \\ \vdots \\ \mathbf{m}^n \end{bmatrix}$$

$$p\left(\mathbf{s}_k \middle| \mathbb{I}_k\right)$$

$$\mathbb{I}_k = \{\mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0\}$$

The Ideal Kalman Filter

- In the ideal world, the Kalman filter computes the *mean* and *covariance* of the probability distribution over the system state

$$\hat{\mathbf{s}}_{i|j} = \mathrm{E}\left[\mathbf{s}_i | \mathbb{I}_j\right]$$

$$\mathbf{P}_{i|j} = \mathrm{E}\left[\left(\mathbf{s}_i - \hat{\mathbf{s}}_{i|j}\right)\left(\mathbf{s}_i - \hat{\mathbf{s}}_{i|j}\right)^\top | \mathbb{I}_j\right]$$

$$\hat{S}_{i|i} = \hat{S}_{i|i-1} + W \nu_{i|i-1}$$

Kalman Filters in the Real World

- In reality the filter rarely – if ever – actually computes the mean and covariance:
  - The models aren't really known
  - The noise models aren't really known
  - Nonlinearities mean that we can't work out the maths precisely

- Therefore, we seek estimates which are "about" right

Kalman Filters in the Real World

- The mean in the filter should be ”close” to the actual mean of the system

$$\hat{\mathbf{s}}_{i|j} \approx \mathrm{E}\left[\mathbf{s}_i | \mathbb{I}_j\right]$$

- We also seek a *covariance consistent estimate*

$$\mathbf{P}_{i|j} \geq \mathrm{E}\left[\left(\mathbf{s}_i - \hat{\mathbf{s}}_{i|j}\right)\left(\mathbf{s}_i - \hat{\mathbf{s}}_{i|j}\right)^\top | \mathbb{I}_j\right]$$

Covariance Consistency in 1D

$$\mathbf{P}_{i|j} \geq \mathrm{E}\left[\left(\mathbf{s}_i - \hat{\mathbf{s}}_{i|j}\right)\left(\mathbf{s}_i - \hat{\mathbf{s}}_{i|j}\right)^\top | \mathbb{I}_j\right]$$

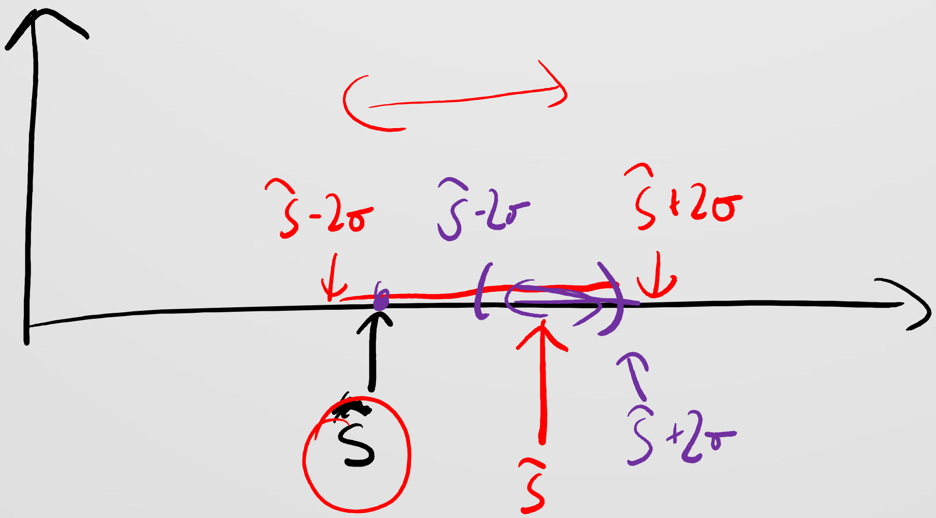
$\leq$

$\sigma_{i|j} = \sqrt{P_{i|j}}$

$\hat{s} \pm c\sigma$

$c=1 \sim 66\%$

$c=2 \sim 93\%$



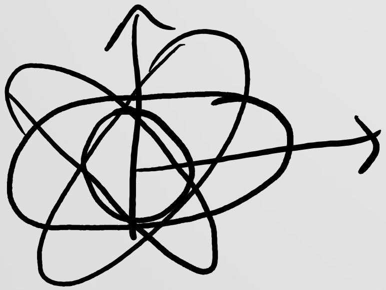
Covariance Consistency in 2D

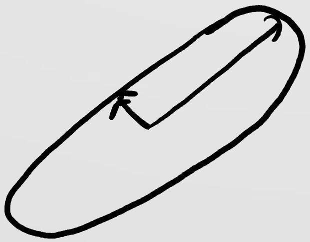
$$\underbrace{\mathbf{P}_{i|j} \geq \mathrm{E}\left[\left(\mathbf{s}_i - \hat{\mathbf{s}}_{i|j}\right)\left(\mathbf{s}_i - \hat{\mathbf{s}}_{i|j}\right)^\top | \mathbb{I}_j\right]}$$

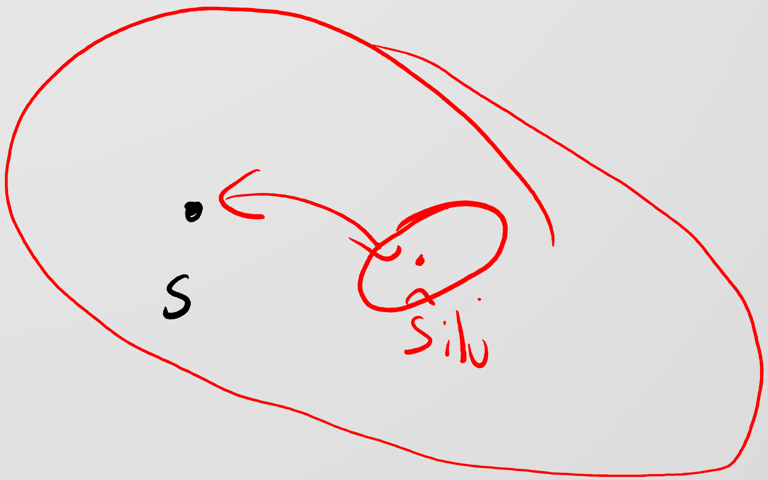
$\hat{x} \pm c\,\sigma$

$\sigma \to 2D?$

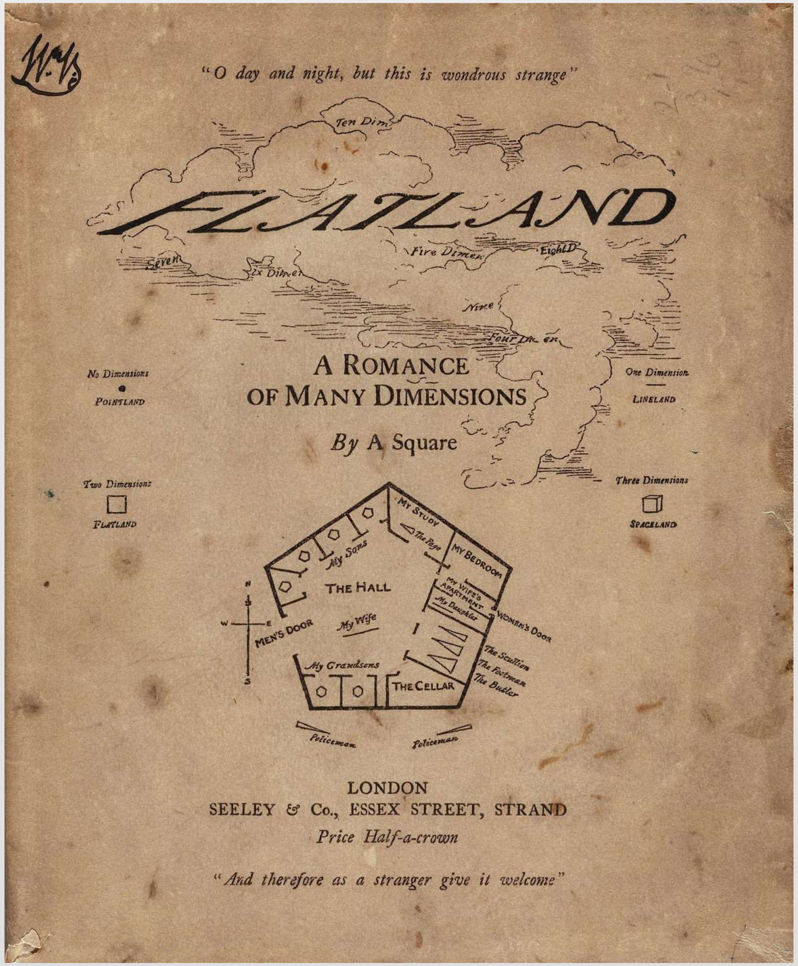
$\underline{s}$: $c = (s - \hat{s}_{i|j})^T P_{i|j}^{-1} (s - \hat{s}_{i|j})$ $\to$ scalar



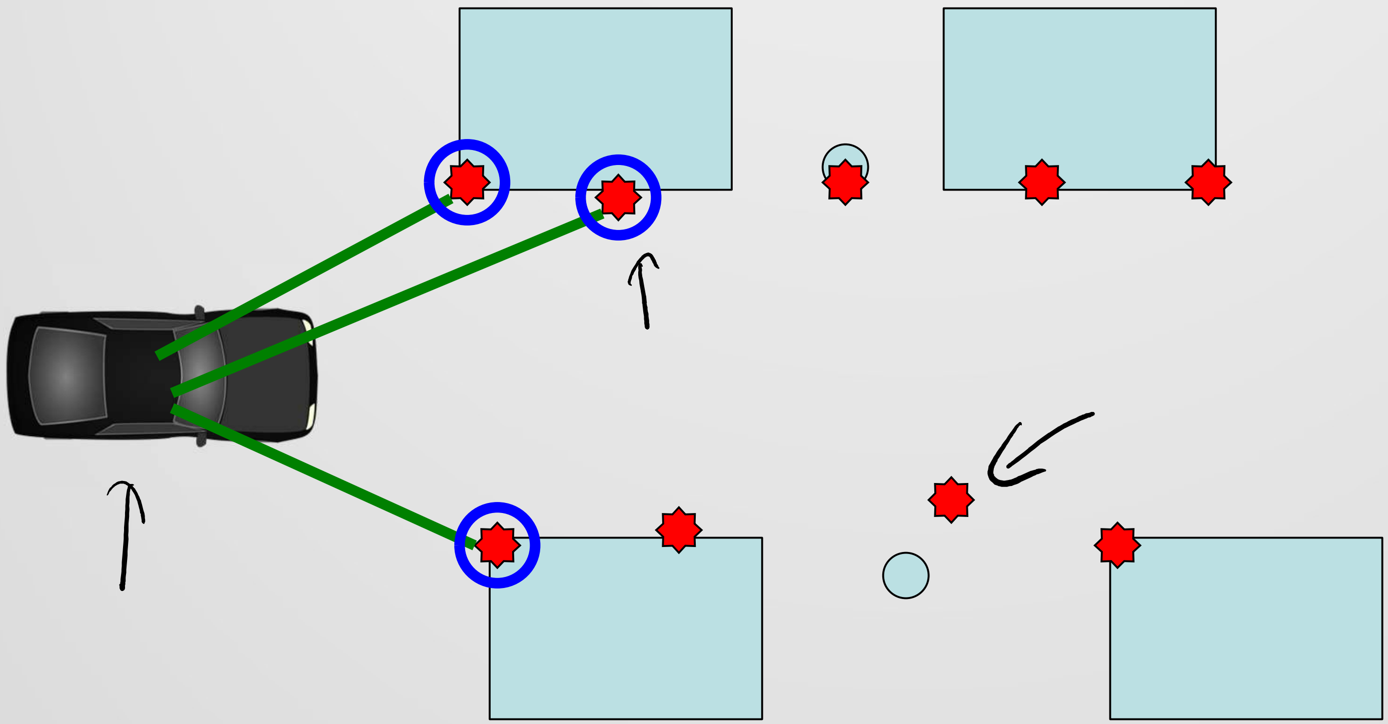




Use Case: Flatland SLAM



Flatland SLAM



Flatland SLAM (FLSLAM)

- The platform and targets are all in 2D
- The features are stationary point-like landmarks
- The features are uniquely identifiable
- There are no false positives

FSLAM Platform and Map Models

- The platform state is

$$\mathbf{x}_k = \begin{bmatrix} x_k & y_k & \psi_k \end{bmatrix}^\top$$

- The state of the ith landmark is

$$\mathbf{m}^i = \begin{bmatrix} u^i & v^i \end{bmatrix}^\top$$

Process and Observation Models

- The vehicle process models are now nonlinear,

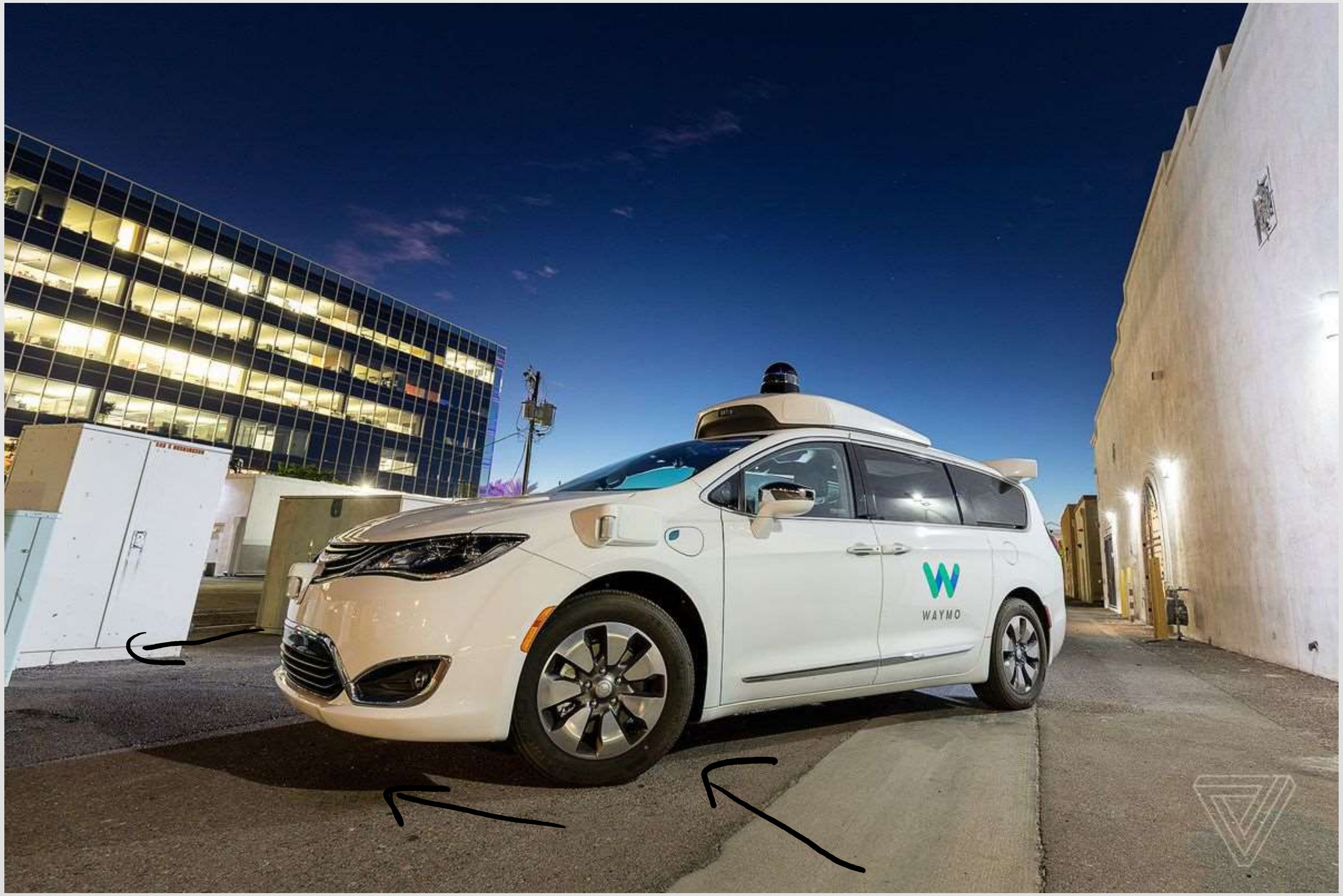
$$\mathbf{x}_k = \mathbf{f}\left[\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k\right]$$

$$\mathbf{z}_k^j = \mathbf{h}\left[\mathbf{x}_k, \mathbf{m}^{i_j}, \mathbf{w}_k^j\right]$$

- The "inverse observation model" is also nonlinear

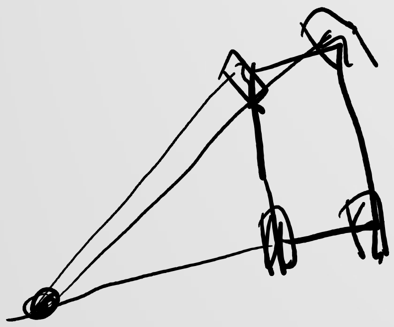
$$\mathbf{m}^{i_j} = \mathbf{g}\left[\mathbf{x}_k, \mathbf{z}_k^j, \mathbf{w}_k^j\right]$$

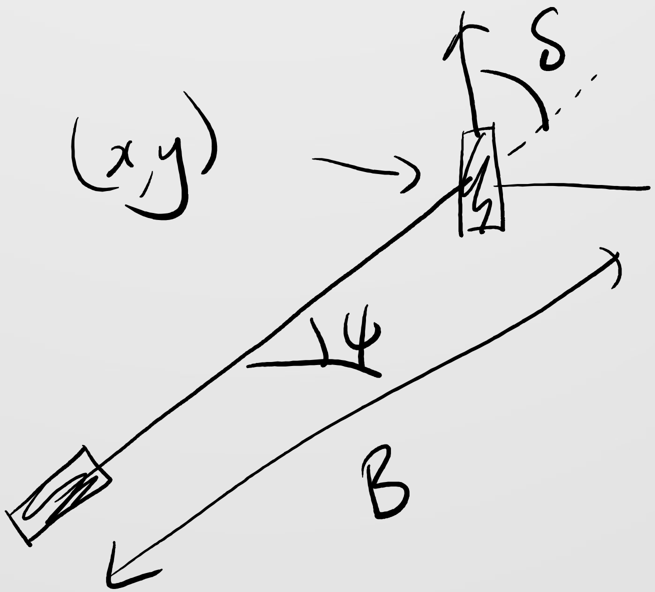
Steer Angle Differs from Heading



From [Riding in Waymo One, the Google spinoff's first self-driving taxi service]

Process Model





Process Model

- The control input is the wheel speed and front wheel steer angle,

$$\mathbf{u}_k = \begin{bmatrix} s_k & \delta_k \end{bmatrix}^\top$$

- Given a time step length of ΔT, the process model is

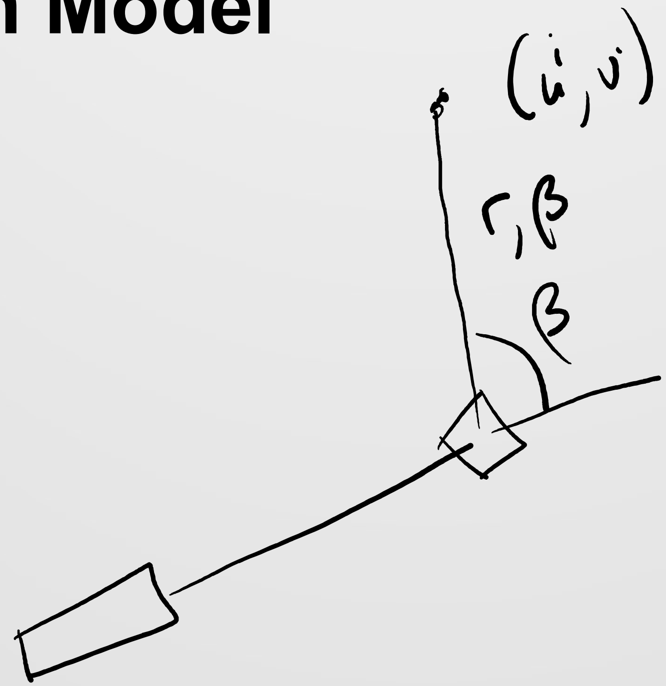
$(s_k + s_v)$ $\delta_k + v_s$

$$x_k = x_{k-1} + s_k \Delta T \cos\left(\psi_{k-1} + \delta_k\right)$$

$$y_k = y_{k-1} + s_k \Delta T \sin\left(\psi_{k-1} + \delta_k\right)$$

$$\psi_k = \psi_{k-1} + \frac{s_k \Delta T \sin \delta_k}{B}$$

Observation Model



Observation Model

- The observation of landmark i at time step k is the range bearing pair

$$\mathbf{z}_k^j = \begin{bmatrix} r_k^j & \beta_k^j \end{bmatrix}^\top + \omega$$

where

$$r_k^j = \sqrt{\left(u^{i_j} - x_k\right)^2 + \left(v^{i_j} - y_k\right)^2} \leftarrow$$

$$\beta_k^j = \underline{\tan^{-1}\left(\frac{v^{i_j} - y_k}{u^{i_j} - x_k}\right)} - \psi_k \leftarrow$$

Inverse Observation Model

r, β

x, y

(r, β)

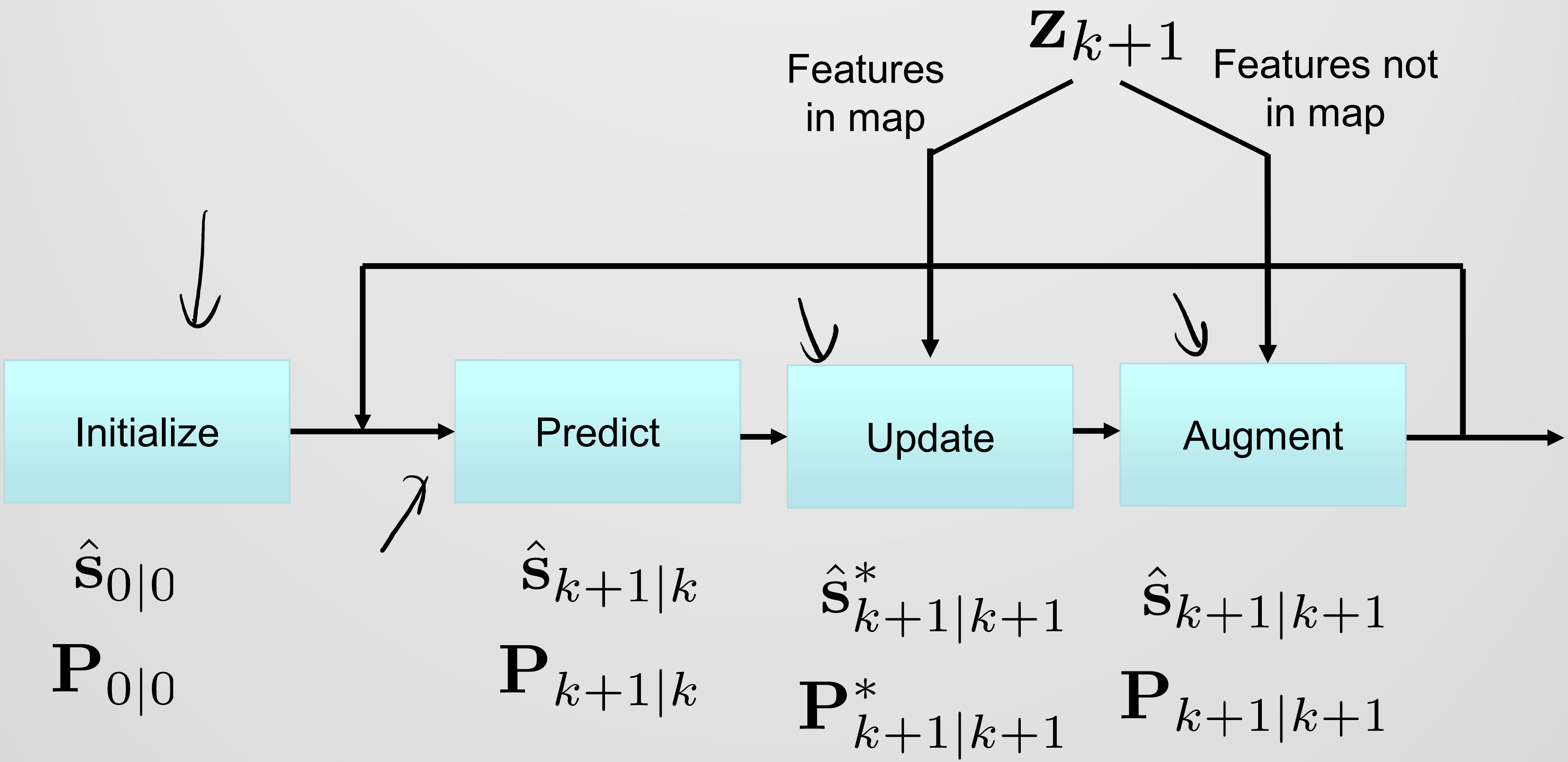
(x, y, ψ)

- The equations are

$$u^{i_j} = x_k + r_k^j \cos\left(\psi_k + \beta_k^j\right)$$

$$v^{i_j} = y_k + r_k^j \sin\left(\psi_k + \beta_k^j\right)$$

EKF-SLAM Kalman Filter Structure



EKF-SLAM Prediction

$$\hat{\mathbf{s}}_{k|k-1} = \begin{bmatrix} \mathbf{f}\left[\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k, \mathbf{0}\right] \\ \hat{\mathbf{m}}^{1}_{k-1} \\ \vdots \\ \hat{\mathbf{m}}^{N_{k-1}}_{k-1} \end{bmatrix}$$

$$E, f \quad E[x_k] = E[f(x_{k-1}, u_k, v_k)]$$

$$\simeq f(E[x_{k-1}], E[u_k], E[v_k])$$

EKF-SLAM Prediction

$$\mathbf{P}_{k+1|k} = \underline{\mathbf{F}_s \mathbf{P}_{k|k} \mathbf{F}_s^\top} + \underline{\mathbf{B}_s \mathbf{Q}_k \mathbf{B}_s^\top}$$

$$\mathbf{F}_s = \begin{bmatrix} \nabla \mathbf{f_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \qquad \mathbf{B}_s = \begin{bmatrix} \nabla_{\mathbf{v}} \mathbf{f} \\ \mathbf{0} \end{bmatrix}$$

$$\nabla f_x = \frac{\partial x_r}{\partial x}$$

EKF-SLAM Augmentation

$$\hat{\mathbf{s}}_{k|k-1} = \begin{bmatrix} \cancel{\mathbf{f}\left[\overset{\hat{x}^{A}_{k|k}}{\hat{\mathbf{x}}_{k-1|k-1}}, \mathbf{u}_k, \mathbf{0}\right]} \\ \hat{\mathbf{m}}^{1}_{k-1} \\ \vdots \\ \hat{\mathbf{m}}^{N_{k-1}}_{k-1} \end{bmatrix}$$

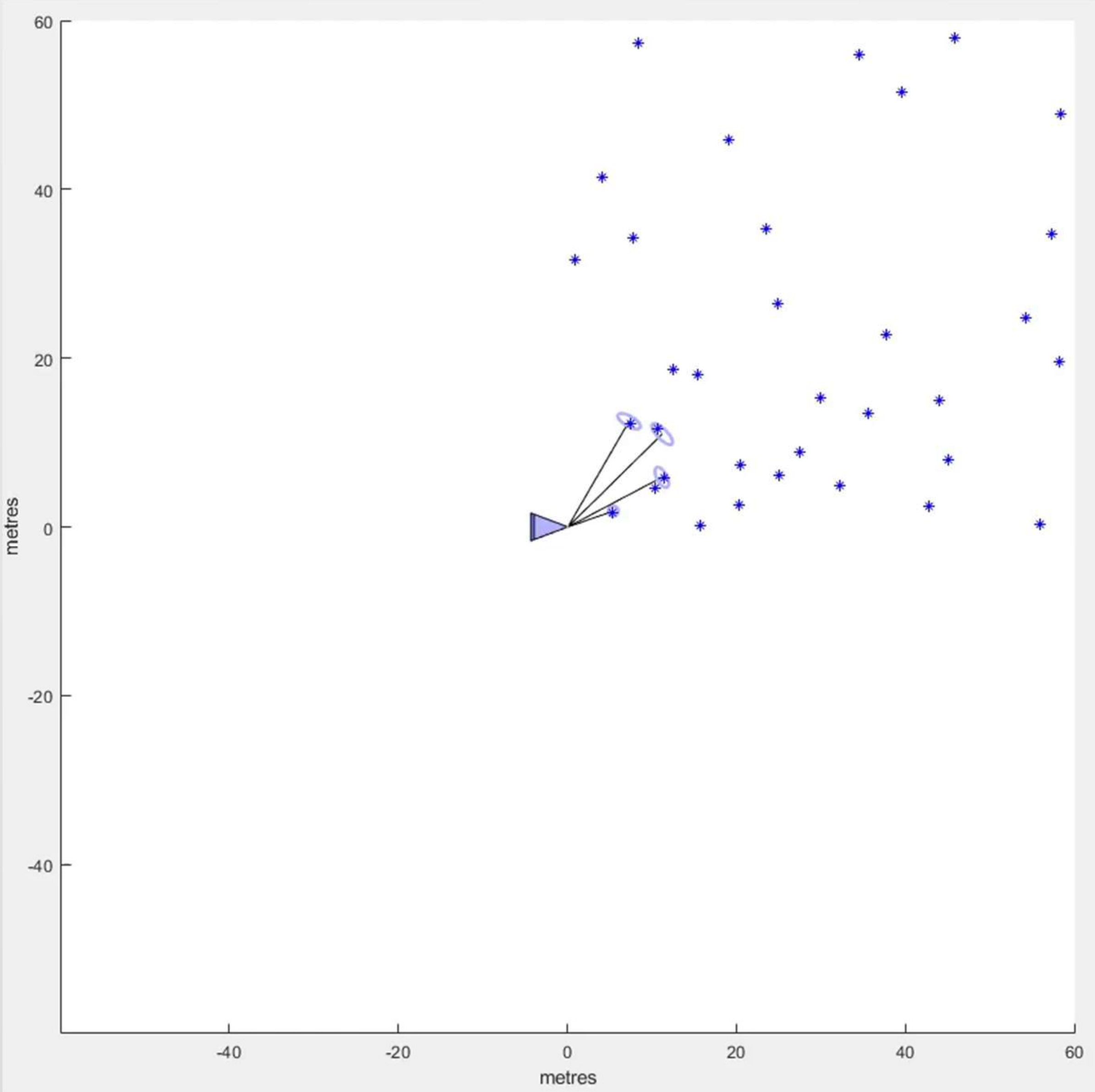
$$g\left[\hat{x}^{*}_{k|k}, 2, 0\right]$$

$$T$$

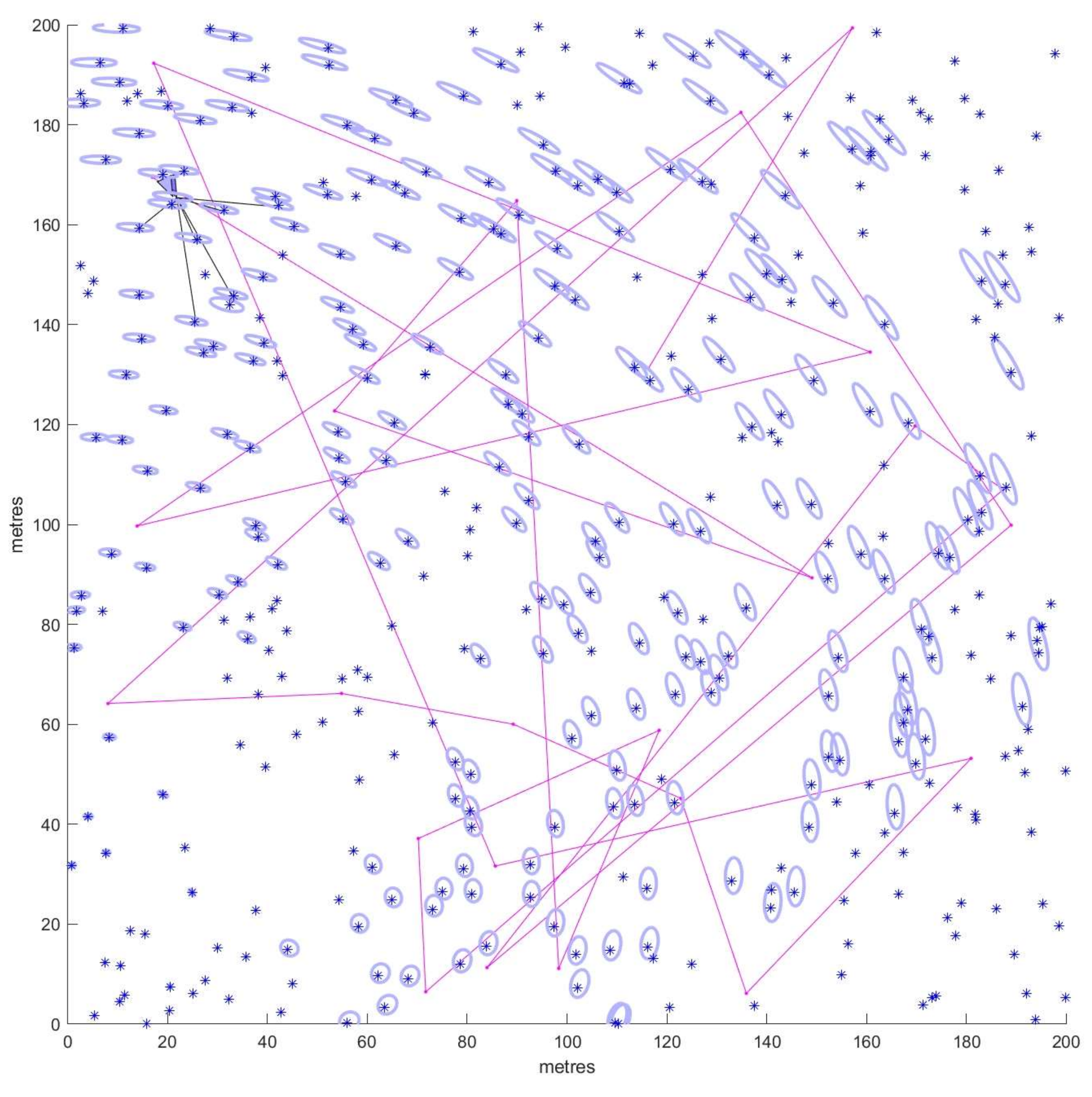
EKF-SLAM Augmentation

$$\mathbf{P}_{k|k} = \mathbf{A}_s \mathbf{P}^*_{k|k} \mathbf{A}_s^\top + \mathbf{J}_s \mathbf{R}_k \mathbf{J}_s^\top$$

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{I} \\ \nabla_{\mathbf{x}} \mathbf{g} \end{bmatrix} \qquad \mathbf{J}_s = \begin{bmatrix} \mathbf{0} \\ \nabla_{\mathbf{w}} \mathbf{g} \end{bmatrix}$$



60
40
20
0
-20
-40
metres
-40
-20
0
20
40
60
metres


200
180
160
140
120
100
80
60
40
20
0
metres
0
20
40
60
80
100
120
140
160
180
200
metres

Theoretical Properties of EKF SLAM

- Assuming linearity holds and the linear approximations are “good”, you get basically the same theoretical performance as BeadSLAM:
  - All the landmarks converge to a rigid structure
  - The covariance in all the landmarks is a function of the initial uncertainty of the vehicle state
- A lot of analysis looked at:
  - How do we scale it up?
  - Does it work that well?

Issues with EKF SLAM

- Scalability
- Performance

Scalability Issues

$$\hat{S}_{k|k} = \begin{bmatrix} \hat{x}_k \\ \hat{m}_1 \\ \vdots \\ \hat{m}_n \end{bmatrix}$$

- The main challenge is that as the map gets bigger, the dimension of the state gets bigger

- This means that the state space and the covariance matrix grow larger and larger

- This causes both the computational cost to increase

Storage Costs

$$\hat{\mathbf{s}}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k^1 \\ \vdots \\ \hat{\mathbf{m}}_k^{N_k} \end{bmatrix}$$

n

$\frac{n(n-1)}{2}$

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}^{\mathbf{xx}} & \mathbf{P}^{\mathbf{xm}^1} & \mathbf{P}^{\mathbf{xm}^2} & \cdots & \mathbf{P}^{\mathbf{xm}^{N_k}} \\ \mathbf{P}^{\mathbf{m}^1\mathbf{x}} & \mathbf{P}^{\mathbf{m}^1\mathbf{m}^1} & \cancel{\mathbf{P}^{\mathbf{m}^1\mathbf{m}^2}} & \cdots & \cancel{\mathbf{P}^{\mathbf{m}^1\mathbf{m}^{N_k}}} \\ \mathbf{P}^{\mathbf{m}^2\mathbf{x}} & \mathbf{P}^{\mathbf{m}^2\mathbf{m}^1} & \mathbf{P}^{\mathbf{m}^2\mathbf{m}^2} & \cdots & \mathbf{P}^{\mathbf{m}^2\mathbf{m}^{N_k}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{\mathbf{m}^{N_k}\mathbf{x}} & \mathbf{P}^{\mathbf{m}^{N_k}\mathbf{m}^1} & \mathbf{P}^{\mathbf{m}^{N_k}\mathbf{m}^2} & \cdots & \mathbf{P}^{\mathbf{m}^{N_k}\mathbf{m}^{N_k}} \end{bmatrix}_{k|k}$$

Computational Costs (Prediction)

$$\hat{\mathbf{s}}_{k+1|k} = \begin{bmatrix} \mathbf{f}\left[\hat{\mathbf{x}}_{k|k}, \mathbf{u}_{k+1}, \mathbf{0}\right] \\ \hat{\mathbf{m}}_k^1 \\ \vdots \\ \hat{\mathbf{m}}_k^{N_k} \end{bmatrix}$$

$\underline{FPF^T + BQB^T}$

$\underline{F = \begin{bmatrix} \nabla f & 0 \\ 0 & I \end{bmatrix}}$

$\underline{\begin{bmatrix} \nabla f \\ 0 \end{bmatrix}}$

Computational Costs (Prediction)

- The predicted covariance is given by

O(N)

$\leftarrow 3\times 2$

$$
\begin{bmatrix}
\mathbf{F}\mathbf{P}^{\mathbf{xx}}\mathbf{F}^\top + \mathbf{Q} & \mathbf{F}\mathbf{P}^{\mathbf{x}\mathbf{m}^1} & \mathbf{F}\mathbf{P}^{\mathbf{x}\mathbf{m}^2} & \dots & \mathbf{F}\mathbf{P}^{\mathbf{x}\mathbf{m}^{N_k}} \\
\mathbf{P}^{\mathbf{m}^1\mathbf{x}}\mathbf{F}^\top & \mathbf{P}^{\mathbf{m}^1\mathbf{m}^1} & \mathbf{P}^{\mathbf{m}^1\mathbf{m}^2} & \dots & \mathbf{P}^{\mathbf{m}^1\mathbf{m}^{n_k}} \\
\mathbf{P}^{\mathbf{m}^2\mathbf{x}}\mathbf{F}^\top & \mathbf{P}^{\mathbf{m}^2\mathbf{m}^1} & \mathbf{P}^{\mathbf{m}^2\mathbf{m}^2} & \dots & \mathbf{P}^{\mathbf{m}^2\mathbf{m}^{n_k}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{P}^{\mathbf{m}^{n_k}\mathbf{x}}\mathbf{F}^\top & \mathbf{P}^{\mathbf{m}^{n_k}\mathbf{m}^1} & \mathbf{P}^{\mathbf{m}^{N_k}\mathbf{m}^2} & \dots & \mathbf{P}^{\mathbf{m}^{n_k}\mathbf{m}^{n_k}}
\end{bmatrix}
$$

O(N)

Computational Costs (Update)

- The update is given by the Kalman filter equations

$$\hat{\mathbf{s}}^{*}_{k+1|k+1} = \hat{\mathbf{s}}_{k+1|k} + \mathbf{W}_{k+1}\boldsymbol{\nu}_{k+1|k}$$

$$\mathbf{P}^{*}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \underline{\mathbf{W}_{k+1}\mathbf{S}_{k+1|k}\mathbf{W}^{\top}_{k+1}}$$

$(I-WH)P$

$O(N^3)$

$O(N^2 M)$

Scalability and Sub-Mapping

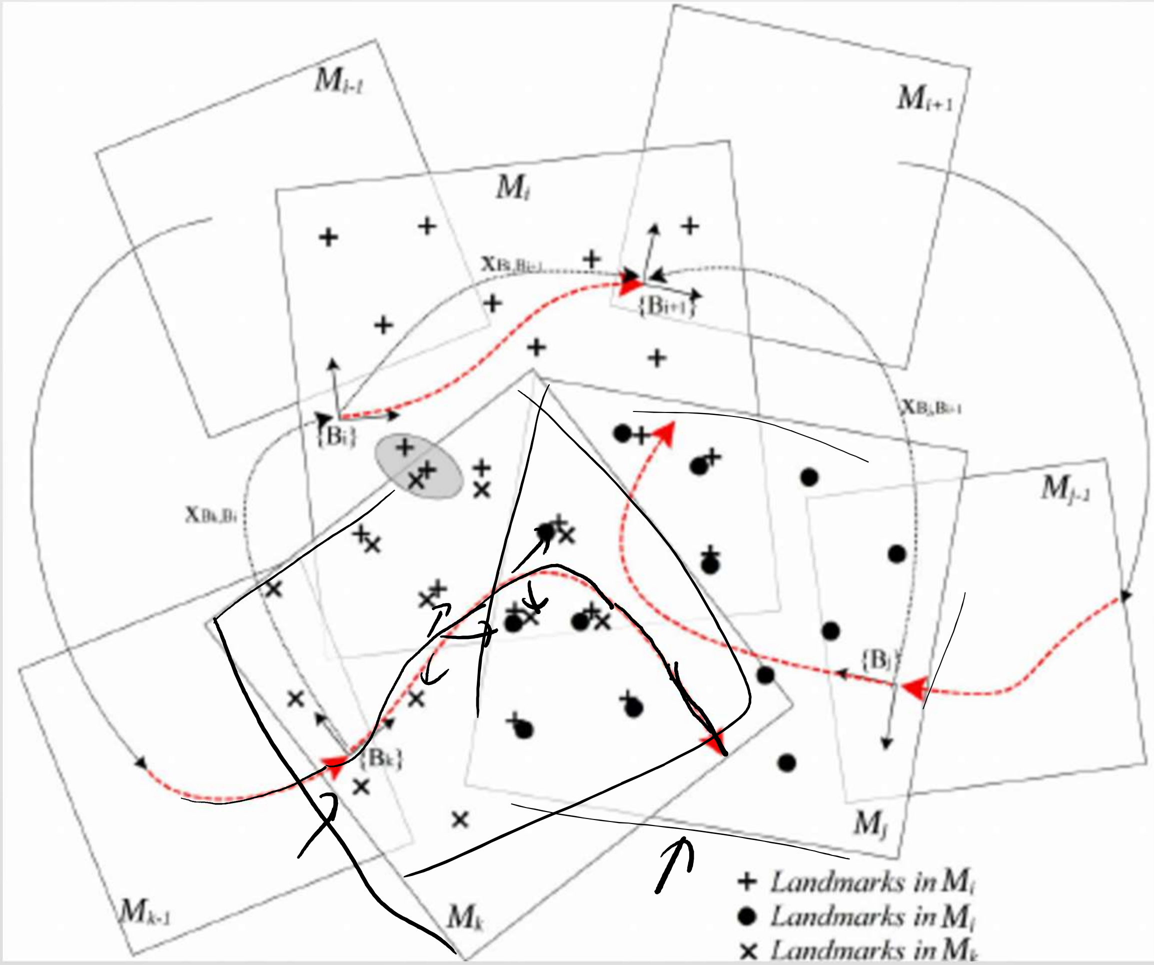
$O(N^2)$ t

$O((N/t)^2 t)$

$O(N^2/t)$

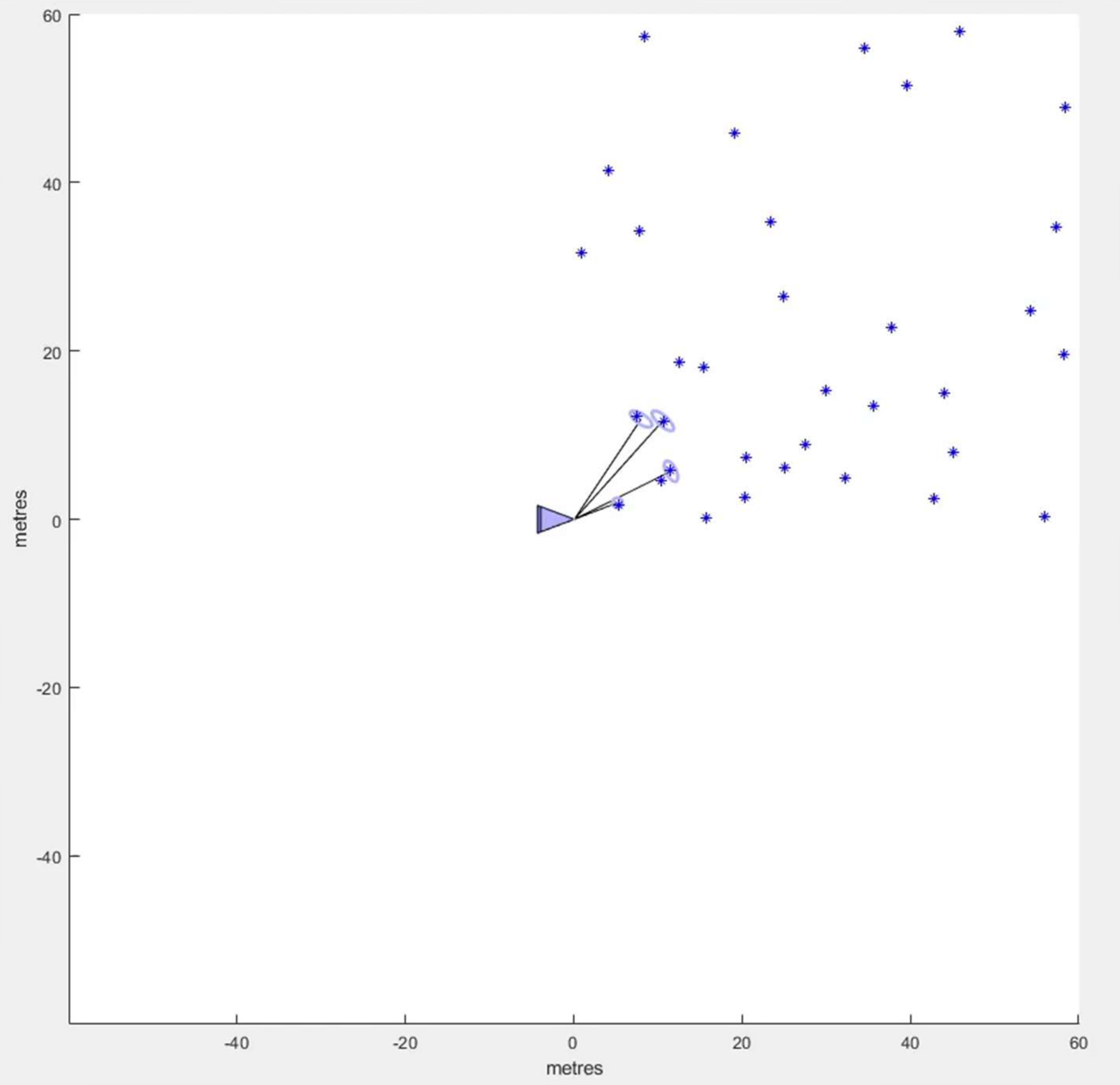
$O((N/t)^3)$

ATLAS

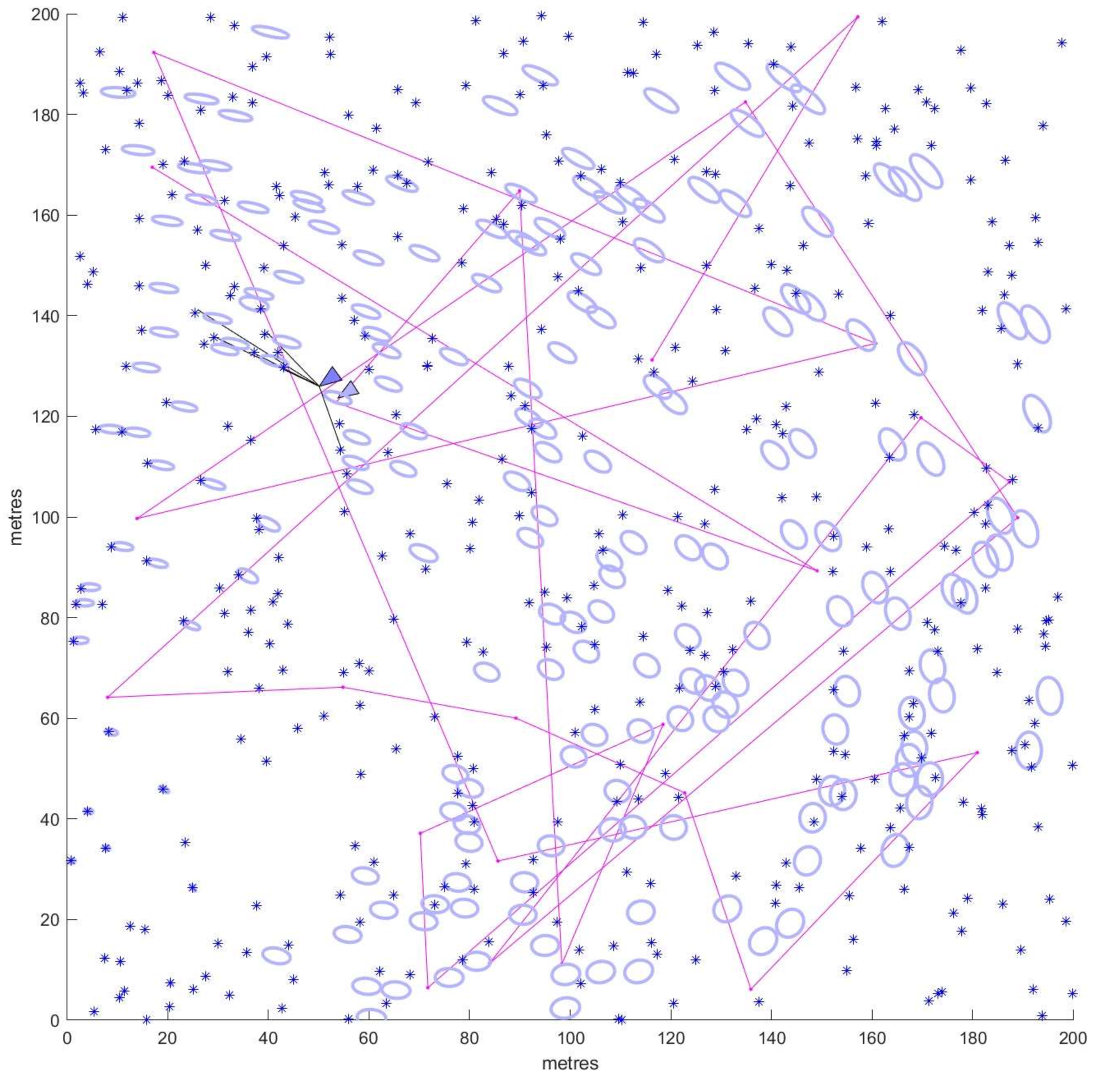


Performance

- The SLAM algorithm should produce covariance-consistent estimates of the vehicle and landmark locations



60
40
20
0
-20
-40
metres
-40
-20
0
20
40
60
metres

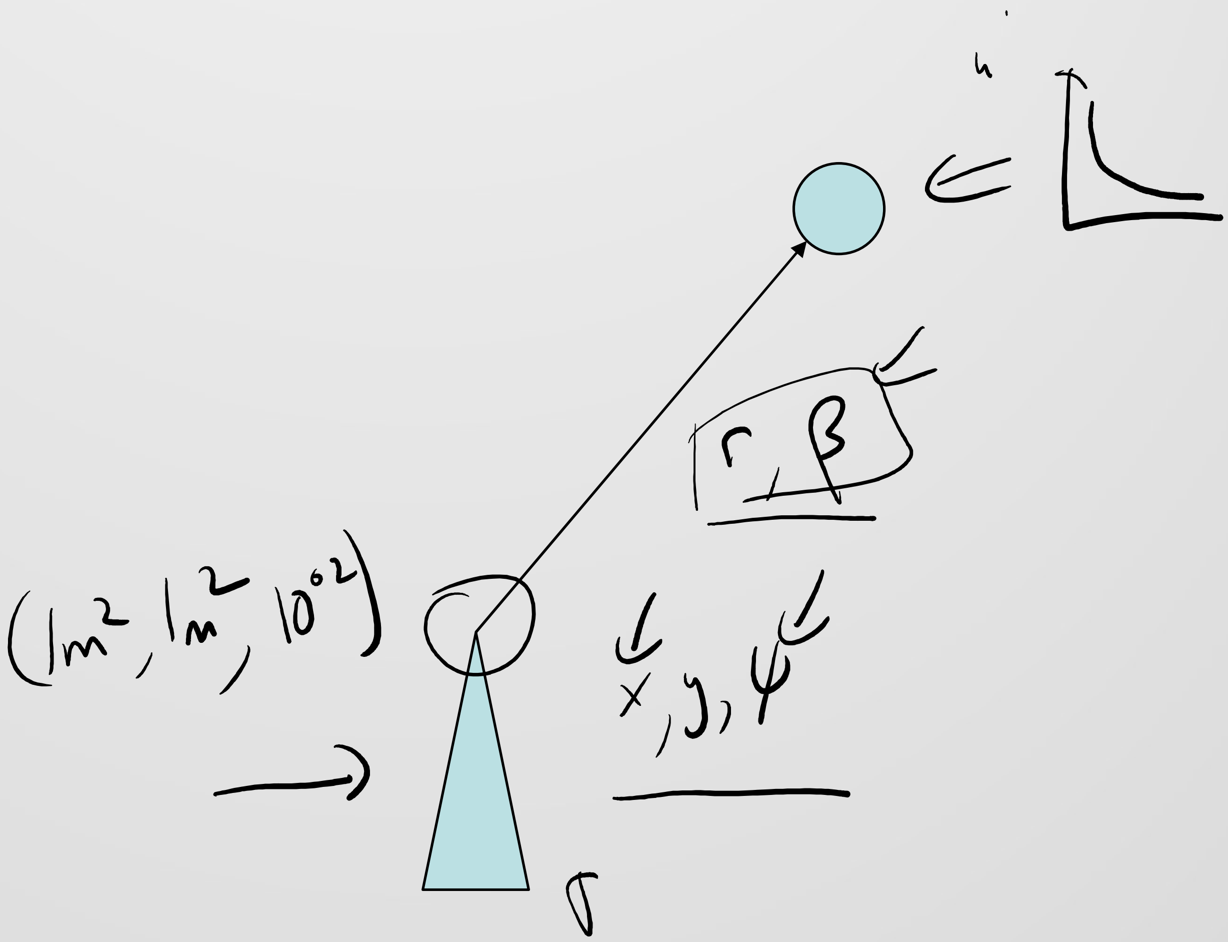


metres
metres
0
20
40
60
80
100
120
140
160
180
200

Why Did EKF SLAM Drift Off?

- The Kalman filter critically depends upon the correct relationship between the state estimate and the covariance matrix
- These are clearly going wrong
- Why is this the case?

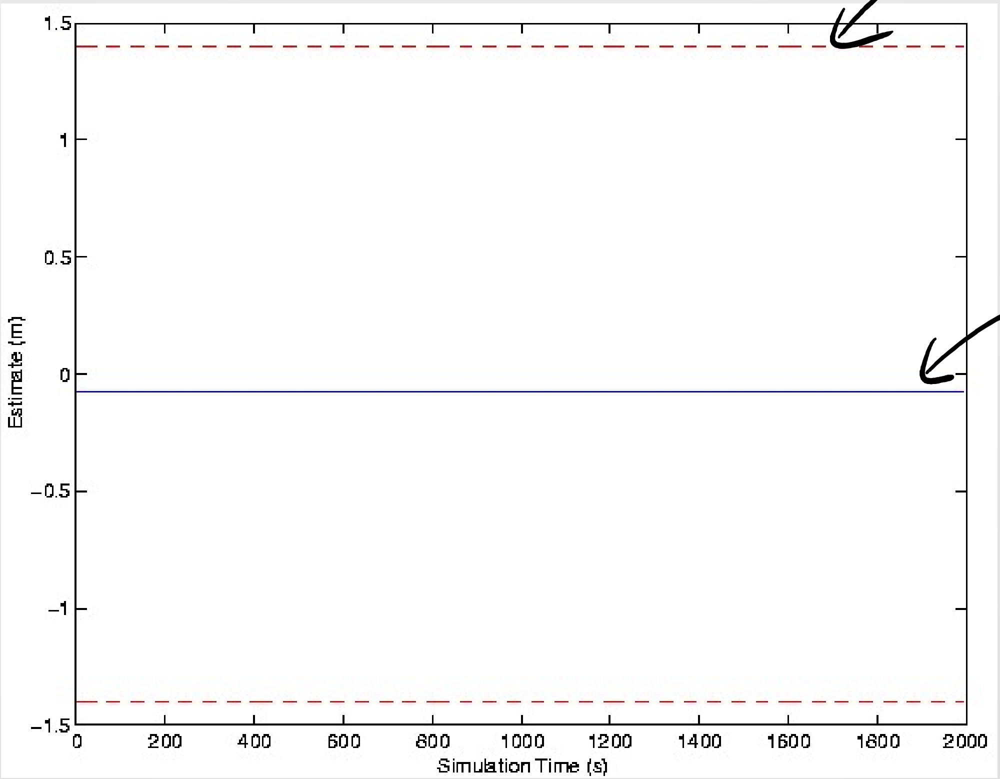
Effects of Angular Errors on Covariances



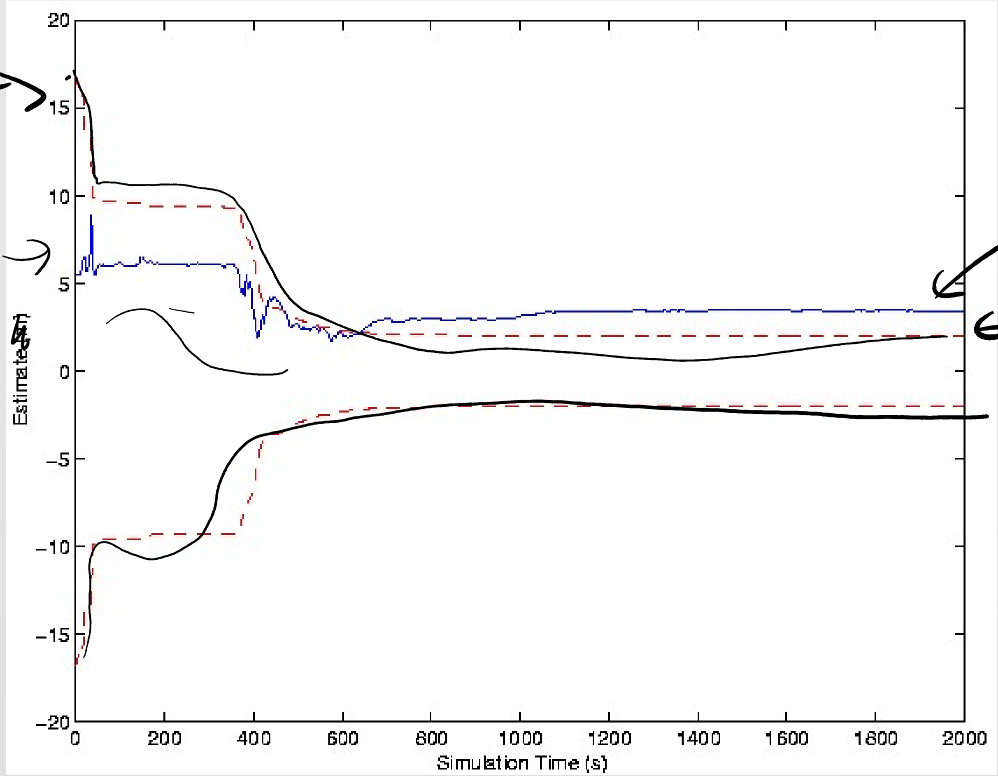
Position Estimate and Covariance

$2\sqrt{P_{\phi xx}}$

$\pm 2\sigma$



Orientation Estimate and Covariance



$\beta = \text{atan2}(\ldots) - \psi$

Why the Phantom Update

- Recall that the update equations are

$$\hat{\mathbf{s}}^{*}_{k+1|k+1} = \hat{\mathbf{s}}_{k+1|k} + \underline{\mathbf{W}_{k+1}}\boldsymbol{\nu}_{k+1|k}$$

$$\mathbf{P}^{*}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{W}_{k+1}\mathbf{S}_{k+1|k}\mathbf{W}^{\top}_{k+1}$$

Why the Phantom Update

$$\hat{\mathbf{s}}^*_{k+1|k+1} = \hat{\mathbf{s}}_{k+1|k} + \mathbf{W}_{k+1}\boldsymbol{\nu}_{k+1|k}$$

$$\begin{matrix} x \\ y \\ \psi \end{matrix} \qquad w = \begin{bmatrix} w_x \\ w_y \\ w_\psi \end{bmatrix} \neq 0$$

$= 0$

Kalman Filter Update Equations

$$\boldsymbol{\nu}_{k+1|k} = \mathbf{z}_k - \mathbf{H}_s \hat{\mathbf{s}}_{k+1|k}$$

$$\mathbf{C}_{k+1|k} = \boxed{\mathbf{P}_{k+1|k} \mathbf{H}_s^\top}$$

$$\mathbf{S}_{k+1|k} = \mathbf{H}_s \mathbf{C}_{k+1|k} + \mathbf{R}$$

$$\mathbf{W}_{k+1|k} = \mathbf{C}_{k+1|k} \mathbf{S}_{k+1|k}^{-1}$$

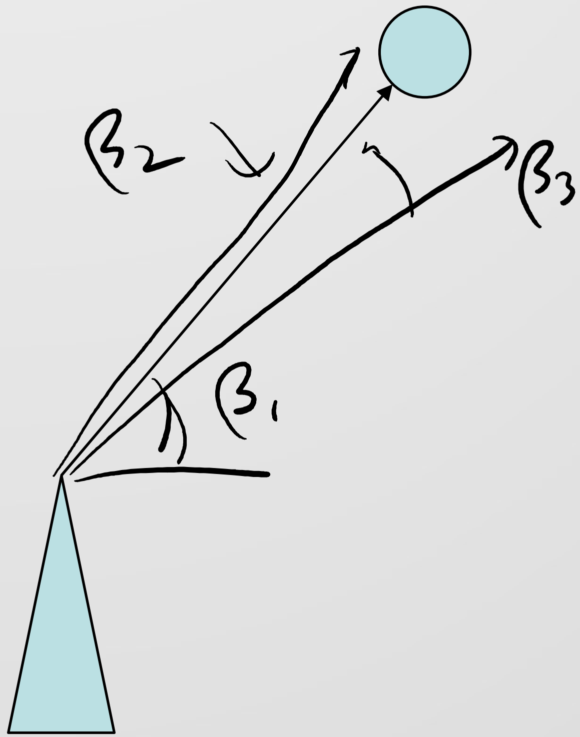
$P_{x\nu}$

Kalman Filter Update Equations

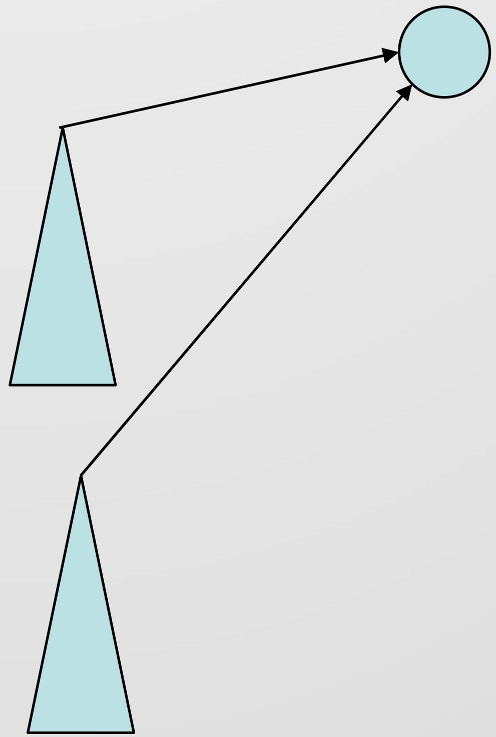
- A closed form analysis was carried out to determine the properties of $\boldsymbol{C}_{k+1|k}$
- The term associated with orientation update is non-zero if the angular observation is different from the angular observation used to initialize the landmark

Effects of Angular Errors on Covariances

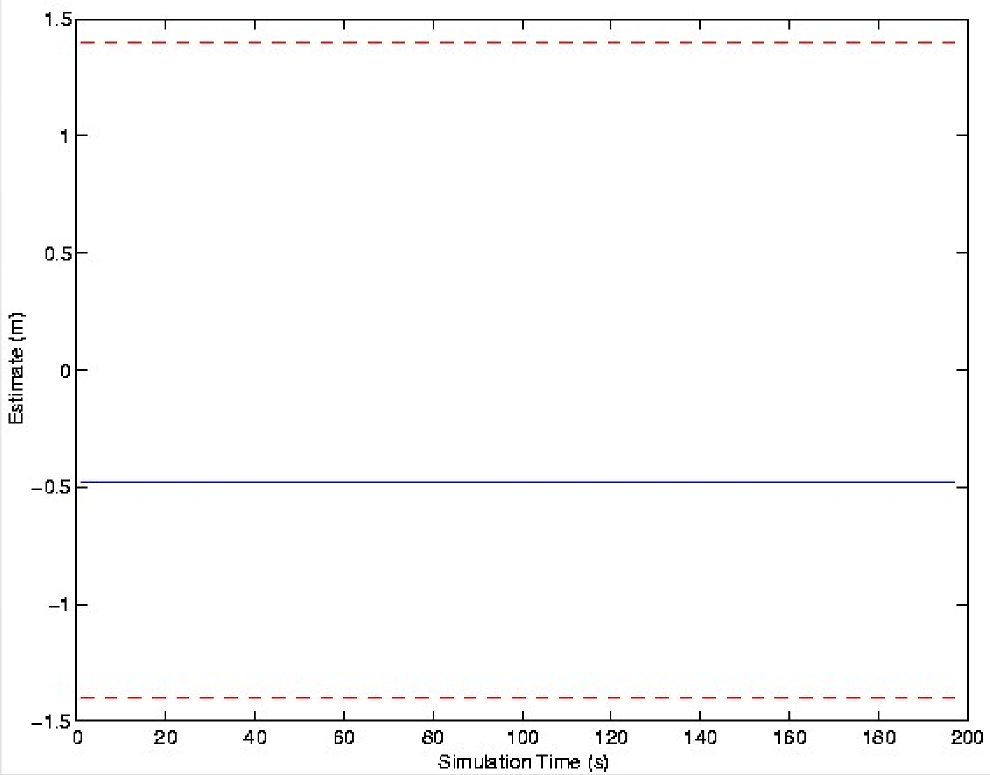
W_ψ



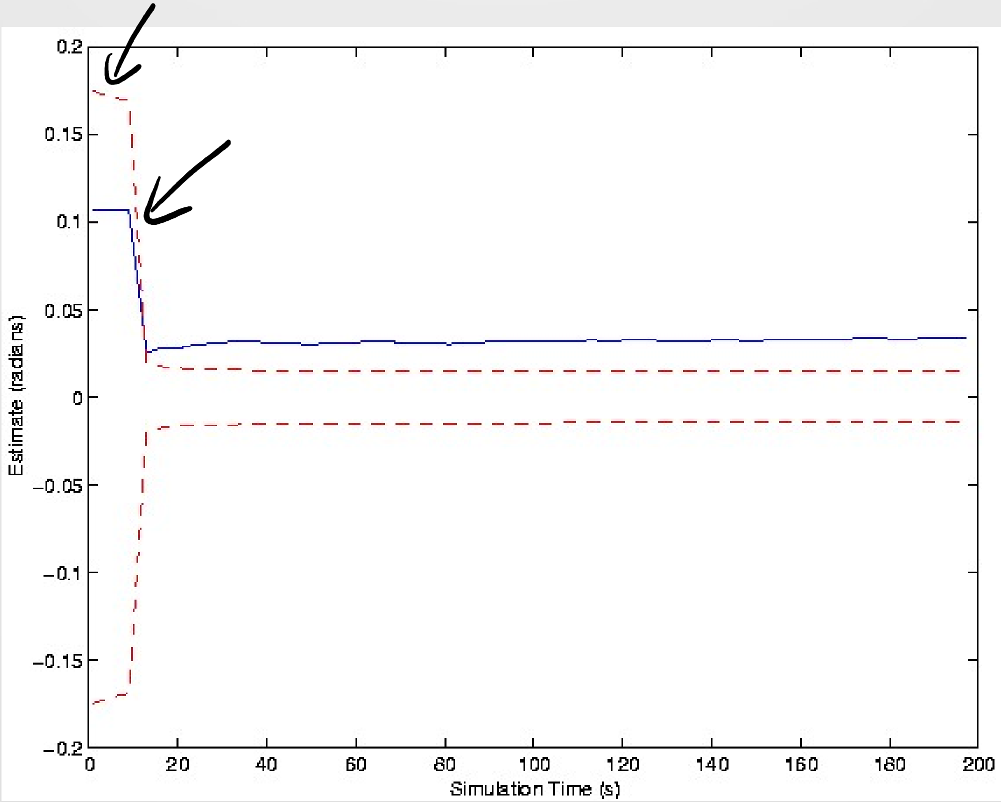
Effects of Angular Errors on Covariances



Position Estimate and Covariance



Orientation Estimate and Covariance



Kalman Filter Update Equations

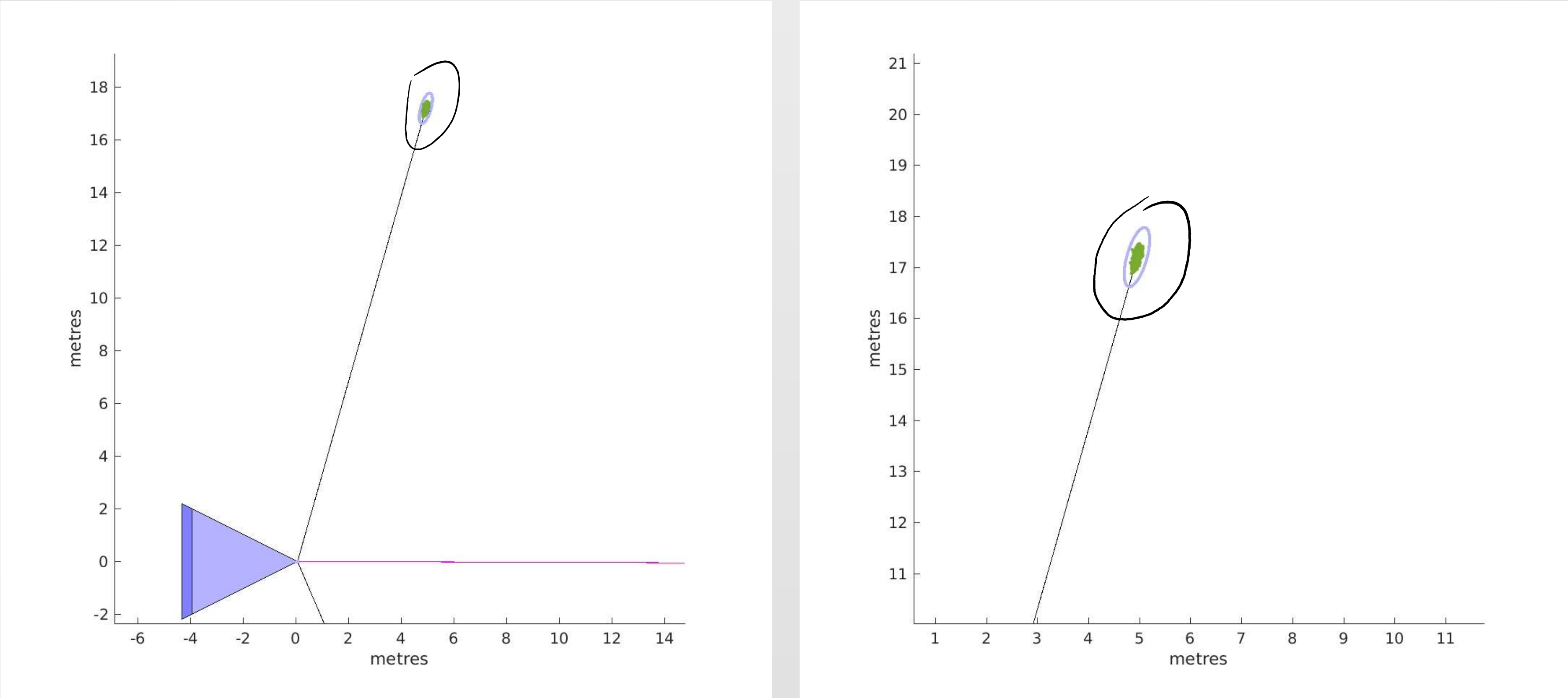
- A closed form analysis was carried out to determine the properties of $\boldsymbol{C}_{k+1|k}$
- The term associated with orientation update is non-zero if the angular observation is different from the angular observation used to initialize the landmark
- This suggests the failure is just a glitch and that things such as higher order Kalman filters will work better

Attempts to Fix the Problem

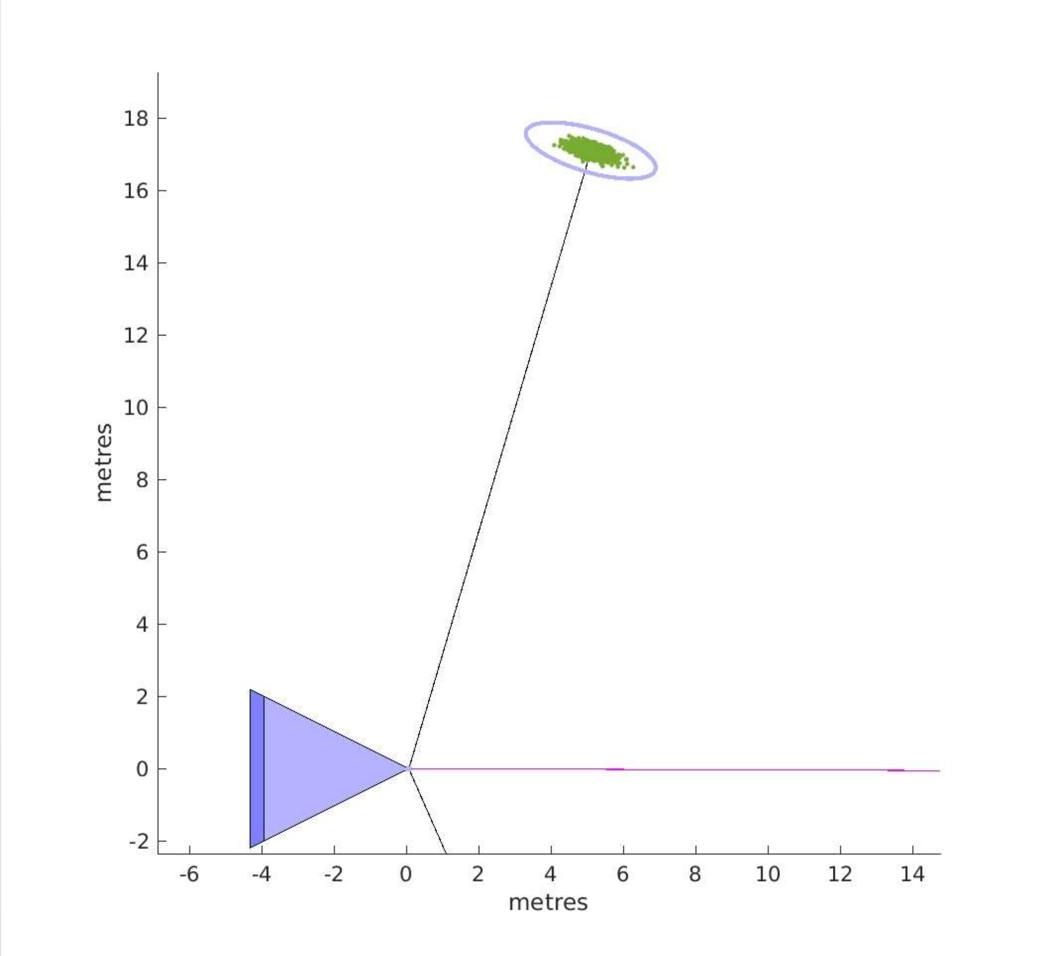
- Higher order Taylor Series Expansions
- Exact analytical solutions given Gaussian assumptions
- However, none of these approaches worked
- The reason is that correlations are not good enough to measure the structure of angular dependencies

Visualising Covariance (0.5 Deg Std)

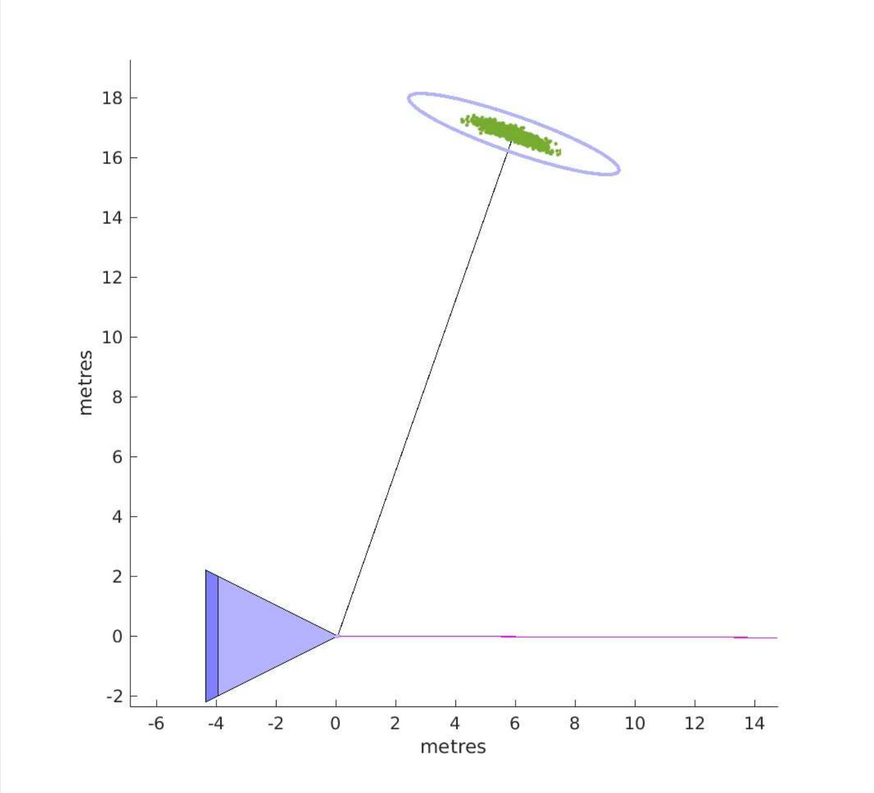
x+r cos β   y+r sin β   β ~ 0.5°

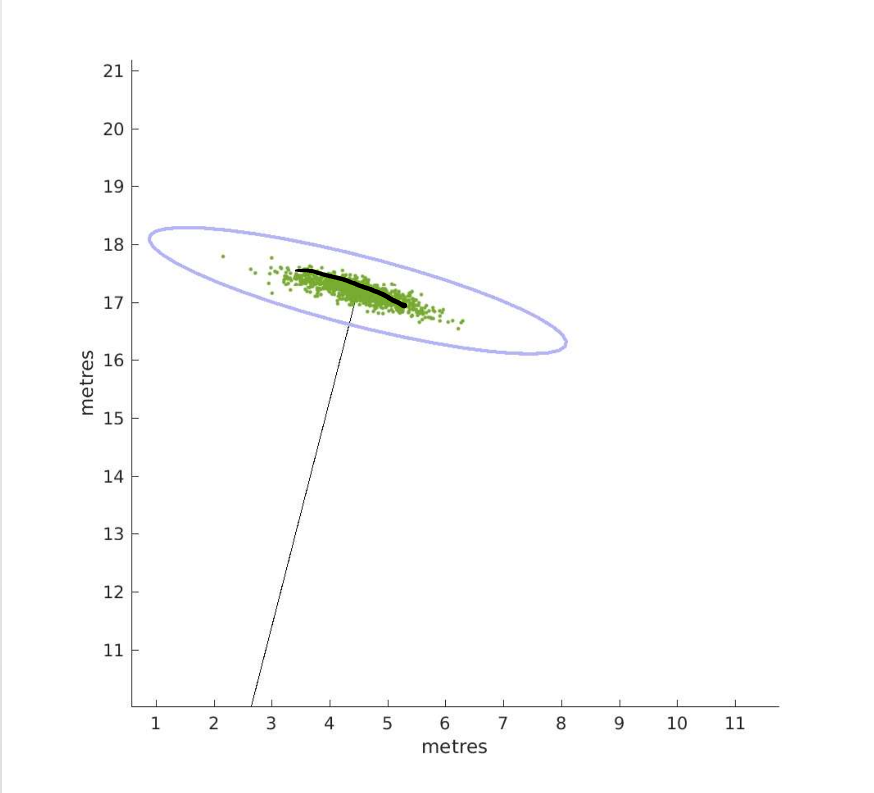


Visualising Covariance (1 Deg Std)

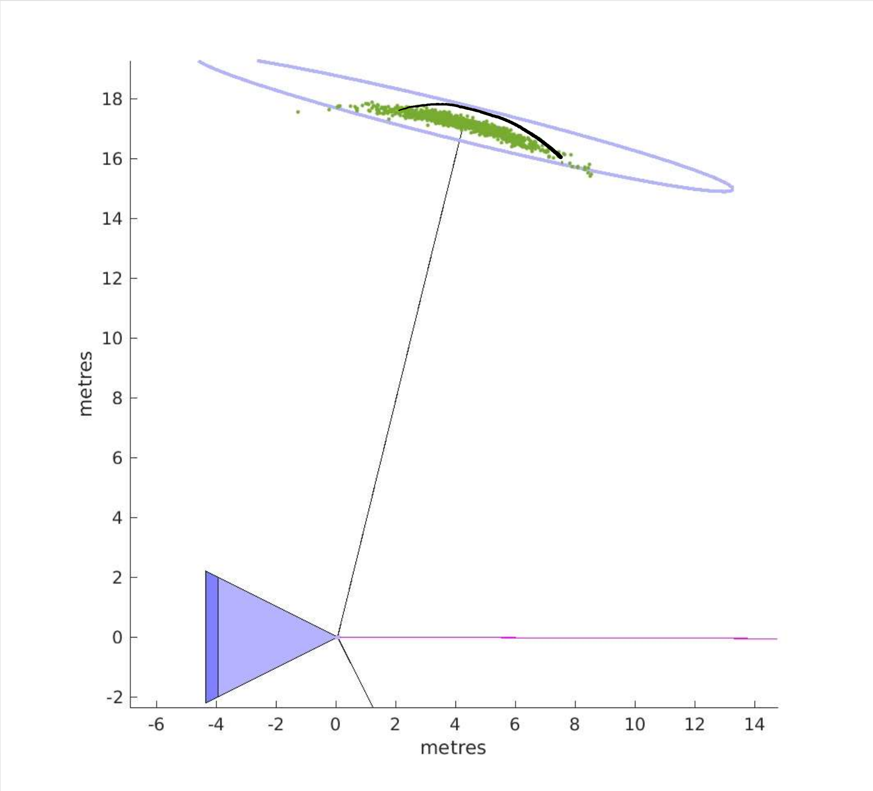


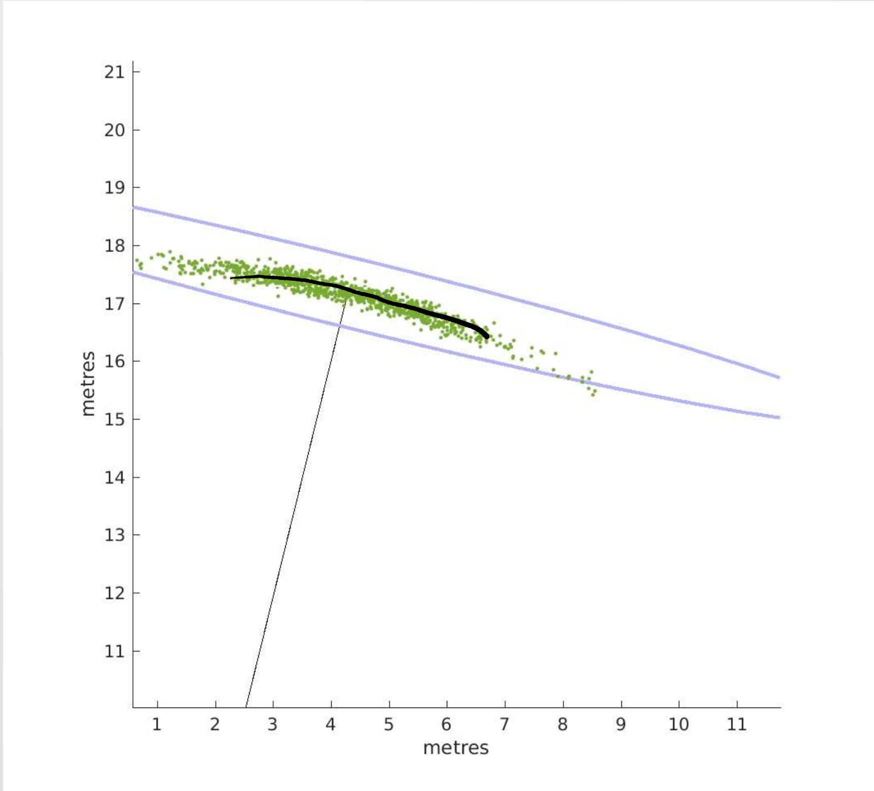
Visualising Covariance (2 Deg Std)



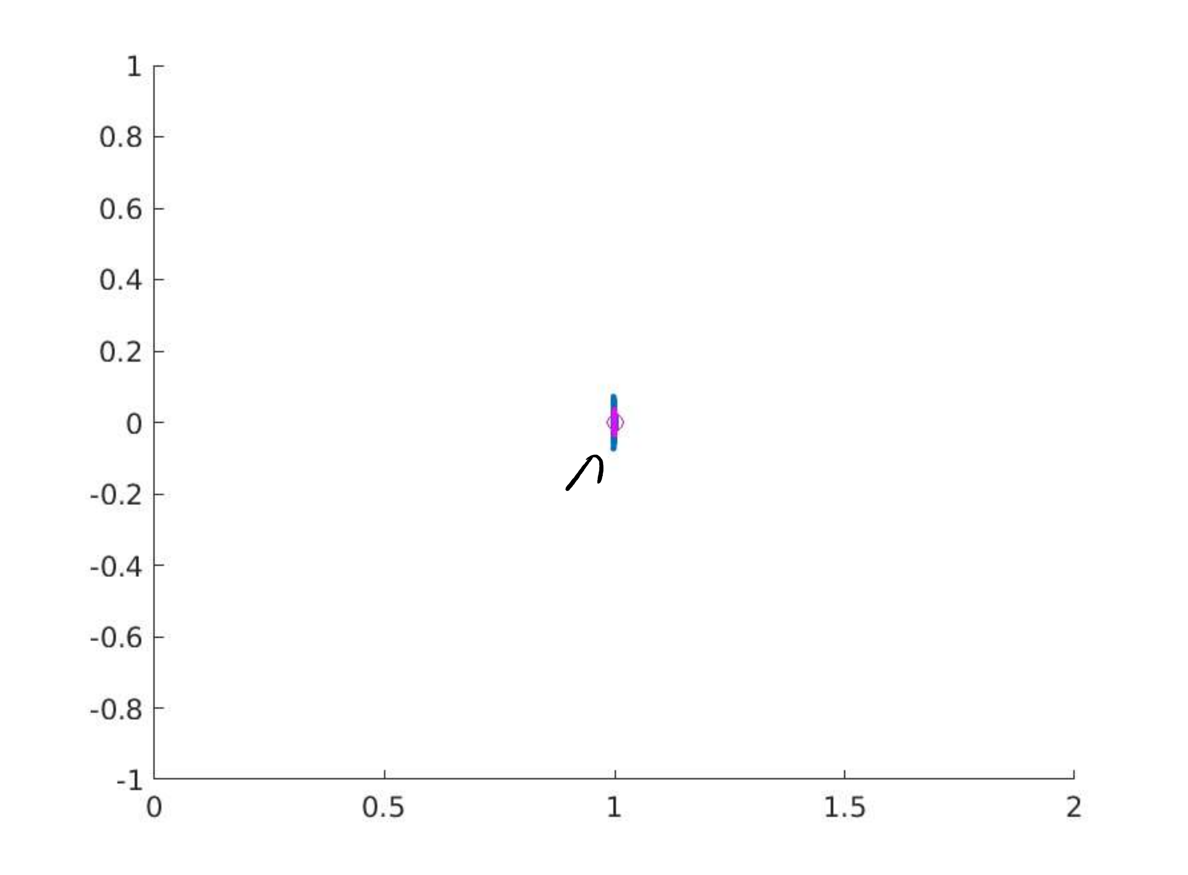


Visualising Covariance (5 Deg Std)

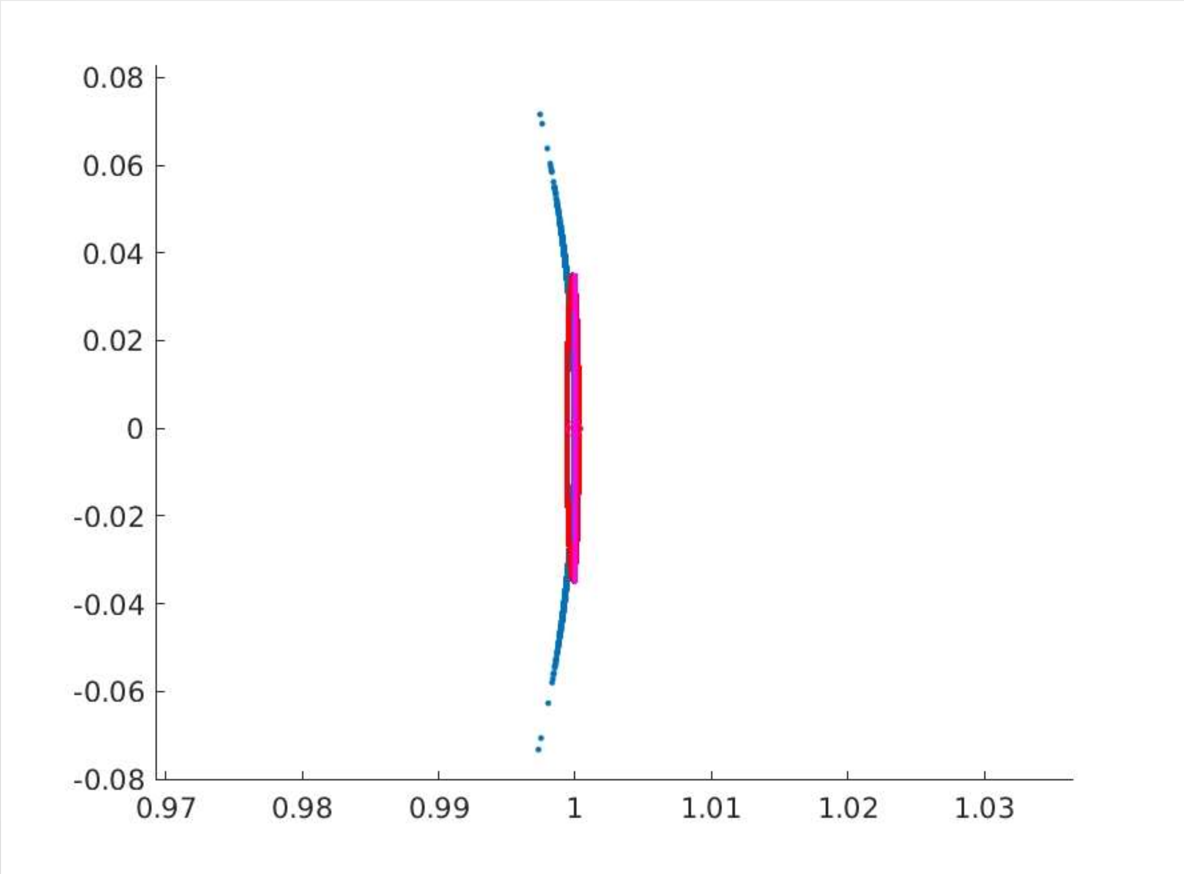




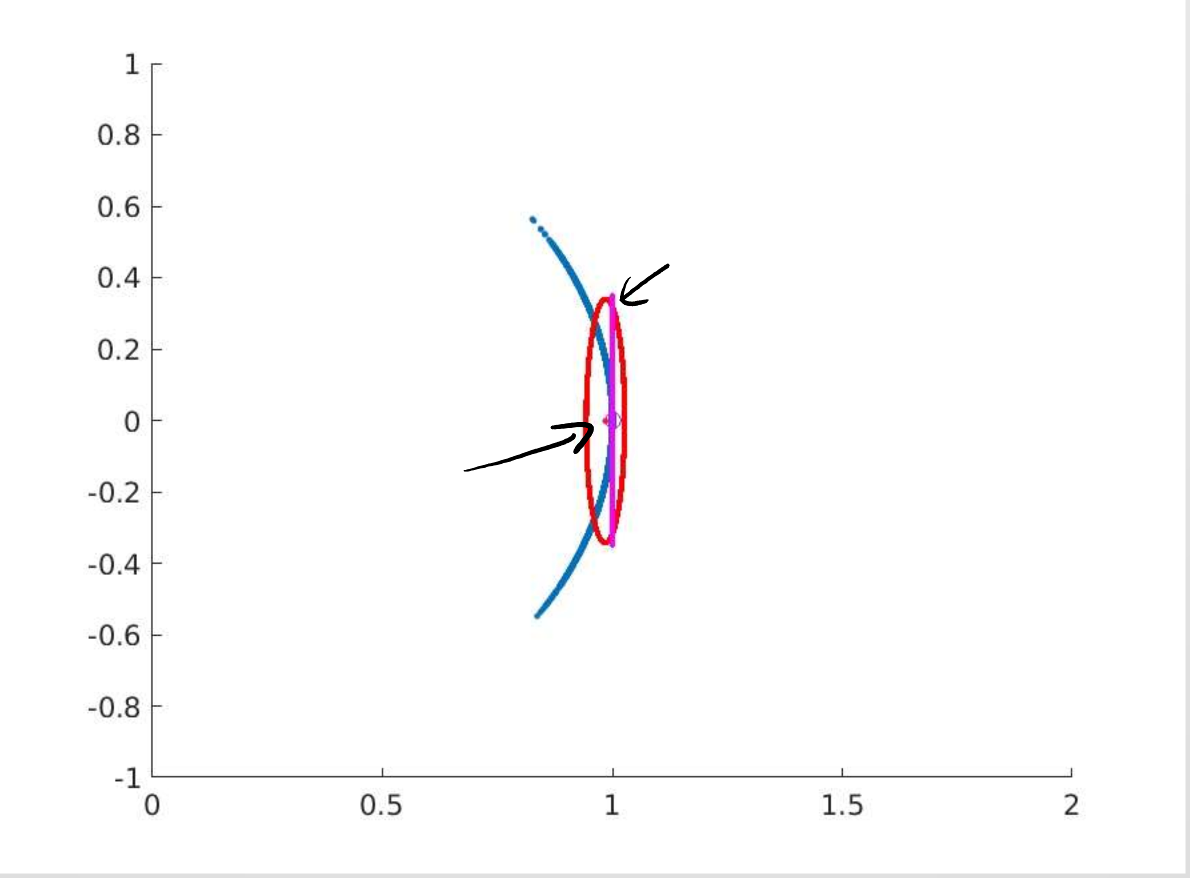
1 Degree Standard Deviation



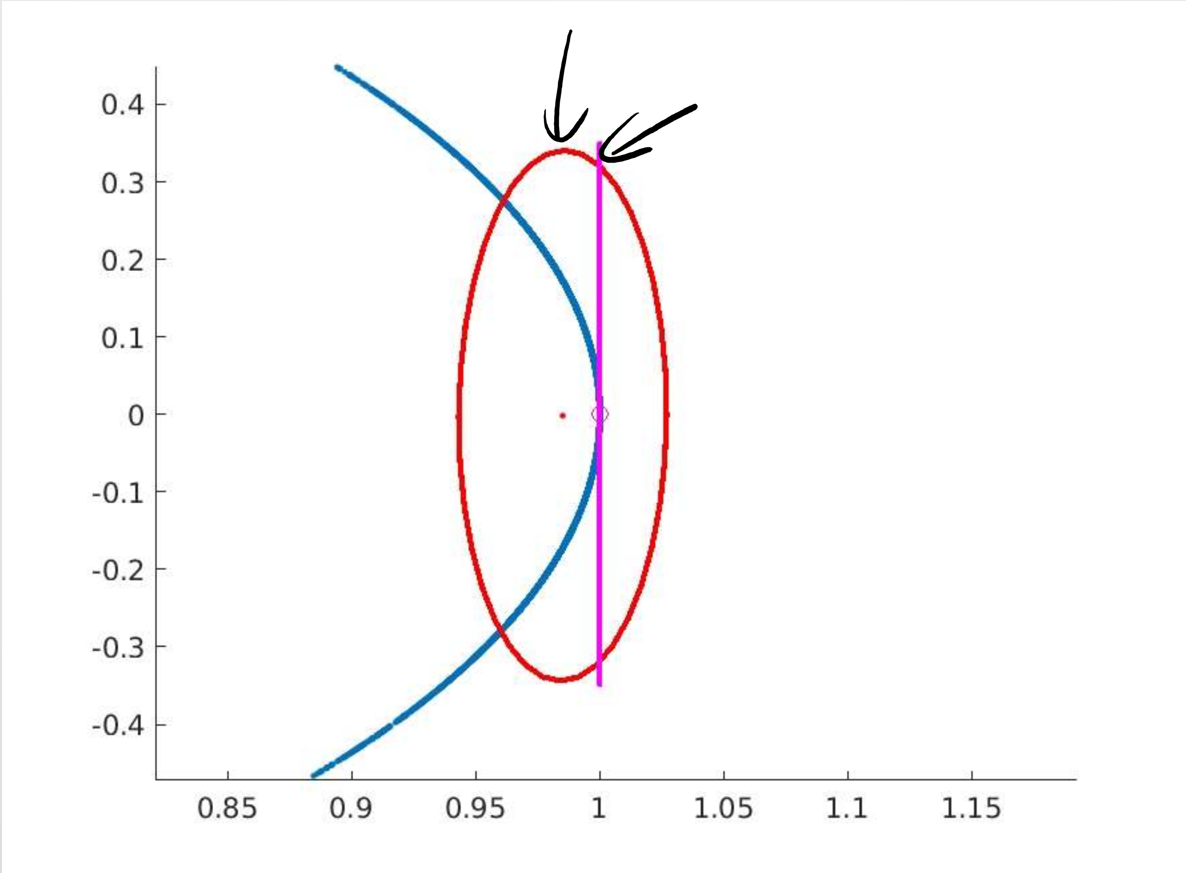
1 Degree Standard Deviation (Zoomed)



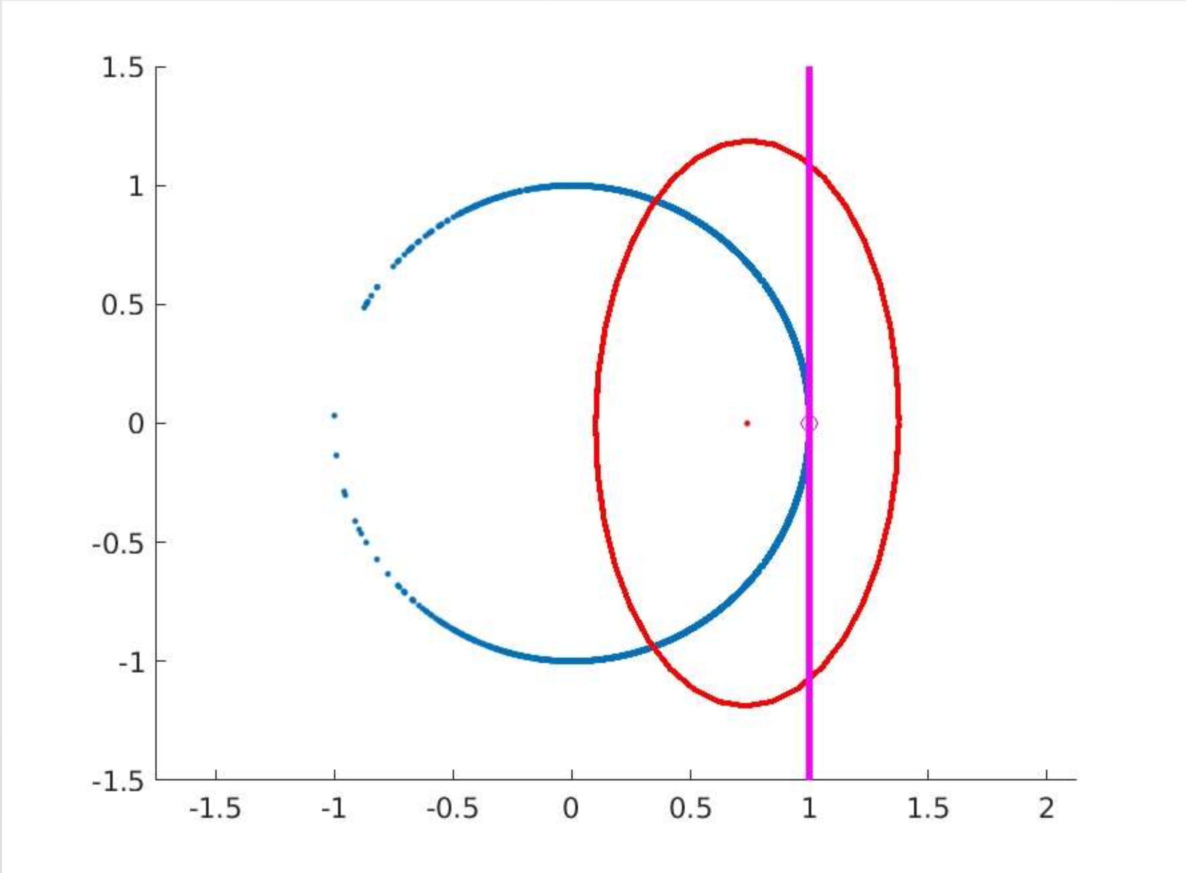
10 Degree Standard Deviation



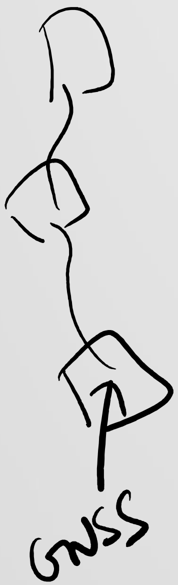
10 Degree Standard Deviation (Zoomed)

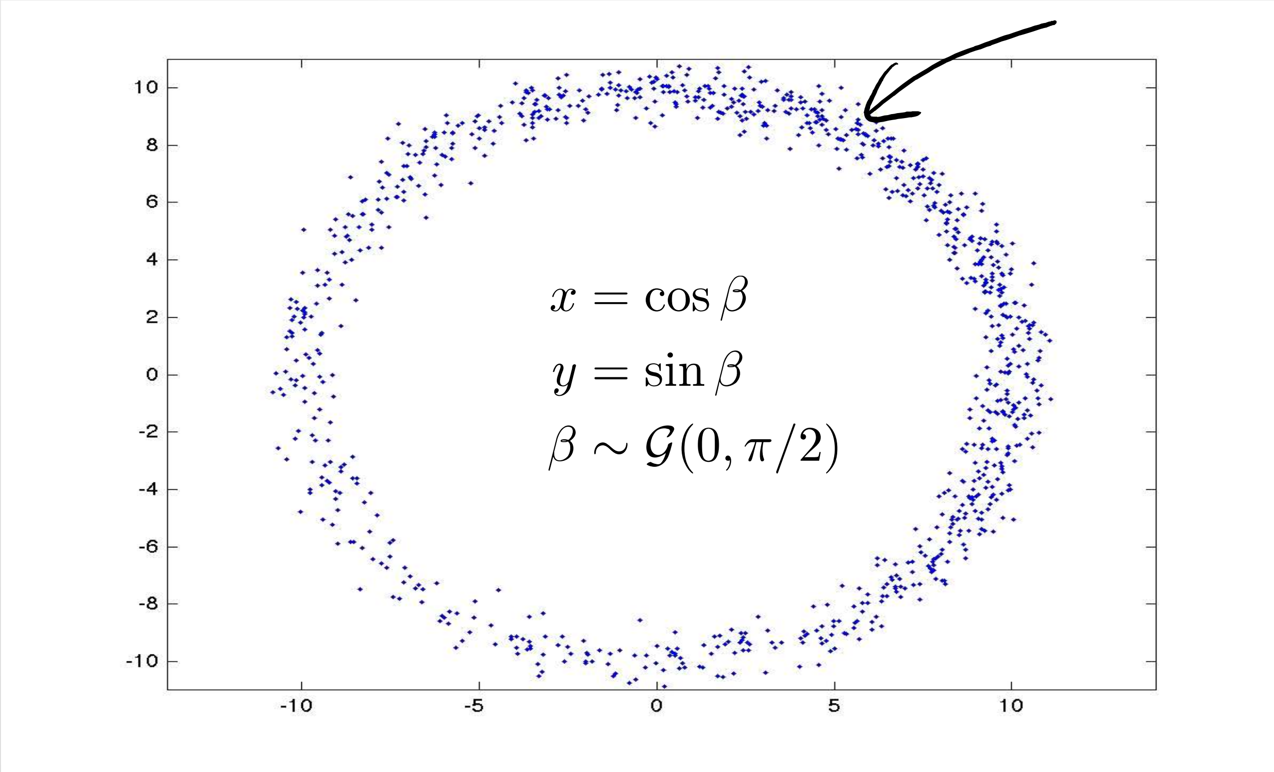


45 Degree Standard Deviation



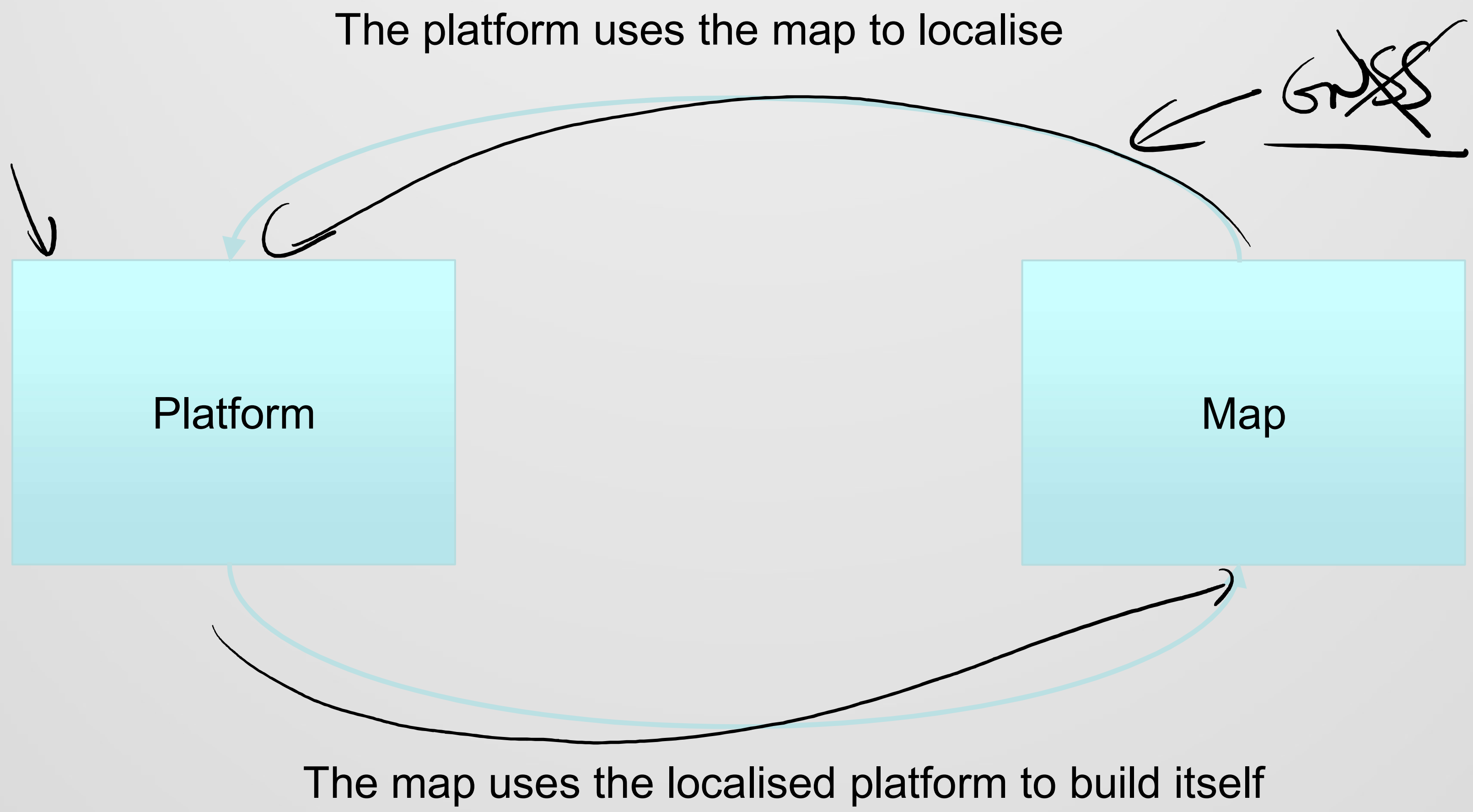
Correlations and Angular Dependencies





The random variables in the x and y direction are almost uncorrelated but are not independent

Self-Mapping Chicken and Egg Problem



Implications

- Because the errors arise because of strangeness in the way the cross correlations are computed, they are independent of magnitude of noise
- It turns out that things like higher order moment expansion or closed for solutions do not address the issue
- Therefore, we need a more sophisticated way to