Day 30

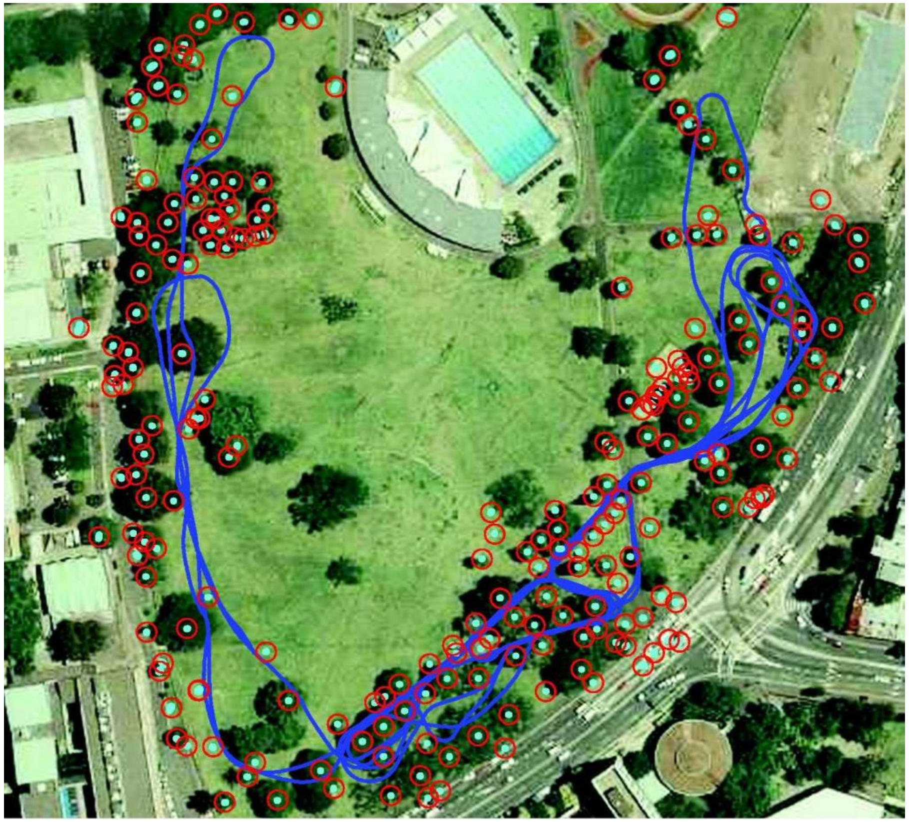
Simultaneous Localization and Mapping

SLAM

- simultaneous localization and mapping
  - one of the most fundamental problems in mobile robotics
- a robot is exploring an unknown static environment
  - robot is given sensor measurements and control inputs
    - does not have a map
    - does not know its pose

SLAM

- robot must acquire a map while simultaneously localizing itself relative to the map
  - harder than just localizing
    - has no map
  - harder than just mapping
    - does not know its pose



E. Nebot http://www-personal.acfr.usyd.edu.au/nebot/victoria_park.htm

Online SLAM

- in the online SLAM problem, we wish to estimate
  - the current pose of the robot x_t and
  - the map variables m
- we are given
  - the sensor measurements $z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ and
  - the control inputs $u_{1:t} = \{u_1, u_2, \ldots, u_t\}$

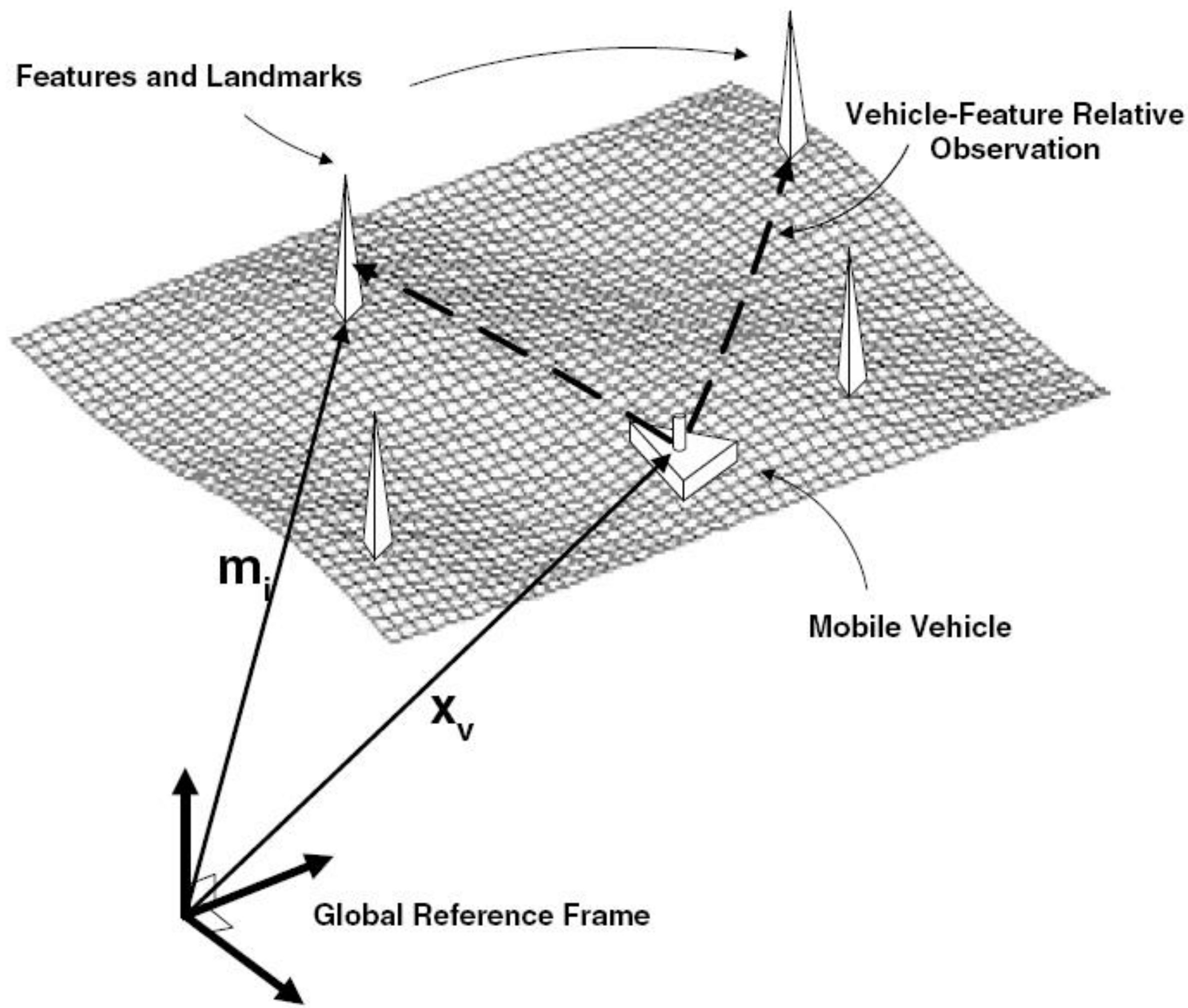
Online SLAM

- the online SLAM problem is often expressed in a probabilistic framework
  - compute the posterior probability

$$p(x_k, m \mid z_{1:k}, u_{1:k})$$

  - what is the probability density function of the robot's current pose and the map given the history of sensor measurements and control inputs?

Landmark-Based SLAM



A Simple Landmark-Based SLAM Problem

- given
  - a directionless robot (i.e., don't care about orientation) that moves in controlled but noisy steps
  - n fixed landmarks
  - the robot can measure all of the landmarks all of the time in a controlled order
  - the robot measures the relative offset from its position to each landmark

Plant Model

- state; dimension $2n + 2$

$$x_t = \begin{bmatrix} x \\ y \\ p_{1,x} \\ p_{1,y} \\ p_{2,x} \\ p_{2,y} \\ \vdots \\ p_{n,x} \\ p_{n,y} \end{bmatrix}$$

robot location

landmark 1 location

landmark 2 location

landmark *n* location

Plant Model

- control input; dimension $2n + 2$

$$u_t = \begin{bmatrix} \Delta x \\ \Delta y \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

robot step

*all zeros because no control is applied to landmarks

Plant Model

- plant noise; dimension $2n + 2$

$$\varepsilon_t = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

noise in the control input (*additive)

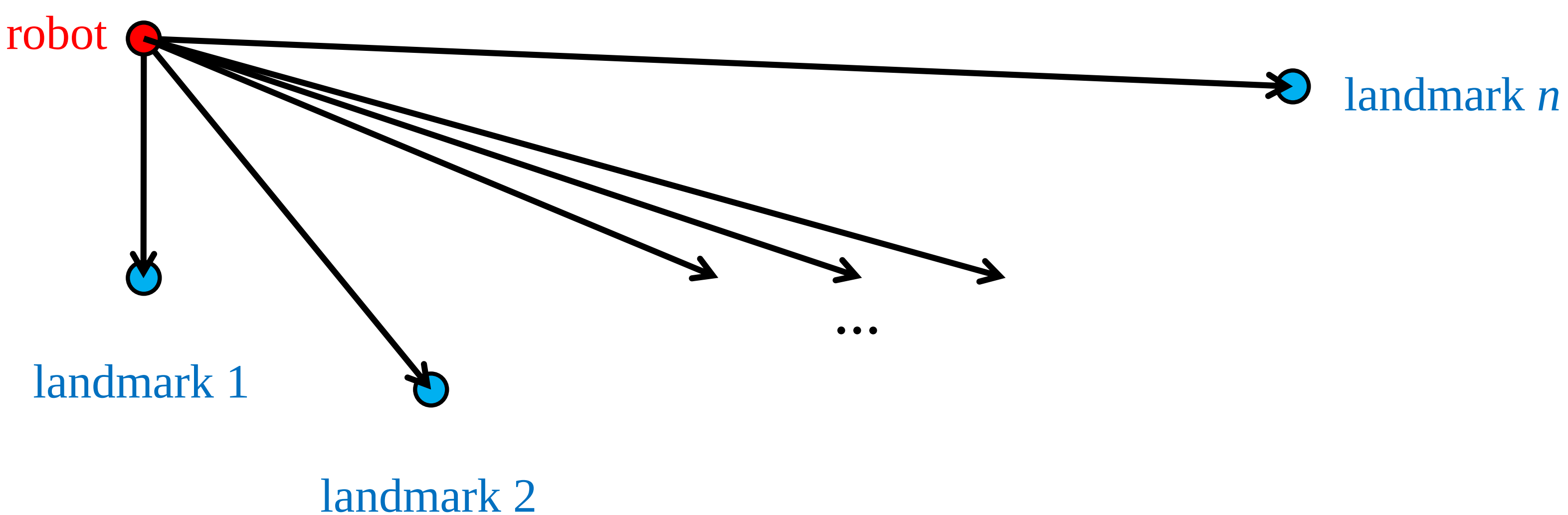
*all zeros because landmarks are static

Plant Model

$$
\begin{aligned}
x_{t+1} &= Ax_t + Bu_t + \varepsilon_t \\
&= I_{2n+2}\, x_t + I_{2n+2}\, u_t + \varepsilon_t \\
&= x_t + u_t + \varepsilon_t
\end{aligned}
$$

Measurement Model

- the robot can measure all of the landmarks all of the time in a controlled order
- the robot measures the relative offset from its position to each landmark



Measurement Model

- dimension $2n$

$$z_t = \begin{bmatrix} p_{1,x} - x \\ p_{1,y} - y \\ p_{2,x} - x \\ p_{2,y} - y \\ \vdots \\ p_{n,x} - x \\ p_{n,y} - y \end{bmatrix}$$

robot offset to landmark 1

robot offset to landmark 2

robot offset to landmark *n*

Measurement Model

- dimension $2n$

$$
z_t = \underbrace{\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ & & & & & & \dots & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} x \\ y \\ p_{1,x} \\ p_{1,y} \\ p_{2,x} \\ p_{2,y} \\ \vdots \\ p_{n,x} \\ p_{n,x} \end{bmatrix}}_{x_t}
$$

Measurement Model

- measurement noise; dimension $2n$

$$w_t = \begin{bmatrix} w_{1,x} \\ w_{1,y} \\ w_{2,x} \\ w_{2,y} \\ \vdots \\ w_{n,x} \\ w_{n,y} \end{bmatrix}$$

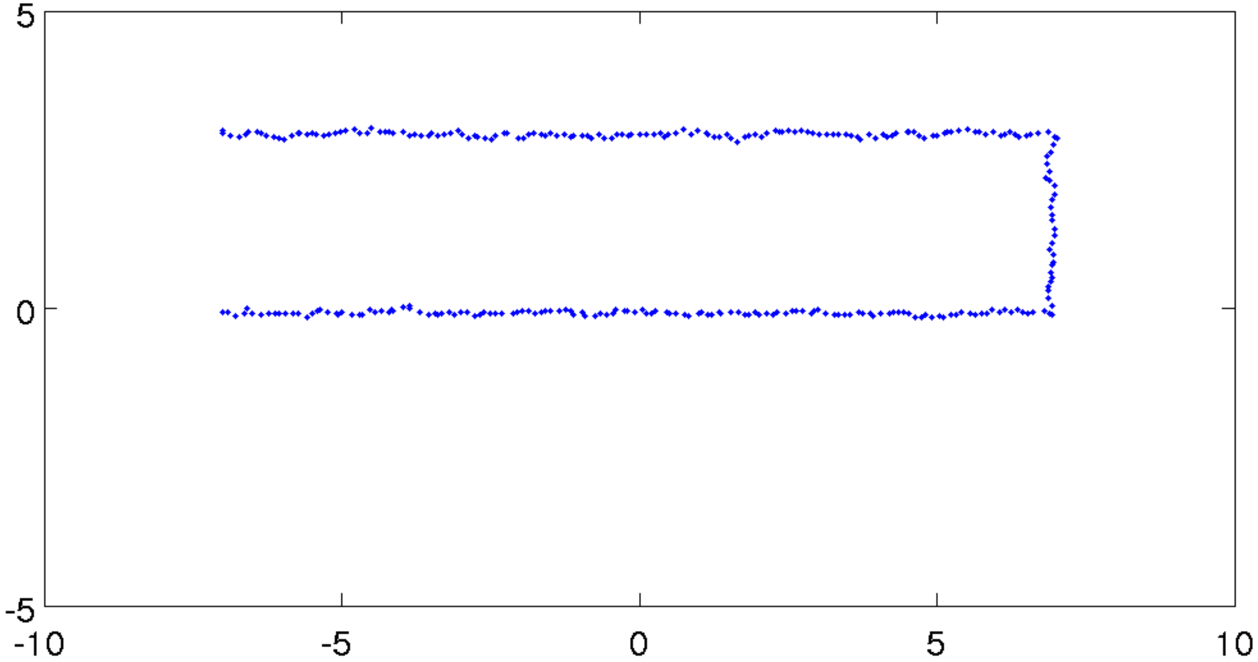
measurement noise for landmark 1 (*additive)

measurement noise for landmark 2 (*additive)

measurement noise for landmark n (*additive)

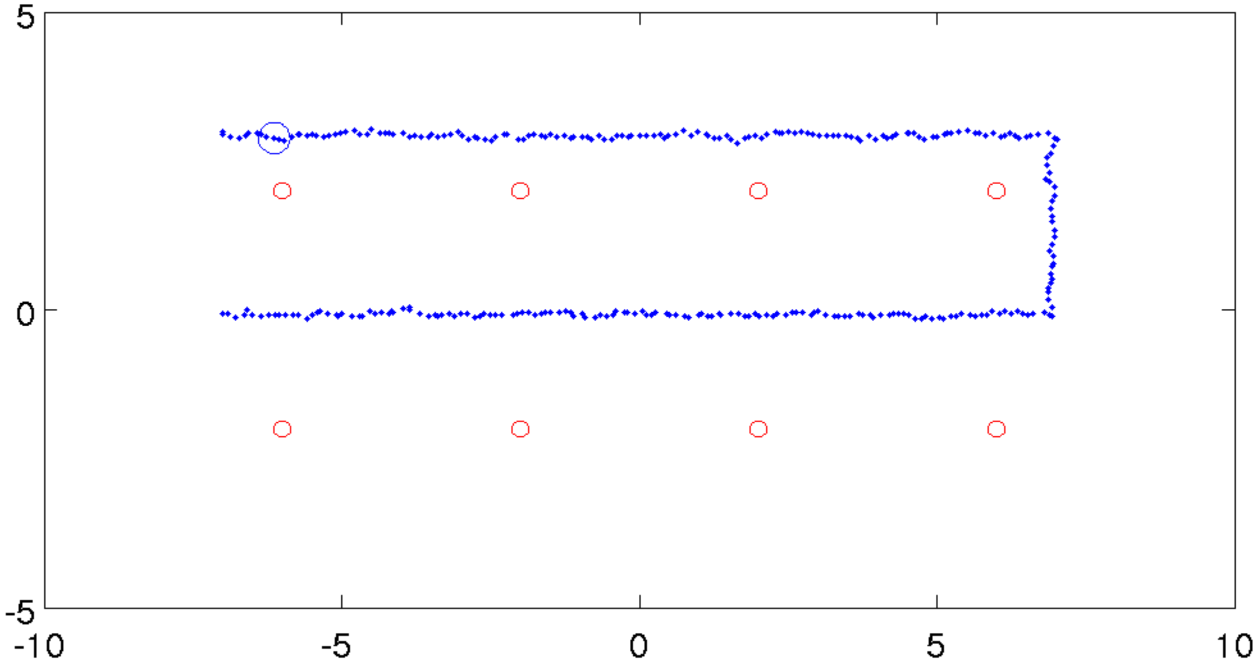
Example 1

- estimated path



Example 1

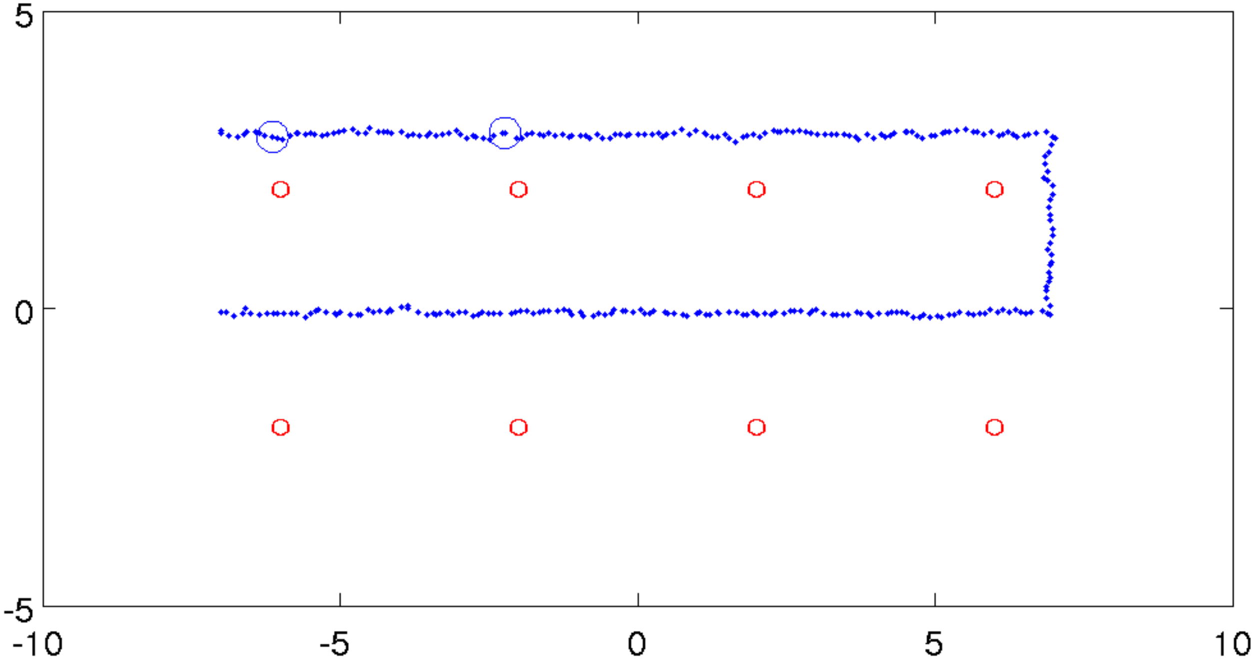
- robot position covariance
- landmark position covariance



- robot position covariance is small near the start point because the robot has not travelled far enough for the control input noise to accumulate

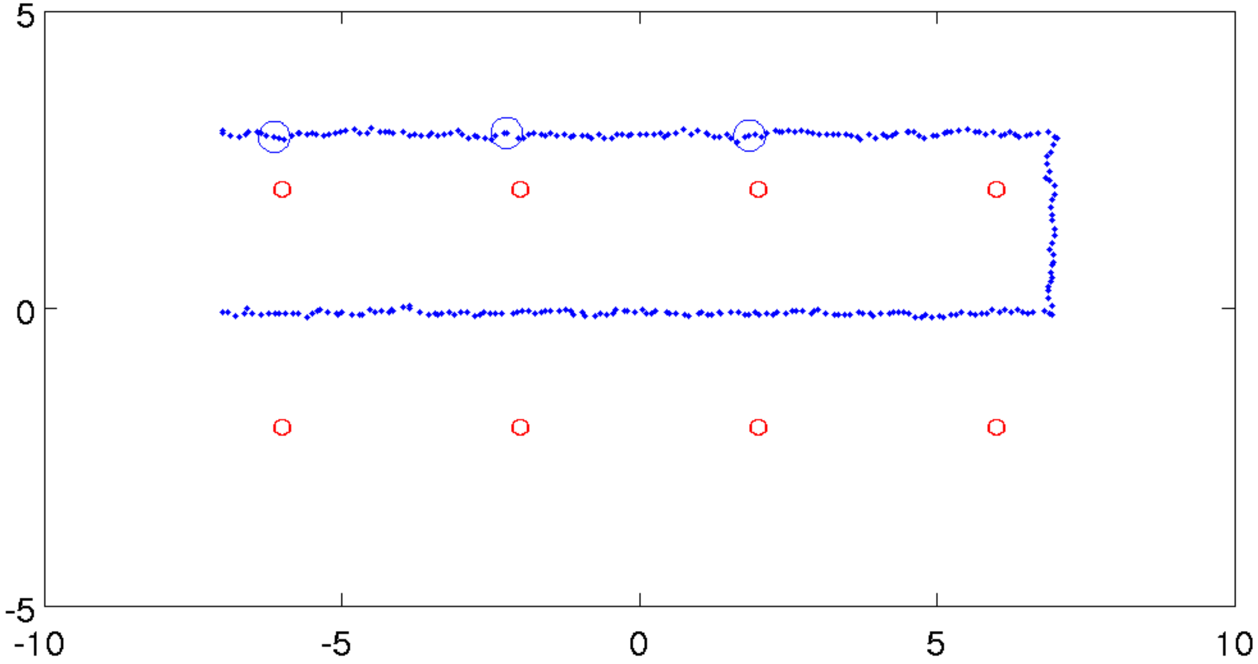
Example 1

- robot position covariance
- landmark position covariance



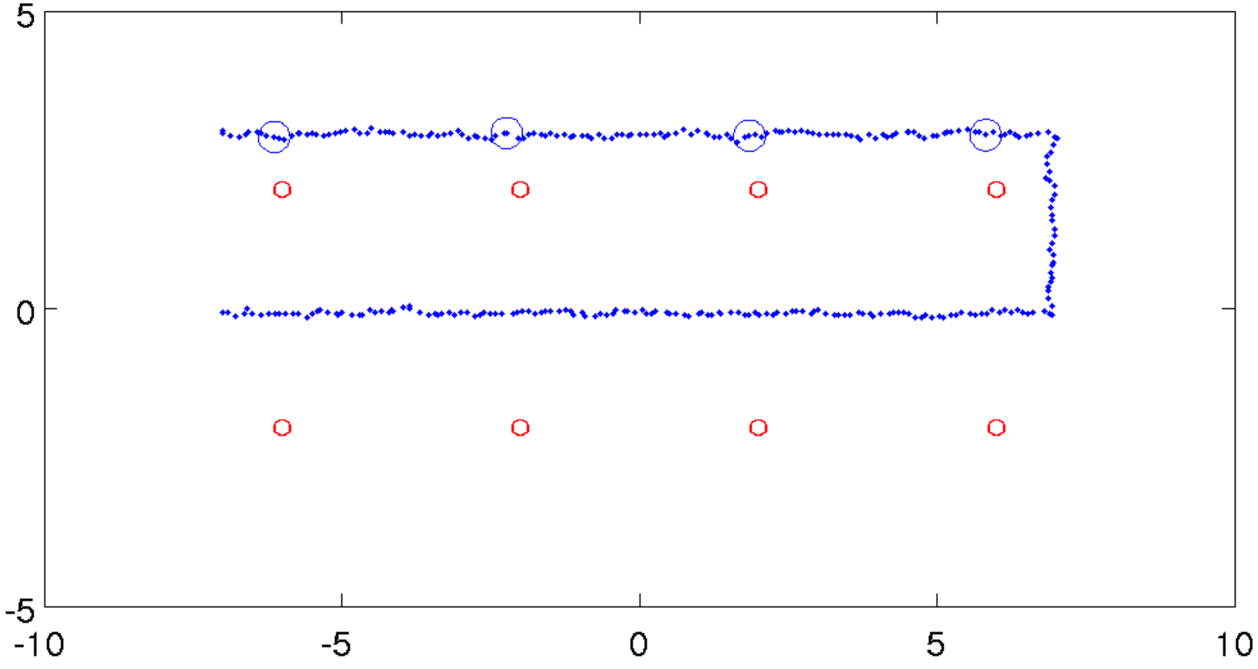
Example 1

- robot position covariance
- landmark position covariance



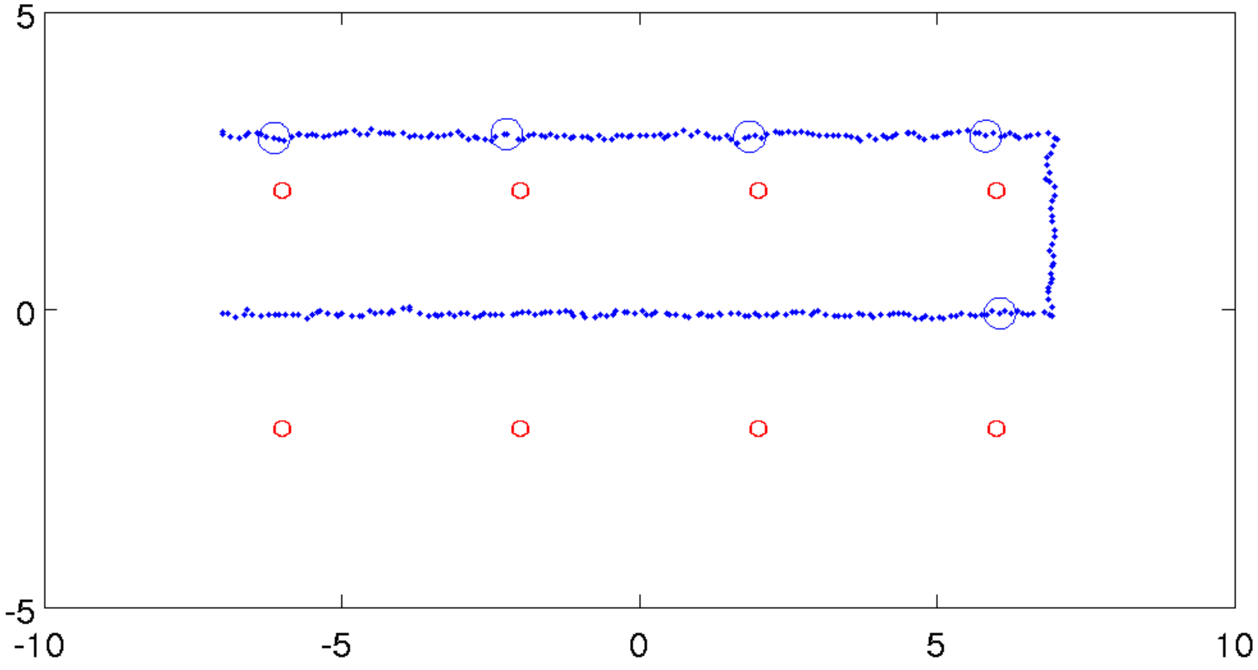
Example 1

- robot position covariance
- landmark position covariance



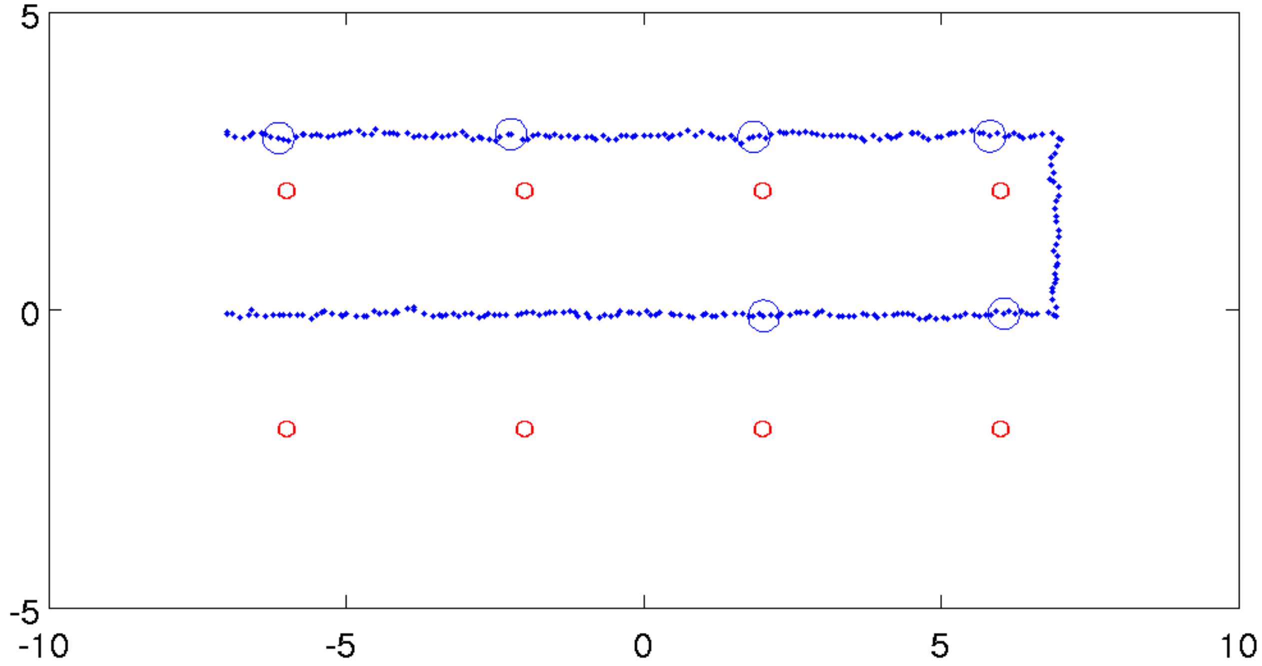
Example 1

- robot position covariance
- landmark position covariance



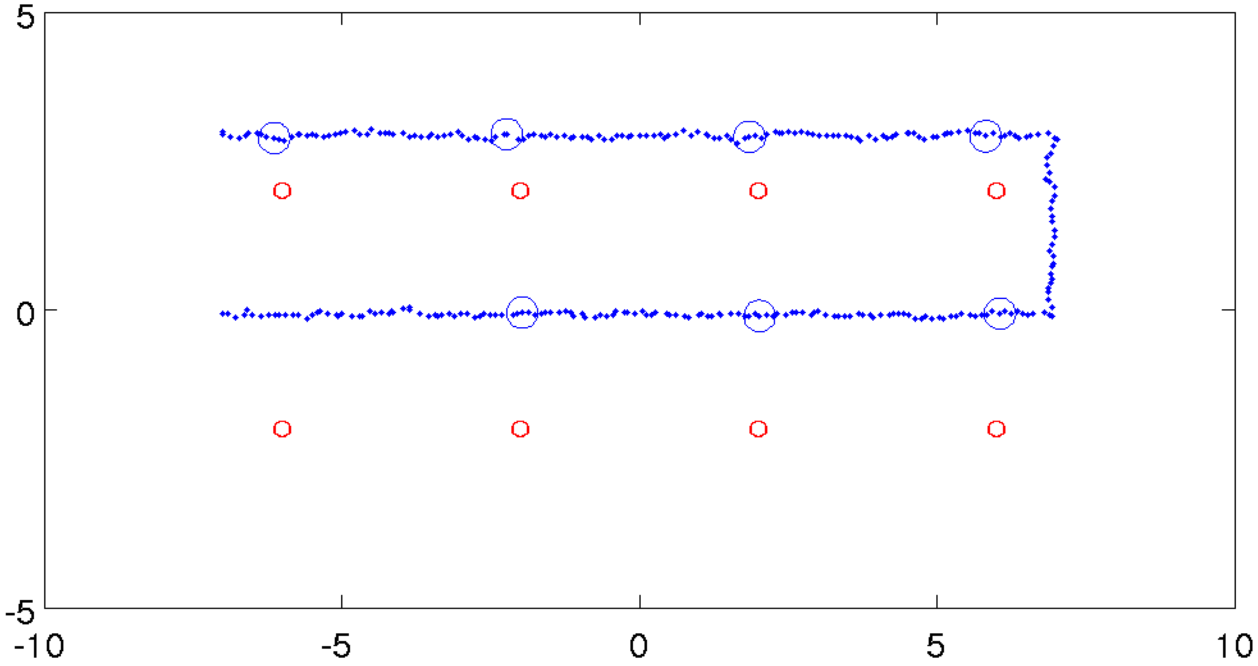
Example 1

- robot position covariance
- landmark position covariance



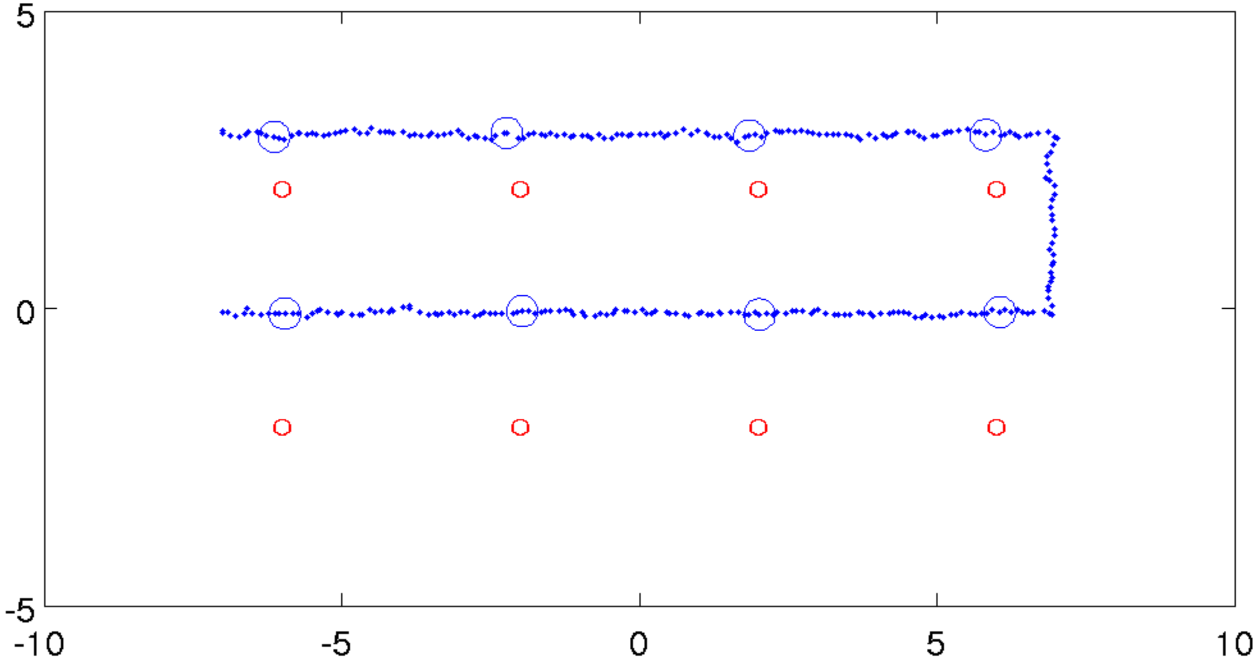
Example 1

- robot position covariance
- landmark position covariance



Example 1

- robot position covariance
- landmark position covariance

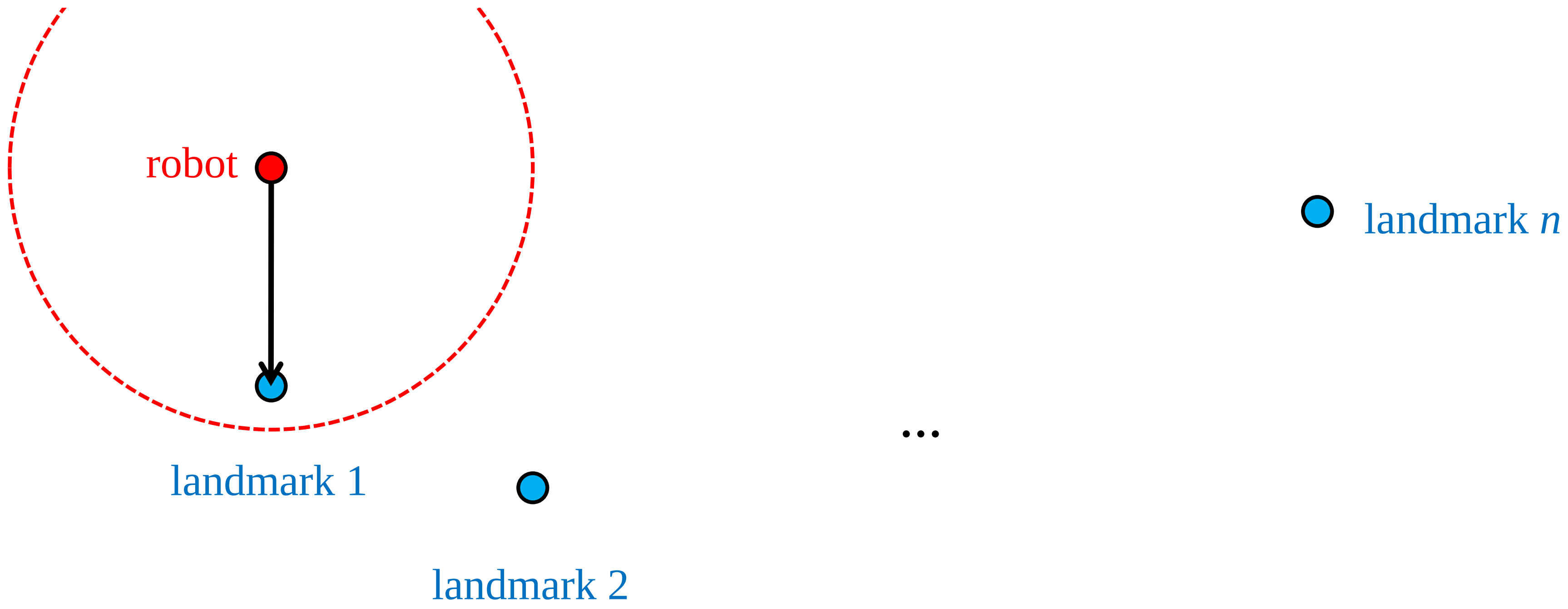


Example 2

- the assumption that the robot can measure all of the landmarks all of the time is unrealistic for many kinds of sensors
  - what do we need to change if the robot can only sense a subset of the landmarks at any given time?
    - not the plant model

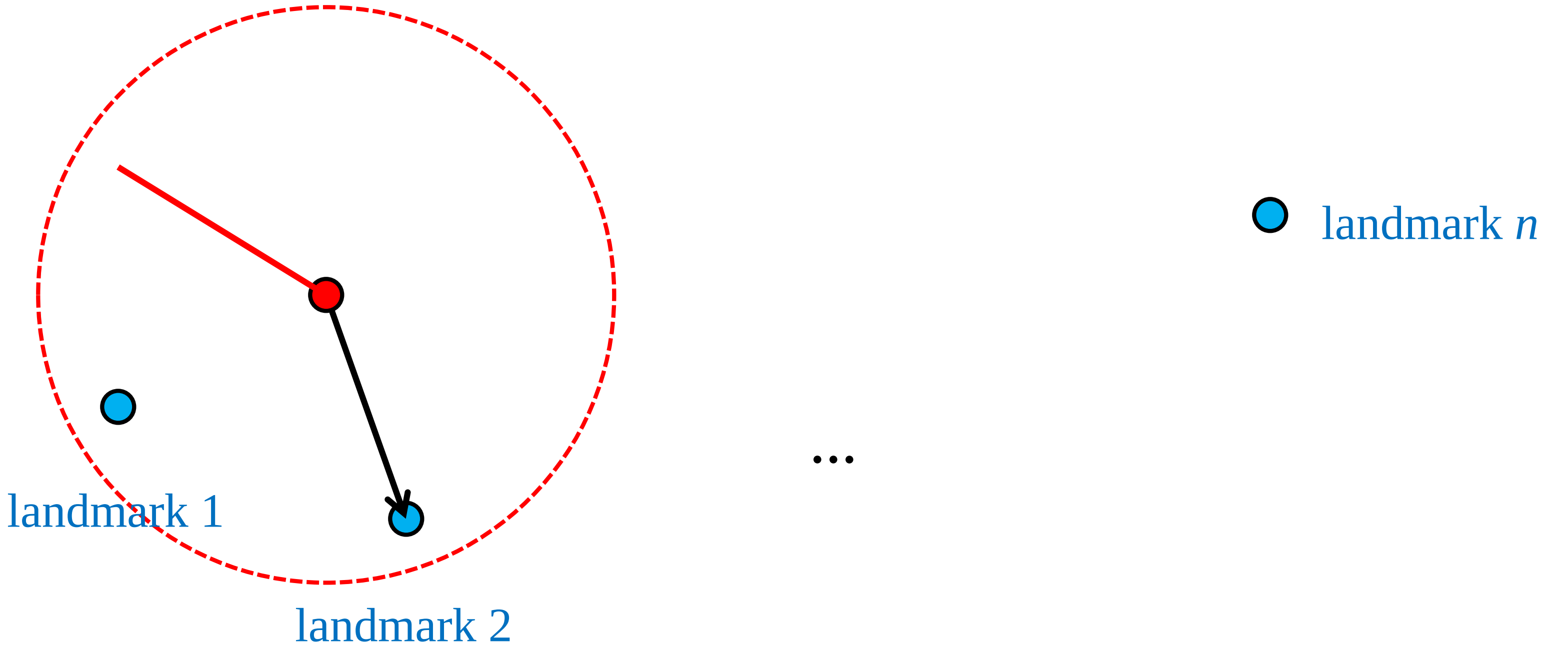
Measurement Model

- the robot can measure all of the landmarks within a fixed radius around the robot
- the robot measures the relative offset from its position to each landmark



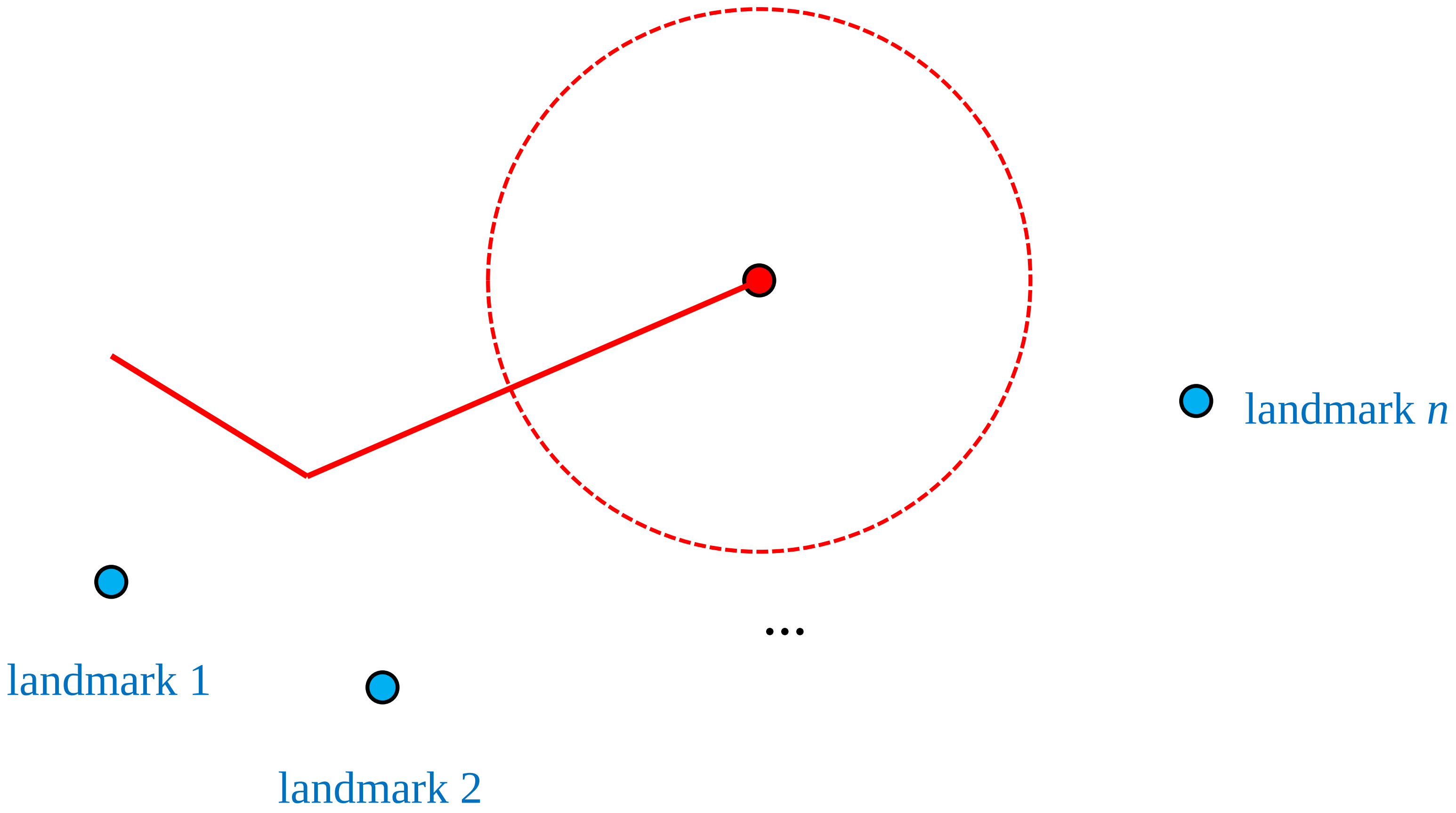
Measurement Model

- the robot might see multiple landmarks along sections of its path



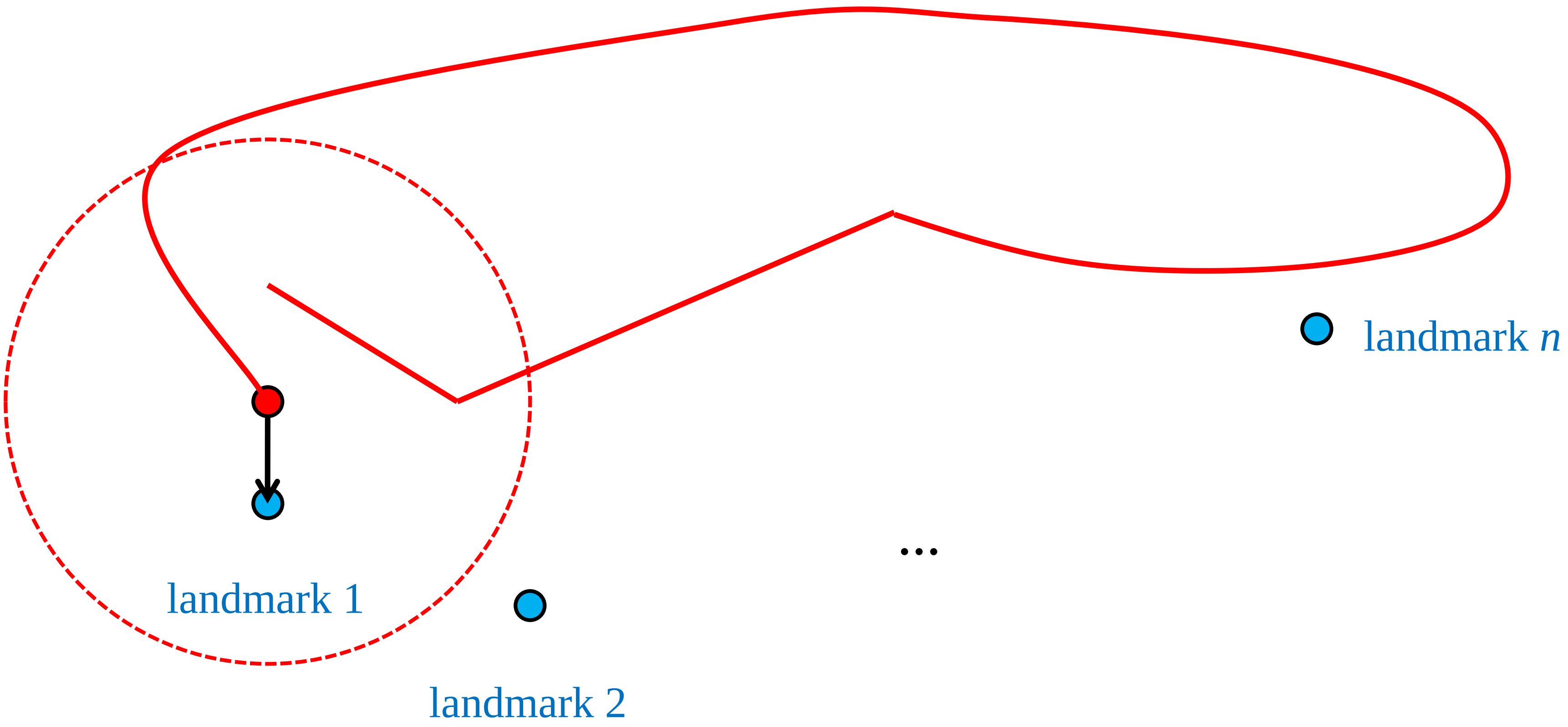
Measurement Model

- the robot may not be able to measure any landmarks along sections of its path



Measurement Model

- the robot might revisit previously seen landmarks



Measurement Model

- assume that the actual sensor measurement for a landmark outside of the sensor range is $[0 \;\; 0]^{\mathrm{T}}$
  - then we need to change the measurement model so that we can "zero out" measurements to landmarks that are outside of the sensor range
    - this is easily done by manipulating C

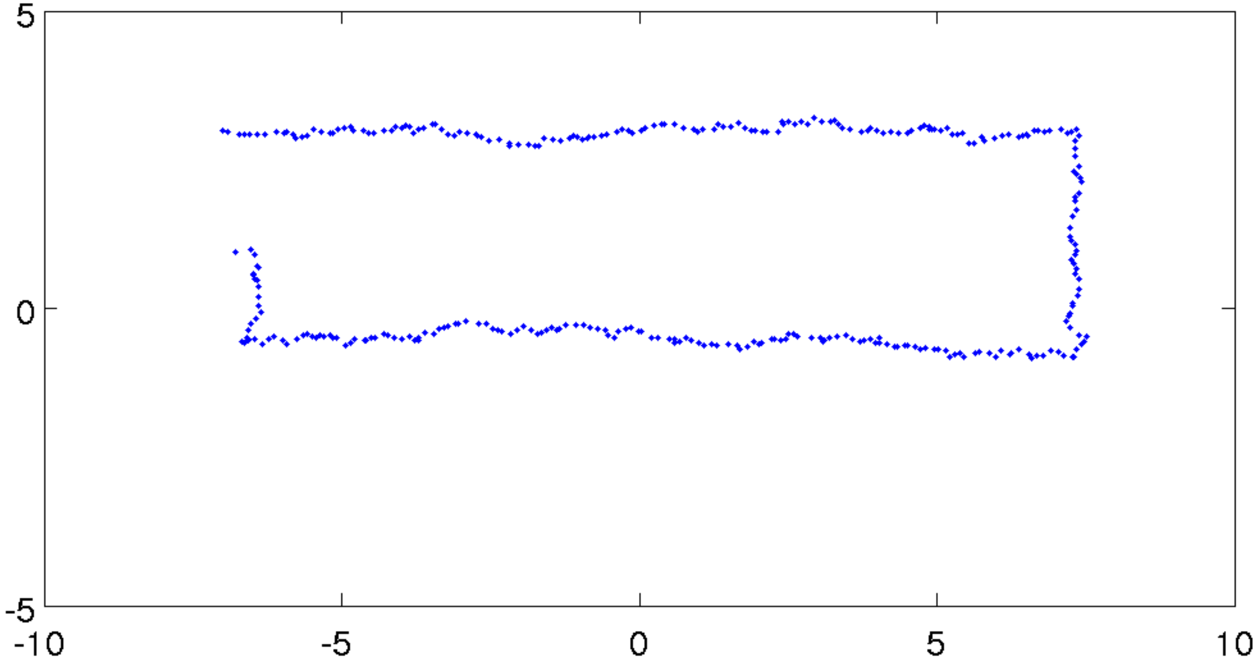
Measurement Model

- suppose that landmark 2 is outside of sensor range
  - zero out the rows corresponding to the measurement for landmark 2
    - C is now a function, C_t, of time

$$z_t = \underbrace{\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ & & & & & & \ldots & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & \ldots & 0 & 1 \end{bmatrix}}_{C_t} \underbrace{\begin{bmatrix} x \\ y \\ p_{1,x} \\ p_{1,y} \\ p_{2,x} \\ p_{2,y} \\ \vdots \\ p_{n,x} \\ p_{n,x} \end{bmatrix}}_{x_t}$$

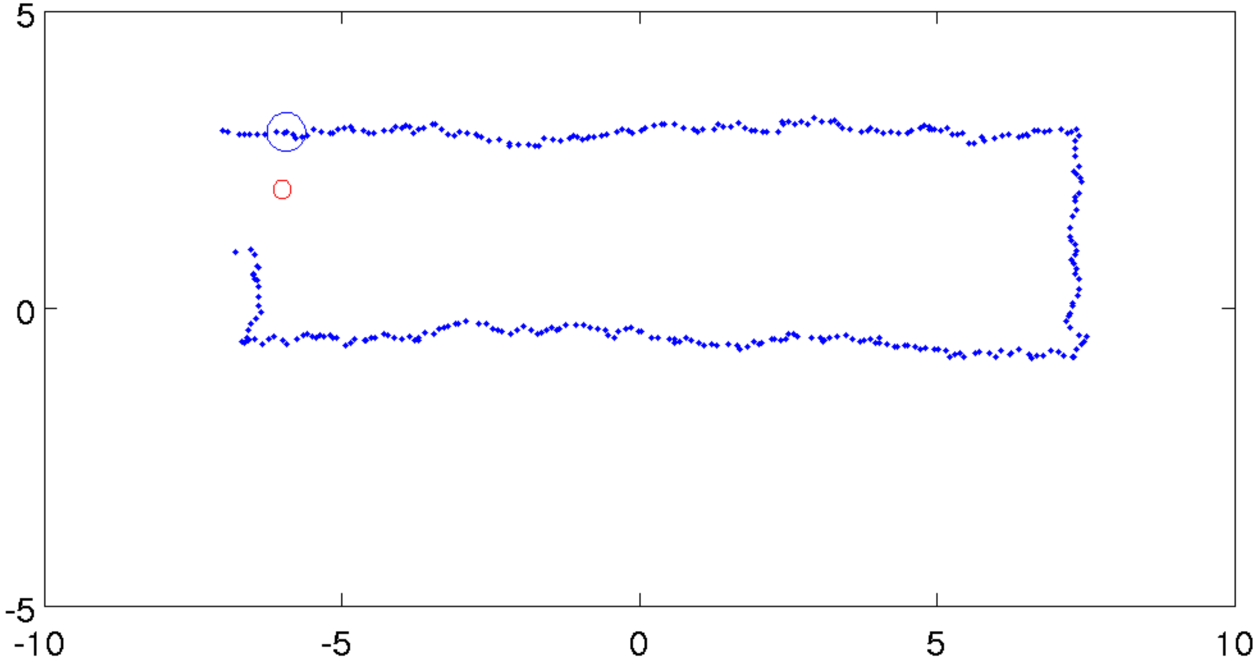
Example 2

- estimated path



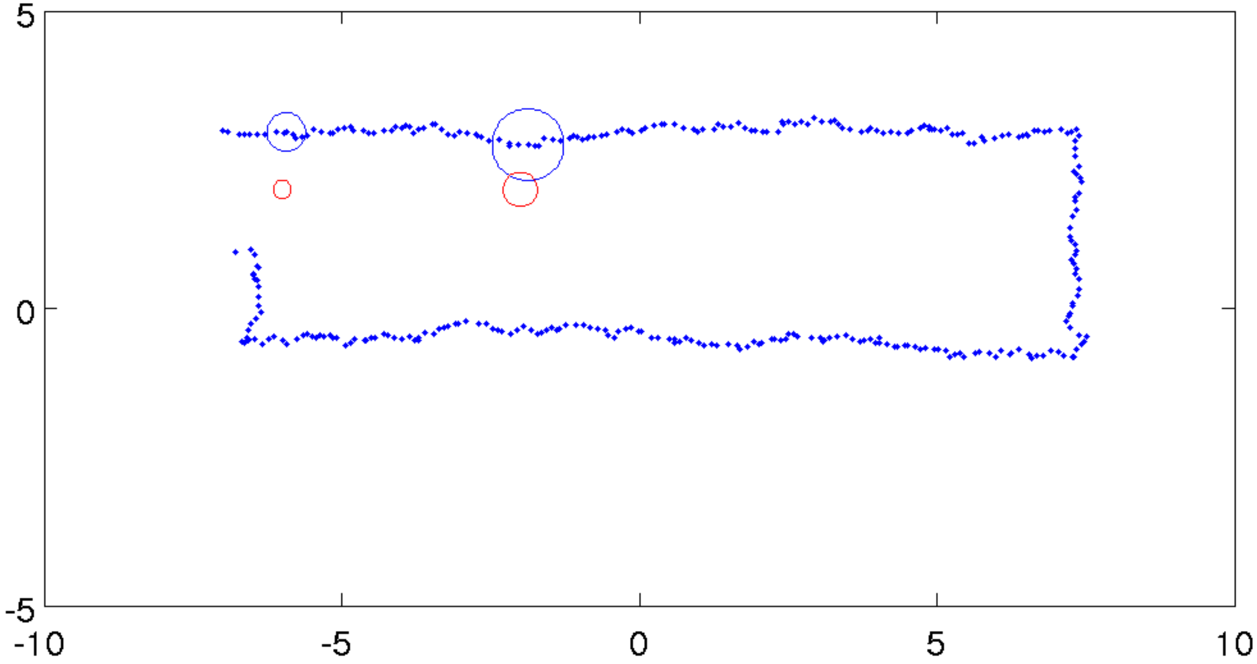
Example 2

- robot position covariance
- landmark position covariance



Example 2

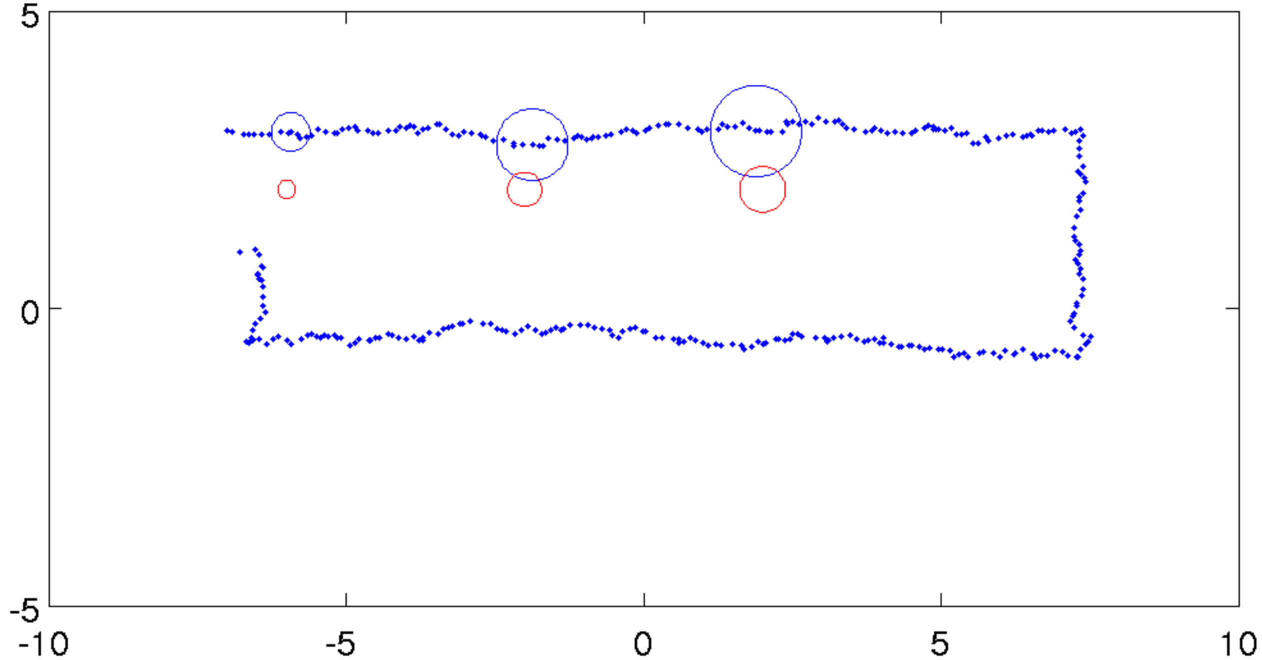
- robot position covariance
- landmark position covariance



- robot position covariance grows as control input noise accumulates (and cannot be completely corrected by observing one landmark at a time)

Example 2

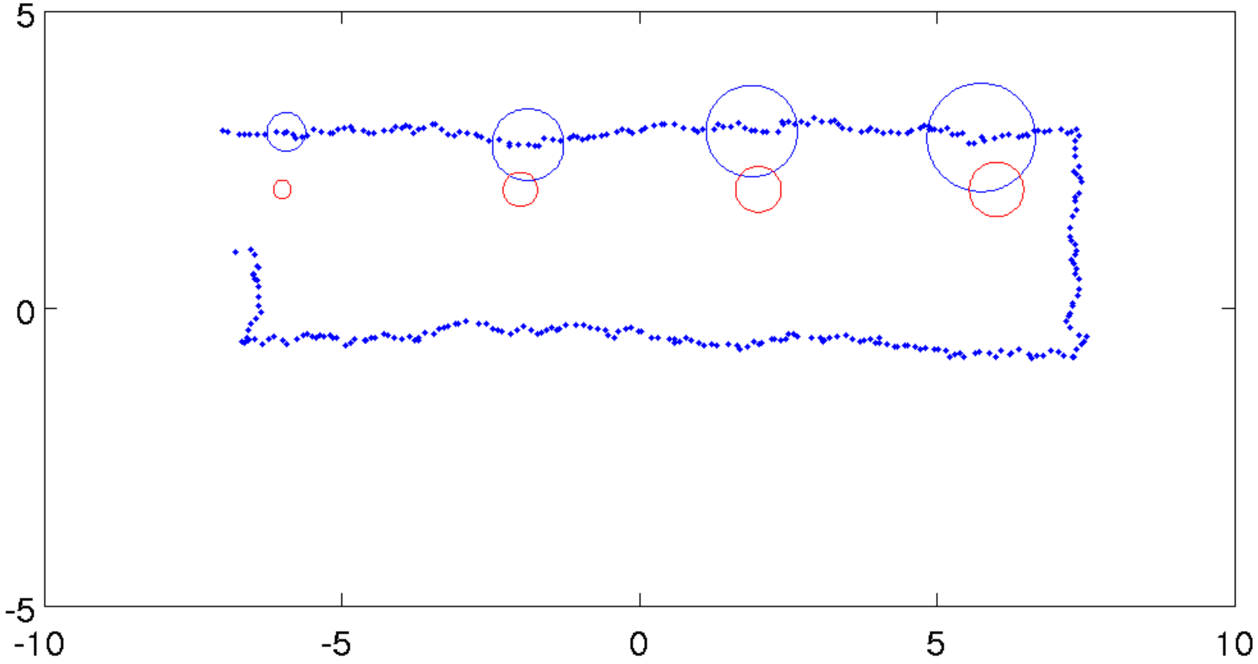
- robot position covariance
- landmark position covariance



- each subsequent landmark position covariance grows because the landmark position estimate is correlated with the robot's position

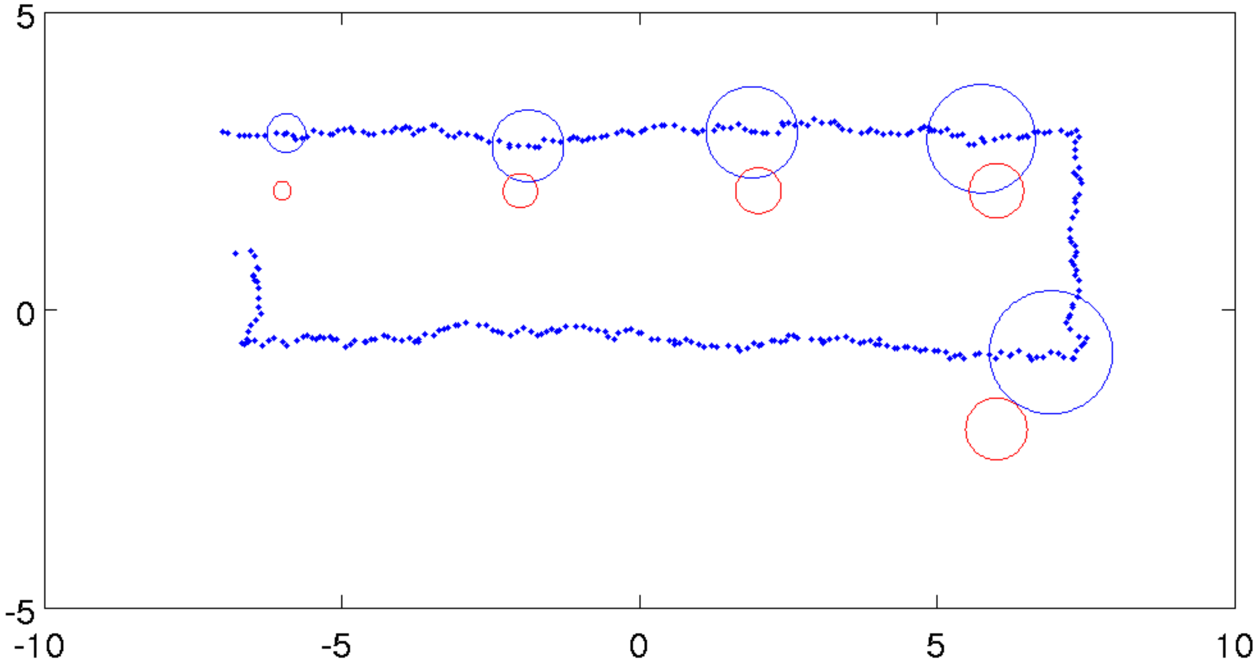
Example 2

- robot position covariance
- landmark position covariance



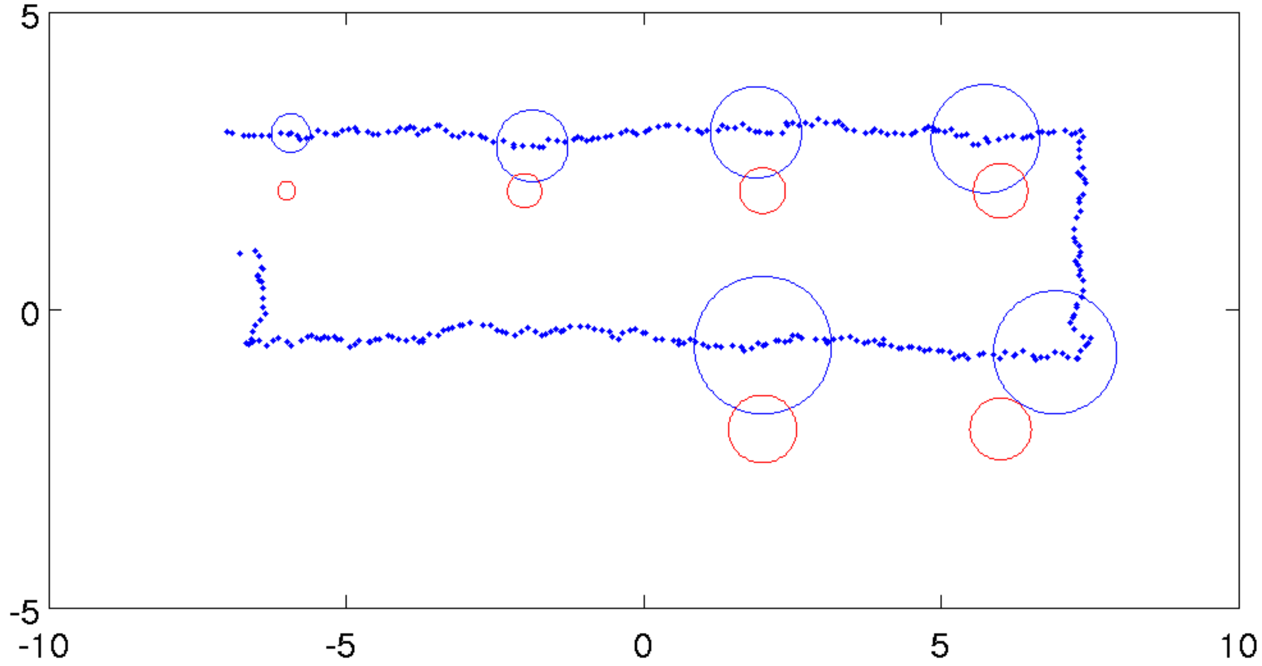
Example 2

- robot position covariance
- landmark position covariance



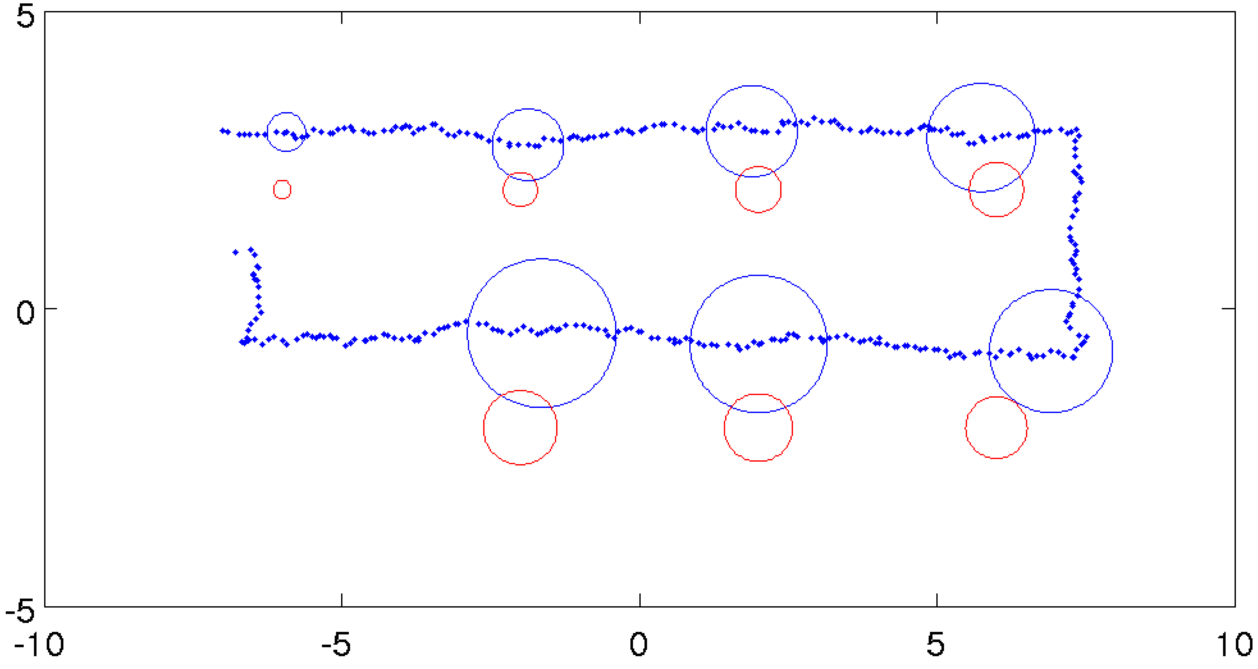
Example 2

- robot position covariance
- landmark position covariance



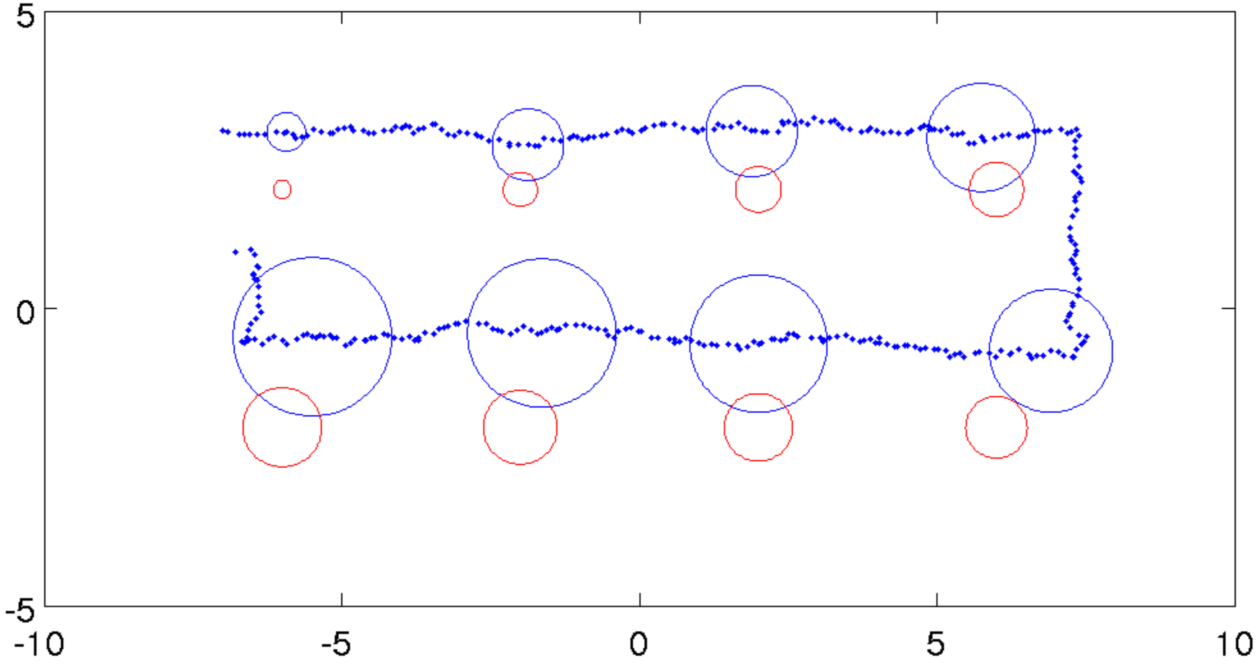
Example 2

- robot position covariance
- landmark position covariance



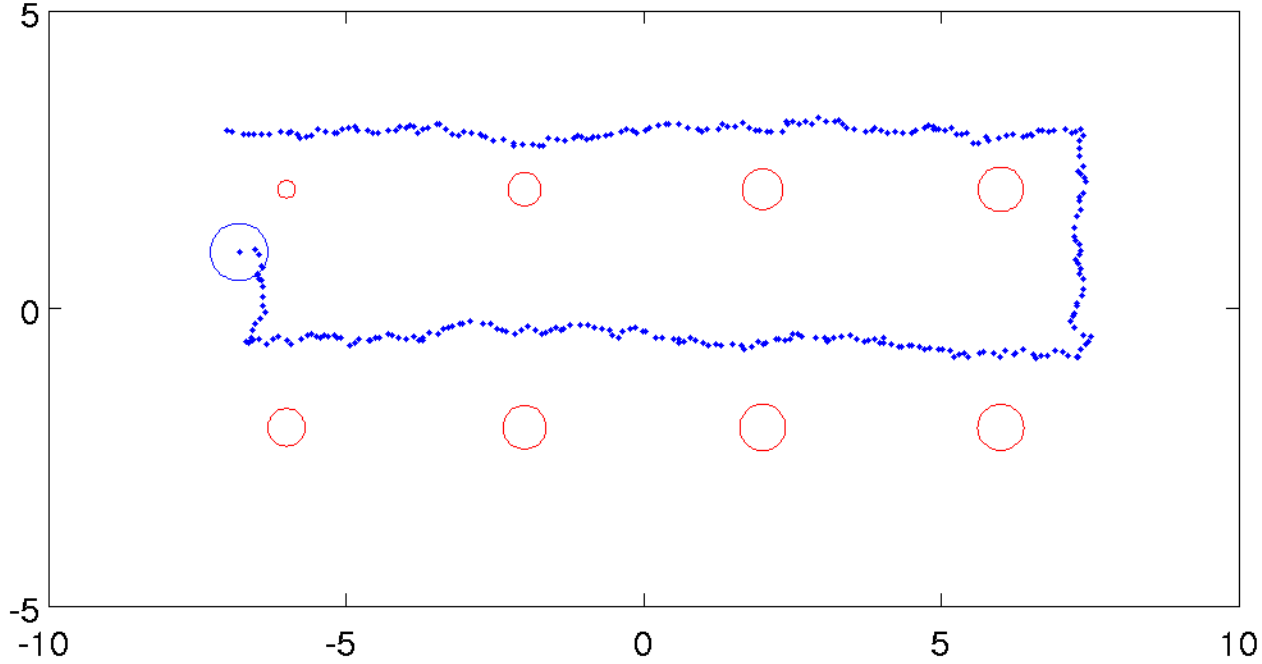
Example 2

- robot position covariance
- landmark position covariance



Example 2

- robot position covariance
- landmark position covariance



- when the robot sees the first landmark for a second time all of the covariances decrease!

EKF Slam Example 3

(a) (b) (c) (d)

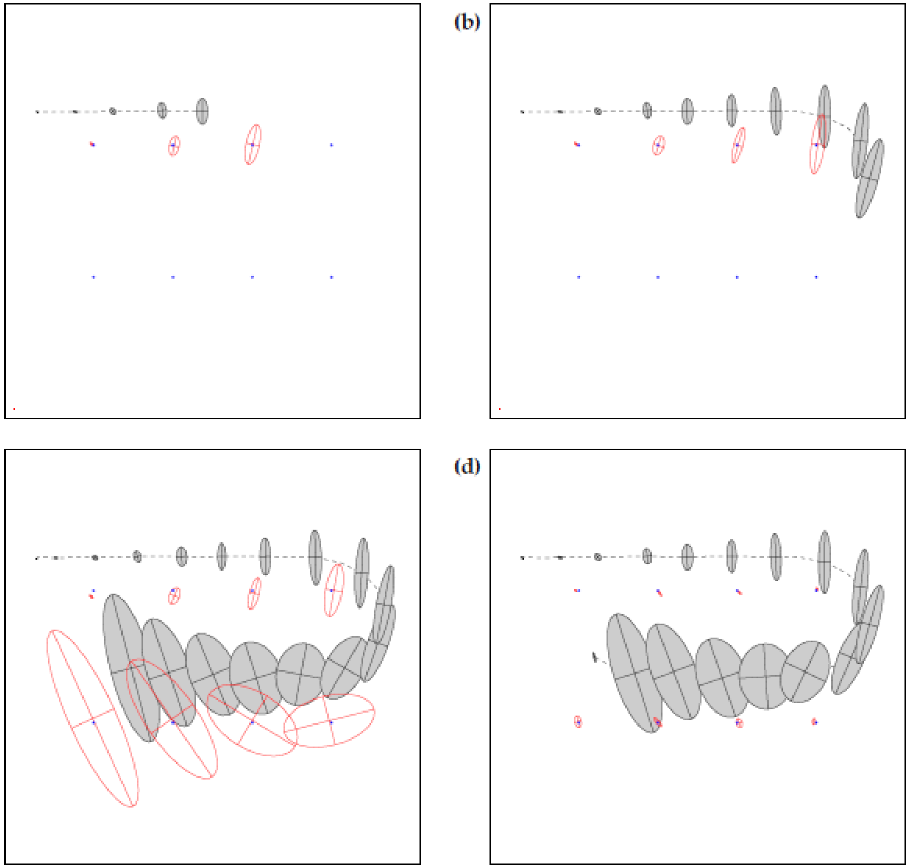


Figure 10.3 EKF applied to the online SLAM problem. The robot's path is a dotted line, and its estimates of its own position are shaded ellipses. Eight distinguishable landmarks of unknown location are shown as small dots, and their location estimates are shown as white ellipses. In (a)–(c) the robot's positional uncertainty is increasing, as is its uncertainty about the landmarks it encounters. In (d) the robot senses the first landmark again, and the uncertainty of *all* landmarks decreases, as does the uncertainty of its current pose. Image courtesy of Michael Montemerlo, Stanford University.

Loop Closure

- the phenomenon illustrated on the previous slide is called "closing the loop"
  - occurs when a robot moving through unknown terrain encounters a landmark not seen for a "long" time
- KF and EKF SLAM intrinsically perform loop closure (with high computational cost)
  - online demo http://rogerstuckey.com/simulation/javascript/slam-html5/
  - not every SLAM algorithm is capable of loop closure without an explicit loop closure process

Loop Closure in KF SLAM

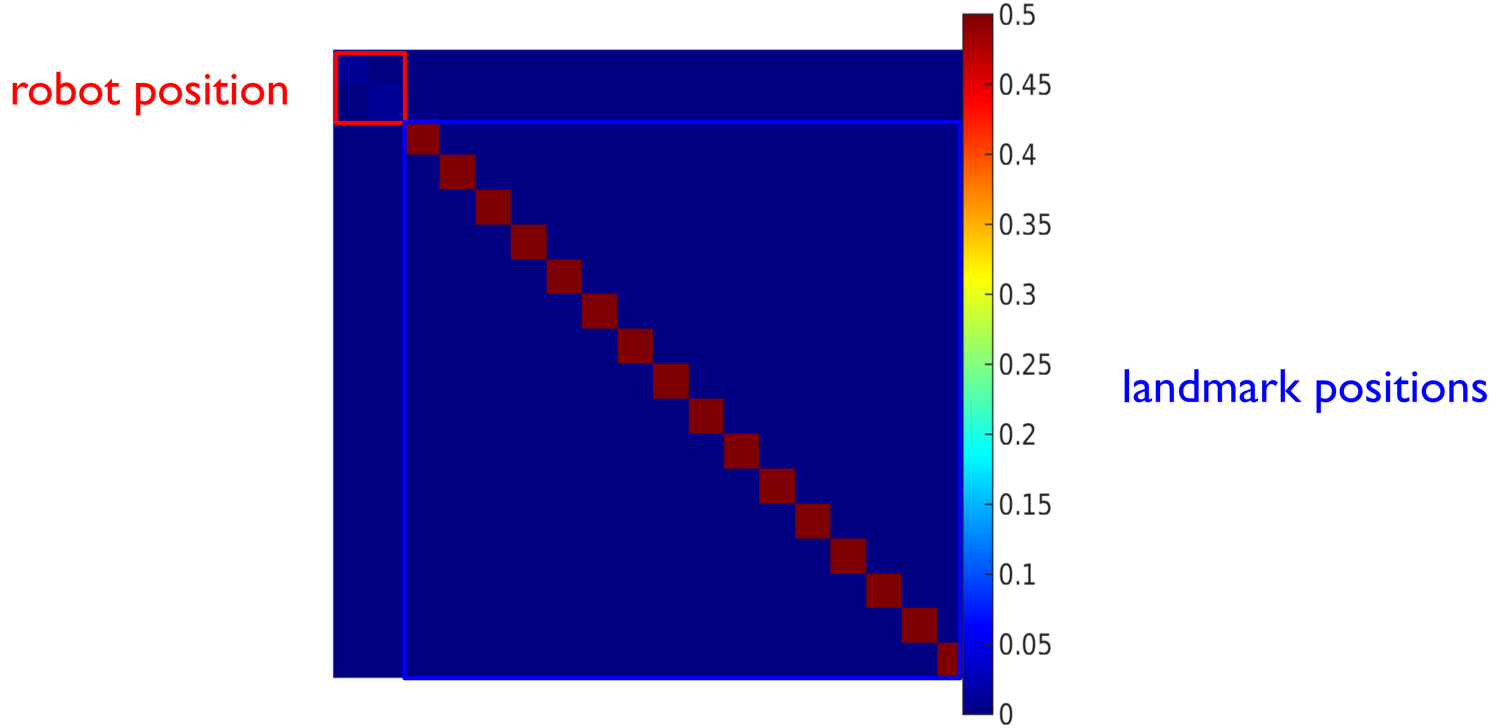
- it is not immediately obvious how KF SLAM performs loop closure until you look at the estimated state covariance matrix

$$x_t = \begin{bmatrix} X_t \\ Y_t \\ p_{1,x} \\ p_{1,y} \\ p_{2,x} \\ p_{2,y} \\ \vdots \\ p_{n,x} \\ p_{n,x} \end{bmatrix}$$

$$\operatorname{cov}(x_t) = \begin{bmatrix}
\operatorname{cov}(X_t) & & & & & & \\
\operatorname{cov}(X_t Y_t) & \operatorname{cov}(Y_t Y_t) & & & & & \\
\operatorname{cov}(X_t p_{1,x}) & \operatorname{cov}(Y_t p_{1,x}) & \operatorname{cov}(p_{1,x} p_{1,x}) & & & & \\
\operatorname{cov}(X_t p_{1,y}) & \operatorname{cov}(Y_t p_{1,y}) & \operatorname{cov}(p_{1,x} p_{1,y}) & \operatorname{cov}(p_{1,y} p_{1,y}) & & & \\
\operatorname{cov}(X_t p_{2,x}) & \operatorname{cov}(Y_t p_{2,x}) & \operatorname{cov}(p_{1,x} p_{2,x}) & \operatorname{cov}(p_{1,y} p_{2,x}) & \ddots & & \\
\operatorname{cov}(X_t p_{2,y}) & \operatorname{cov}(Y_t p_{2,y}) & \operatorname{cov}(p_{1,x} p_{2,y}) & \operatorname{cov}(p_{1,y} p_{2,y}) & & \ddots & \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \\
\operatorname{cov}(X_t p_{n,x}) & \operatorname{cov}(X_t p_{n,x}) & \operatorname{cov}(p_{1,x} p_{n,x}) & \operatorname{cov}(p_{1,y} p_{n,x}) & \cdots & \cdots & \\
\operatorname{cov}(X_t p_{n,x}) & \operatorname{cov}(Y_t p_{n,y}) & \operatorname{cov}(p_{1,x} p_{n,y}) & \operatorname{cov}(p_{1,y} p_{n,y}) & \cdots & \cdots & \operatorname{cov}(p_{n,y} p_{n,y})
\end{bmatrix}$$

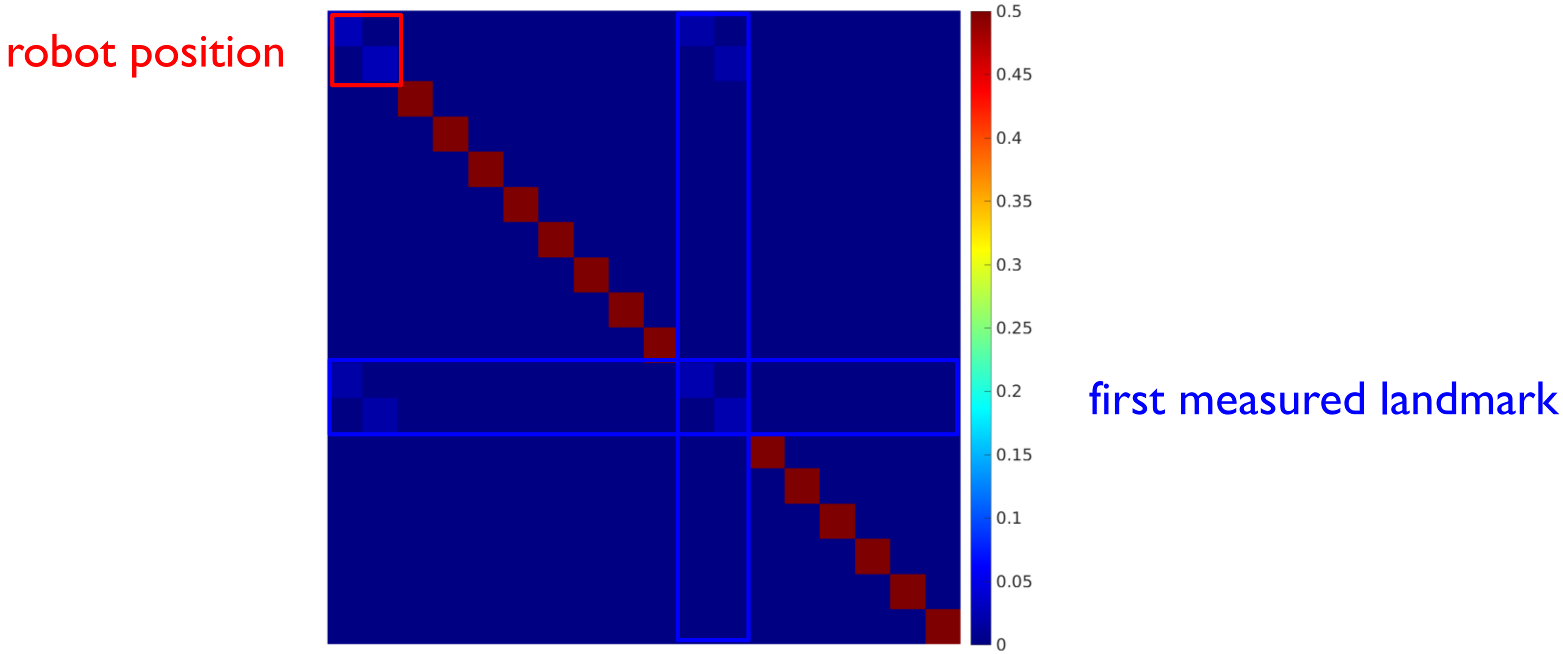
Loop Closure in KF SLAM

- at the start of the path the robot's position is assumed known and the landmark positions are assumed to be uknown
  - following images show state covariance matrix
  - black = 0 and white > 0.08



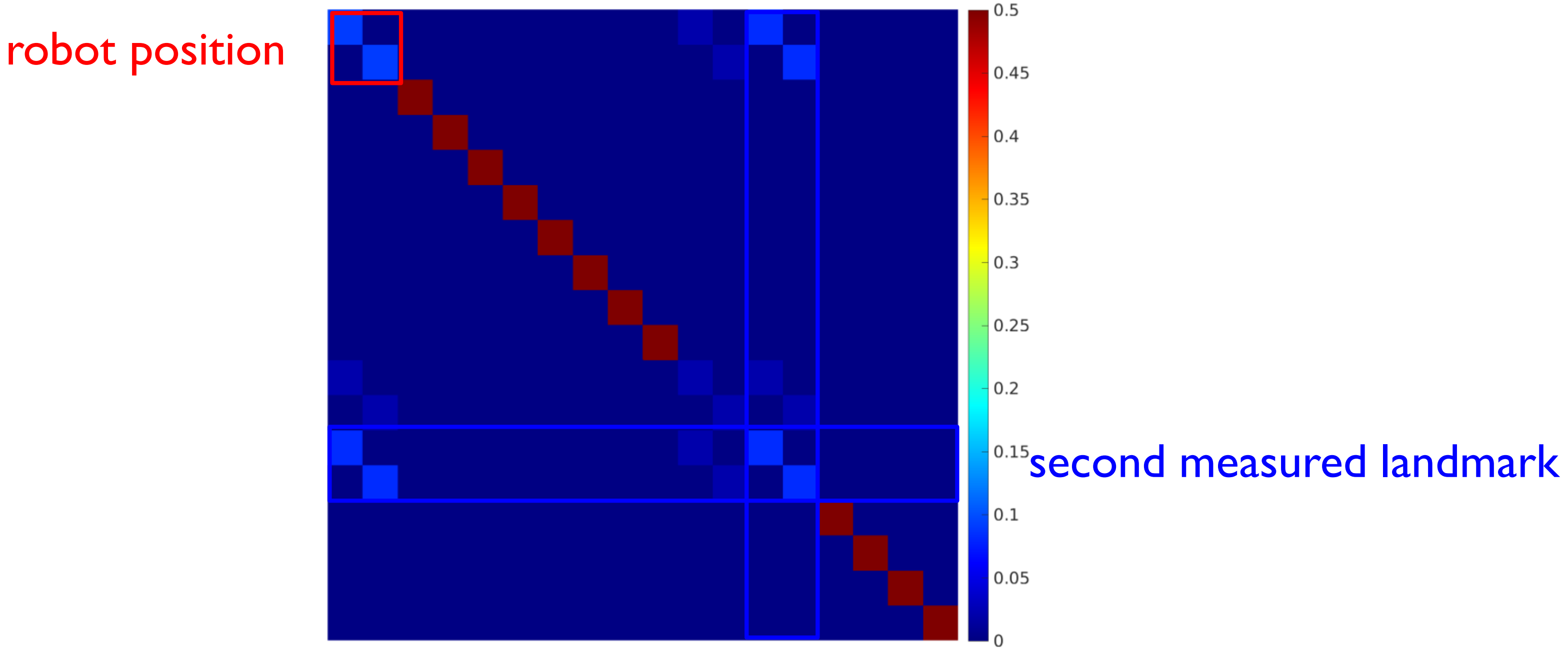
Loop Closure in KF SLAM

- the covariance of the first measured landmark is related to the covariance of the estimated position of the robot



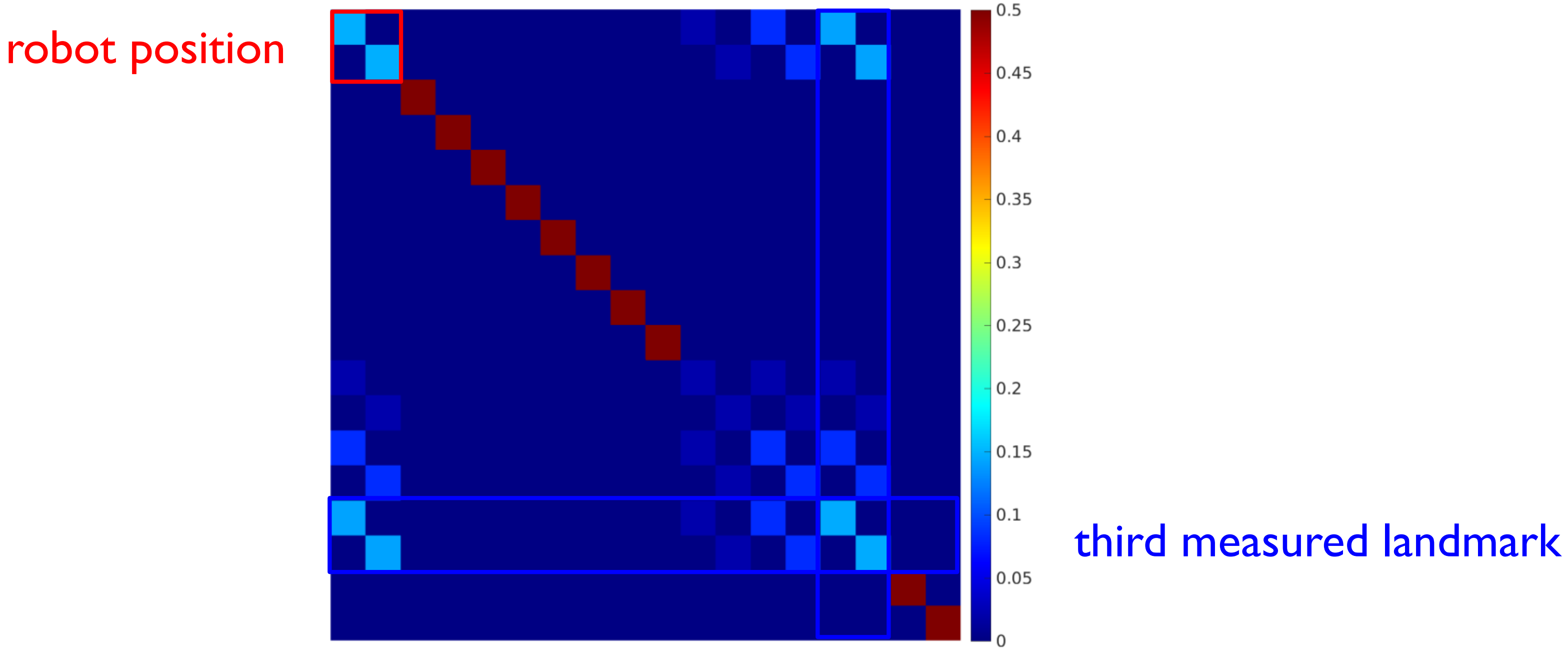
Loop Closure in KF SLAM

- the covariance of the second measured landmark is related to the covariance of the estimated position of the robot, and less so to the covariance of the first measured landmark



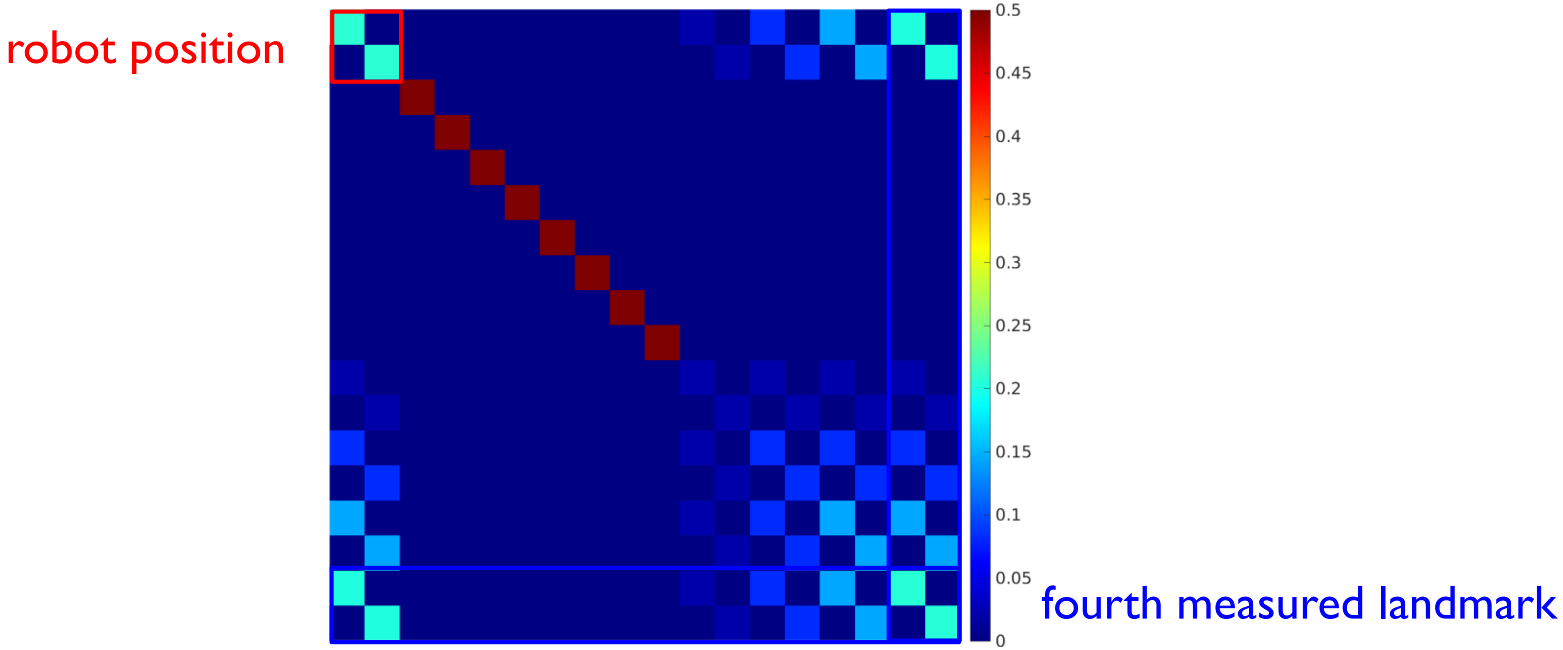
Loop Closure in KF SLAM

- the covariance of the third measured landmark is related to the covariance of the estimated position of the robot, and less so the covariances of the first and second landmarks



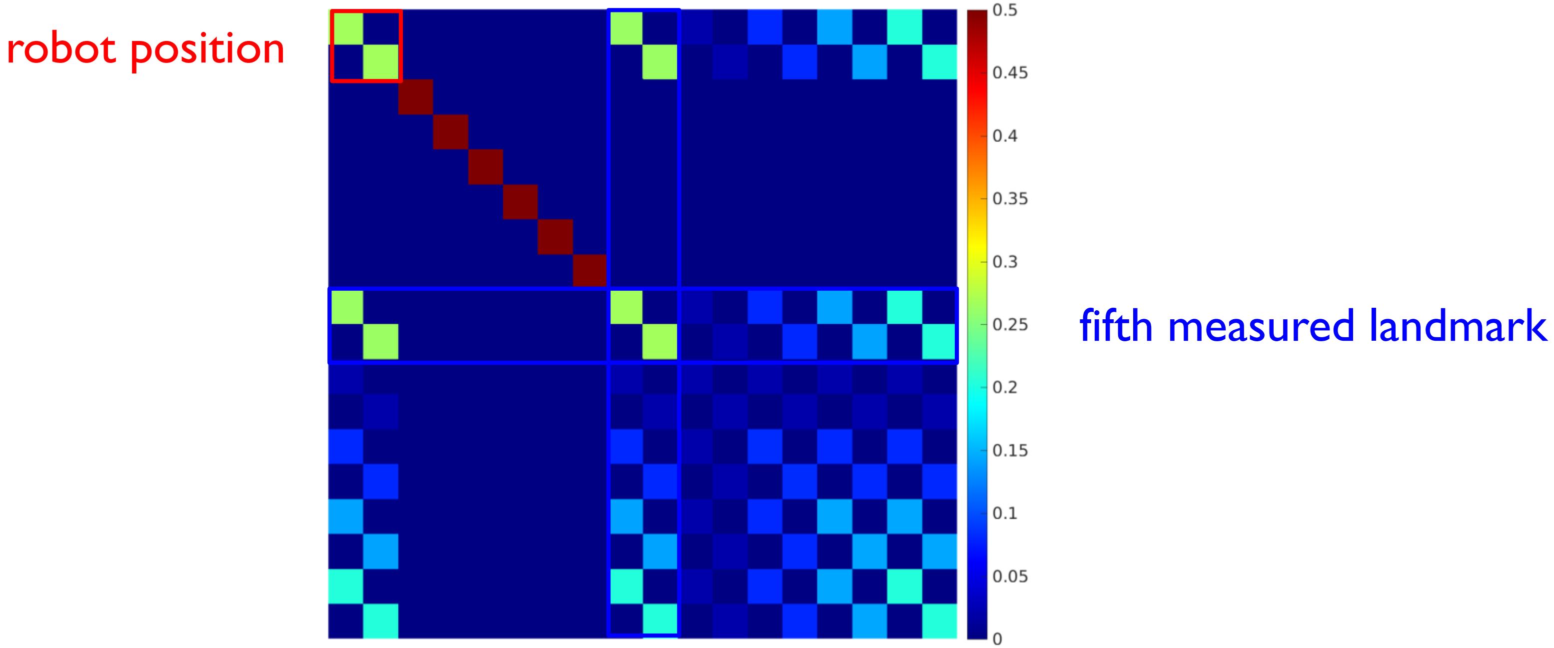
Loop Closure in KF SLAM

- the covariance of the fourth measured landmark is related to the covariance of the estimated position of the robot, and less so the covariances of the previous landmarks



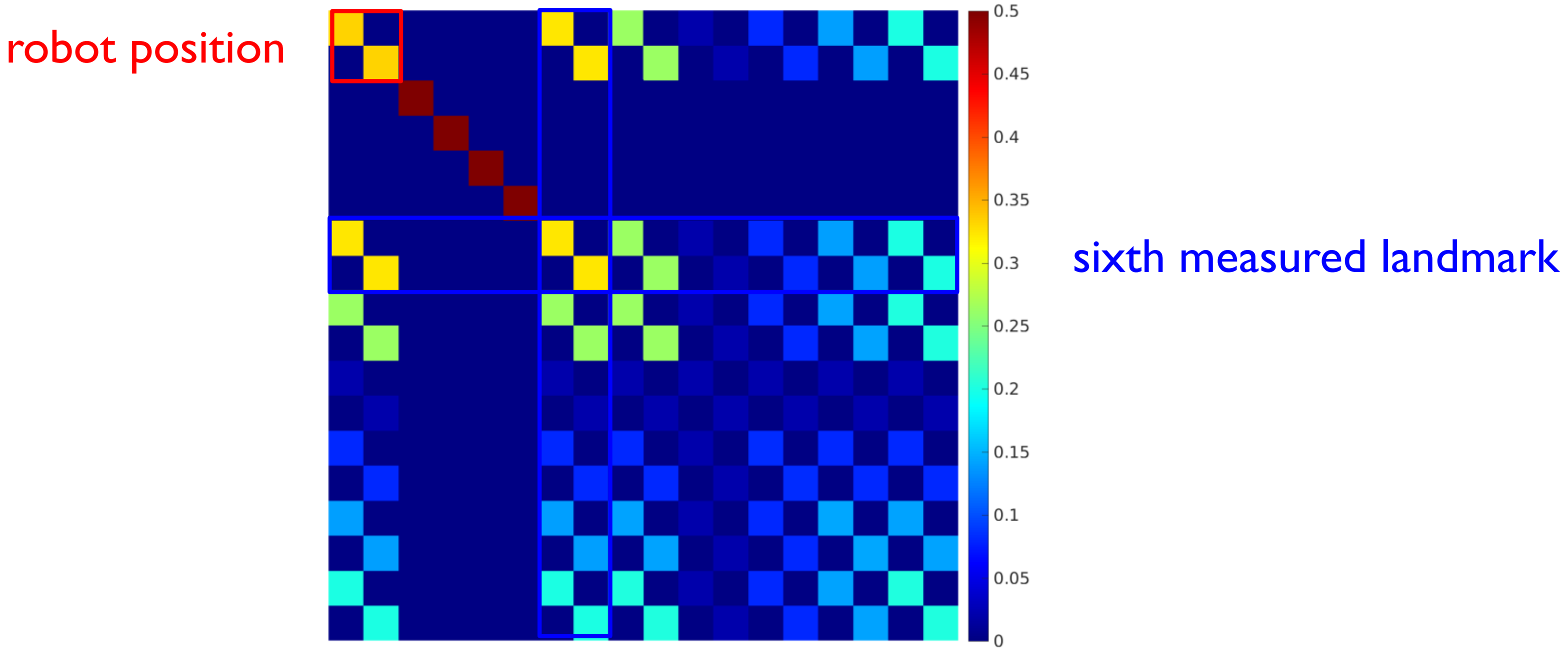
Loop Closure in KF SLAM

- the covariance of the fifth measured landmark is related to the covariance of the estimated position of the robot, and less so the covariances of the previous landmarks



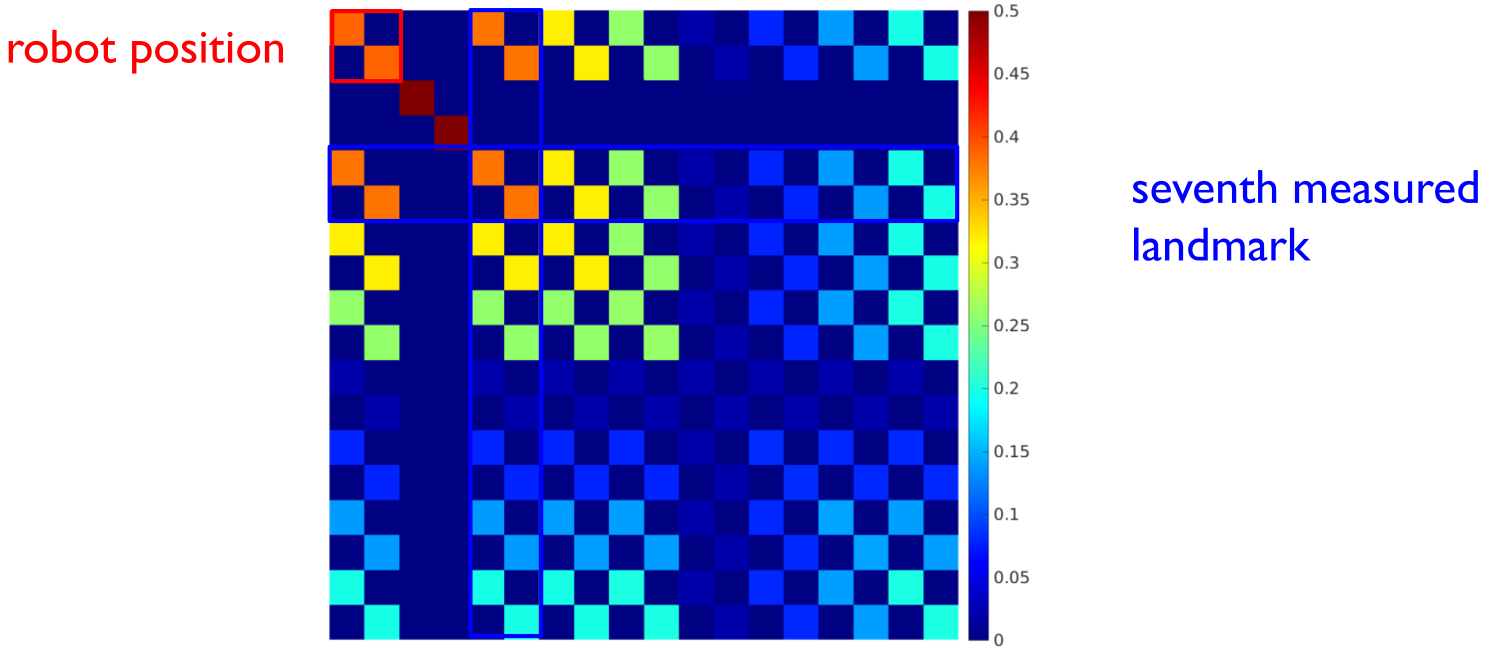
Loop Closure in KF SLAM

- the covariance of the sixth measured landmark is related to the covariance of the estimated position of the robot, and less so the covariances of the previous landmarks



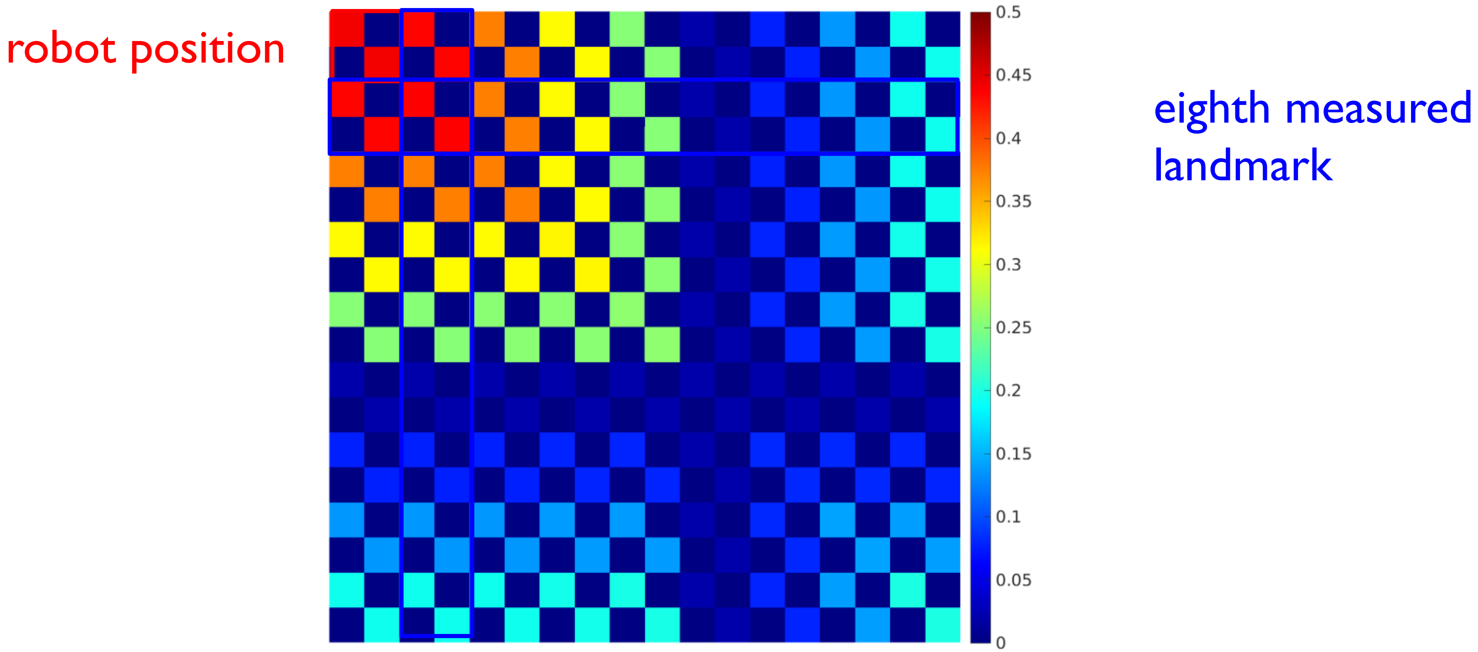
Loop Closure in KF SLAM

- the covariance of the seventh measured landmark is related to the covariance of the estimated position of the robot, and less so the covariances of the previous landmarks



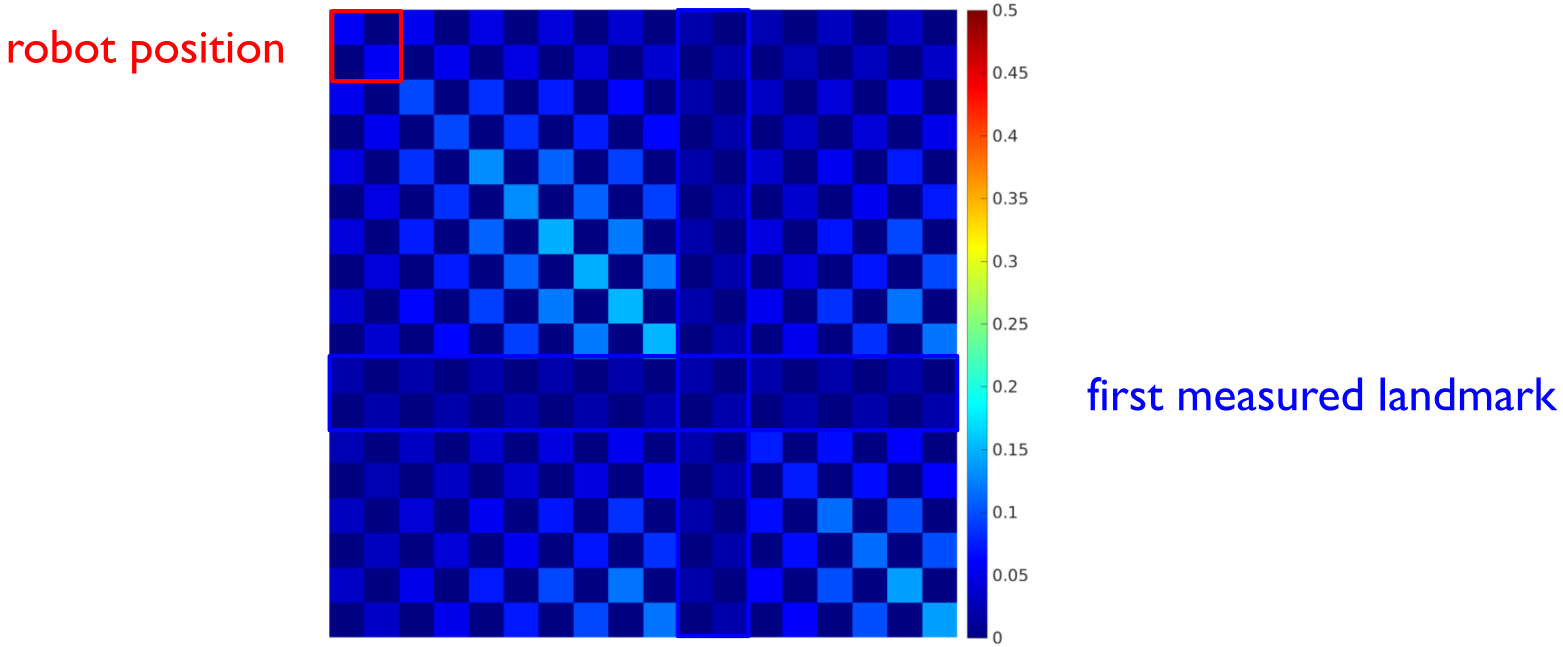
Loop Closure in KF SLAM

- the covariance of the eighth measured landmark is related to the covariance of the estimated position of the robot, and less so the covariances of the previous landmarks



Loop Closure in KF SLAM

- the covariance of the first measured landmark is related to the covariance of the estimated position of the robot and all of the other landmarks



Loop Closure in KF SLAM

- loop closure in KF SLAM comes with a high computational and storage cost
  - state covariance matrix size is $O(N^2)$ for N landmarks
  - direct implementation of KF equations require matrix multiplication and inversion
    - theoretical complexity $O(N^{2.376})$, $O(N^3)$ in practice
    - if the robot only observes a small number of landmarks then it is possible to perform an update in $O(N^2)$

KF SLAM in Practice

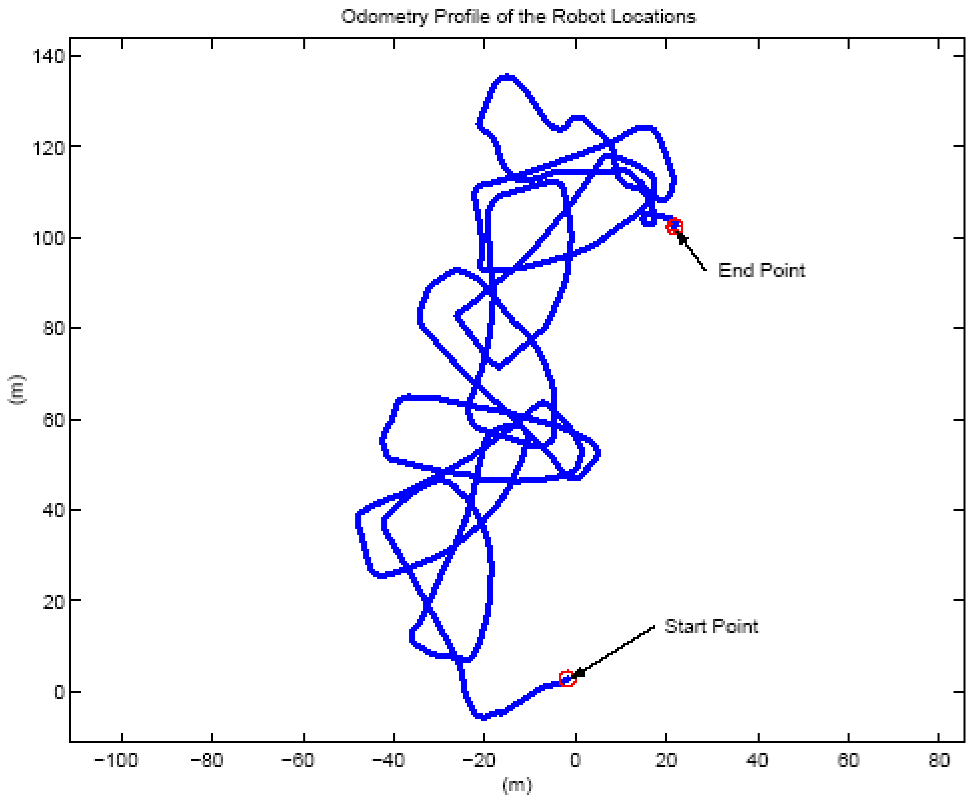
- most Kalman filter approaches are actually EKF SLAM
  - non-linear plant and measurement models
- many other issues not described in class
  - dealing with gross errors in sensor data
  - feature extraction from sensor data
  - landmark correspondences (called the data association problem)
  - 3D instead of 2D motion
  - landmark management
    - choice of landmarks
    - provisional landmarks
    - landmark existence probability
    - addition of new landmarks
    - removal of false landmarks

EKF SLAM Application

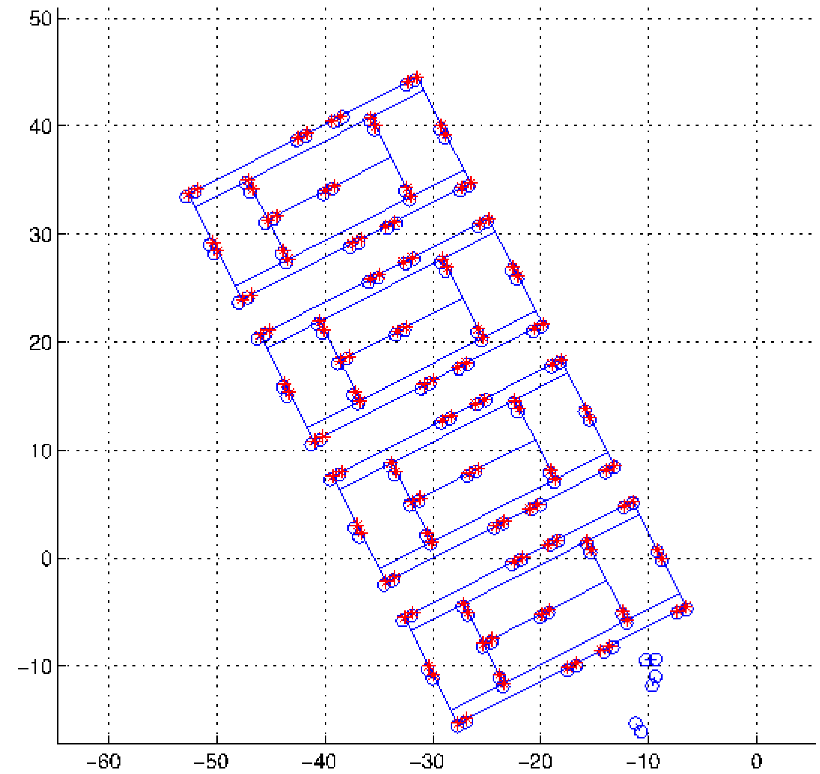
www.probabilistic-robotics.org

[MIT B21, courtesy by John Leonard]

EKF SLAM Application



raw odometry



estimated trajectory

www.probabilistic-robotics.org

[courtesy by John Leonard]

EKF SLAM Application

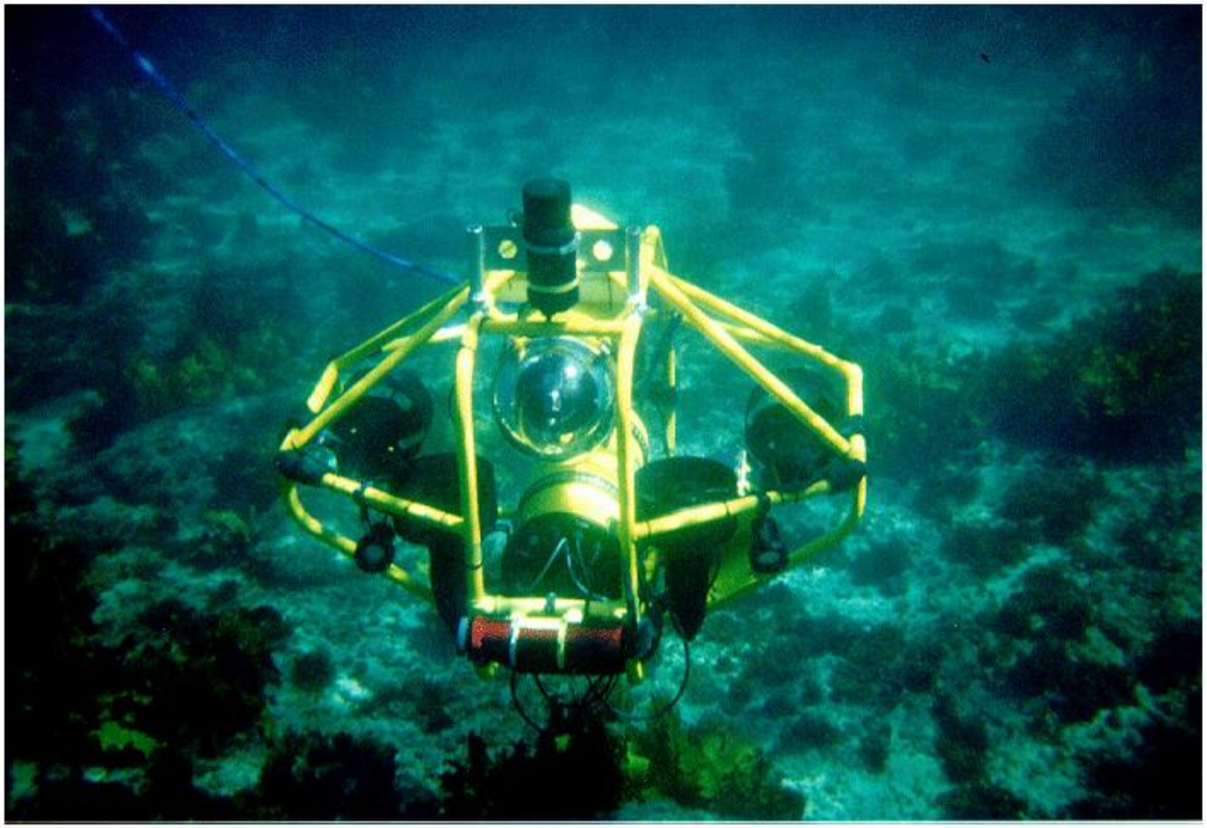


Figure 10.6 Underwater vehicle Oberon, developed at the University of Sydney. Image courtesy of Stefan Williams and Hugh Durrant-Whyte, Australian Centre for Field Robotics.

EKF SLAM Application

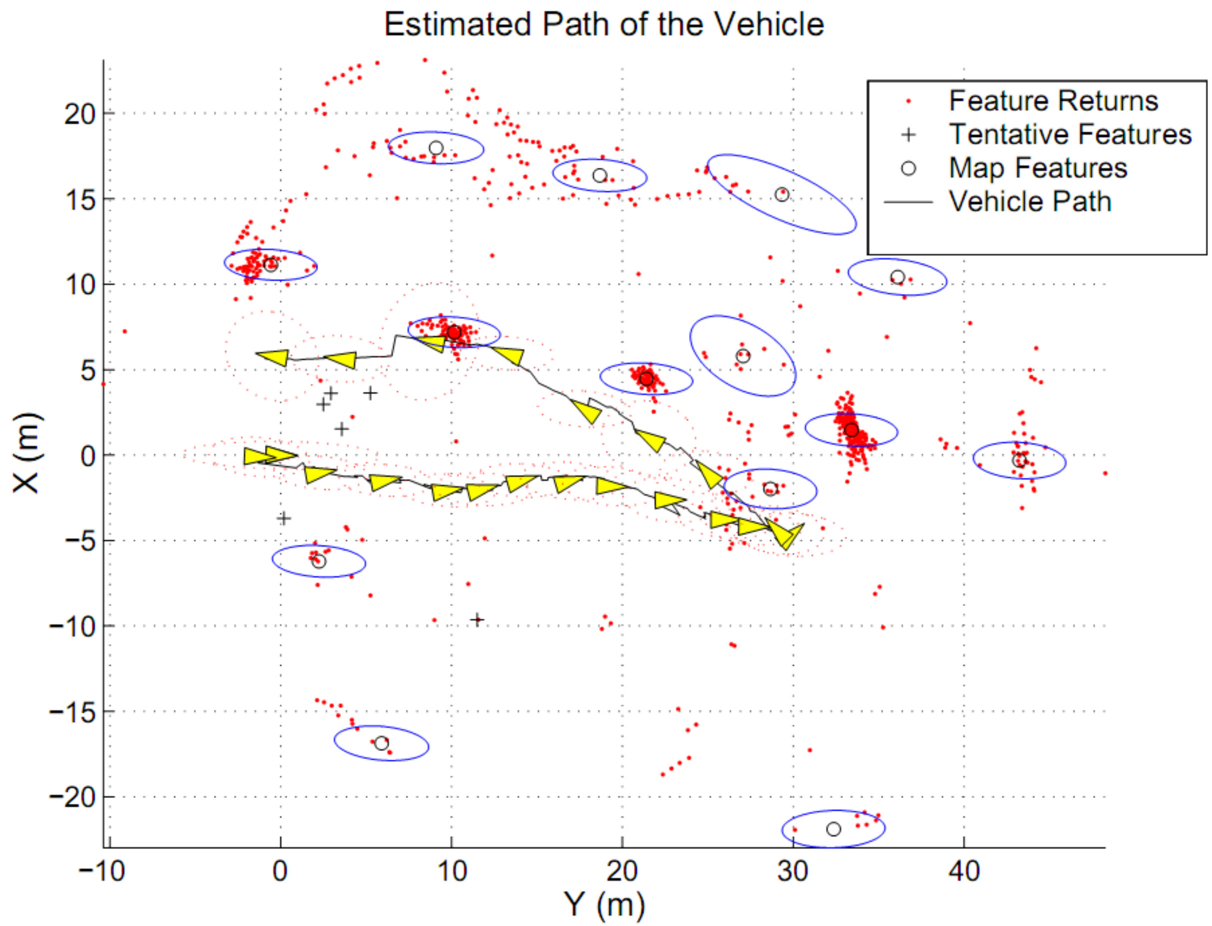


Figure 10.5 Example of Kalman filter estimation of the map and the vehicle pose. Image courtesy of Stefan Williams and Hugh Durrant-Whyte, Australian Centre for Field Robotics.

Bearing Only SLAM

- in the KF SLAM problem we assumed a robot with sensors that could measure the vector from the robot to a landmark
  - similar to, but not exactly the same as, a robot with sensors that can measure distance and bearing
- suppose that a robot has a sensor that can only measure the bearing of a landmark (or multiple landmarks)
  - perhaps using a monocular camera
  - what is a possible measurement model?
    - is it linear?

Bearing Only SLAM

- assuming that the landmarks are uniquely identifiable explain
  - how to use an EKF to solve the bearing only SLAM problem
  - how to use a PF to solve the bearing only SLAM problem