Time Update
$\bar{x}_k = \bar{x}_{k-1}$
$\bar{P}_k = \bar{P}_{k-1} + Q$
Measurement Update
$K_k = \bar{P}_k(\bar{P}_k + R)^{-1}$
$x_k = \bar{x}_k + K(z_k - \bar{x}_k)$
$P_k = (I - K_k)\bar{P}_k$

1. [20 points] You have a round differential drive robot in which the wheelbase is 40cm and the diameter of each wheel is 10 cm. Assume that the drive wheels are mounted along the centerline of the robot and that the center of the wheelbase is located at the center of the robot.

5.6 PTS

a) How fast, in revolutions per second, must the right and left wheels rotate to have the robot move straight ahead at a velocity of 2 meters/second?

5.6 PTS

b) If both wheels are rotated in opposite directions, at what angular rate must the wheels turn to rotate the robot one full revolution in one second?

5.6 PTS

c) You are given that the center of the robot is located at coordinate (x = 4, y = 4, θ = π/4) in the global coordinate system and that at the current instant in time the left wheel is rotating 25 revolutions/second and the right wheel is rotating 32 revolutions per second. Where is the ICC located in the global coordinate system?

a.) $d = 10\ cm \Rightarrow 0.1\ m$

$r = d/2 = .05\ m$

$u = (w_1 + w_2) * r/2$

$w_1 = w_2$

$u = 2\ m/s$

$2 = 2w * \frac{.05}{2}$

$w = 40\ rad/s$

$40\ rad/s * 1/2\pi\ rev/s = 6.366$

b.) $b = 0.4\ m$

$r = 0.05\ m$

$w = 2\pi$

$w_1 = -w_2$

$w_1 = bw/2r = 25.133\ rad/s$

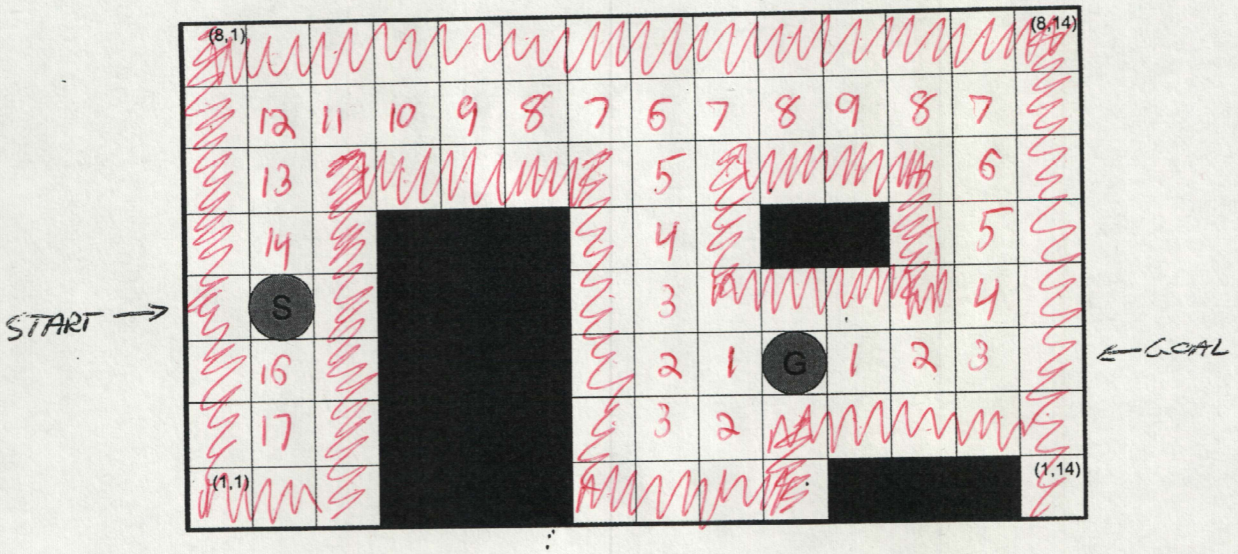
c.) $V_1 = 25\ rev/s * .1\pi\ m/rev = 7.85398\ m/s$

$V_2 = 32\ rev/s * .1\pi\ m/rev = 10.0531\ m/s$

$R_{ICC} = \frac{l}{2}\frac{V_2 + V_1}{V_2 - V_1} = \frac{.4}{2}\frac{10.0531 + 7.85398}{10.0531 - 7.85398} = 1.52857\ m = R$

$I_{cc} = [4 - R\sin(\pi/4);\ 4 + R\cos(\pi/4)] \Rightarrow [2.84843,\ 5.15157]$

2. [20 points] You are given the map below showing a room containing a start position, S, and goal location, G, and a number of obstacles. The border of the room is also an obstacle.



10 PTS
-0.16 PTS/cell missed
(61 cells)

a) Assume the robot completely fills a grid cell. For purposes of path planning, we wish to interpret the robot as a point and thus must expand the walls and obstacles based on the robot's size to determine the viable paths through the space. Since planning is constrained to the grid, if any portion of a grid cell is excluded, the entire cell must be excluded. Based on these criteria, shade those cells in the above occupancy grid from which the robot is excluded.

b) Based on a 4-connected wavefront algorithm, annotate all of the permissible cells of the occupancy grid from part a above by numbering the cells of the graph according to the wavefront algorithm **and** identify one of the paths leading from the start to the goal.

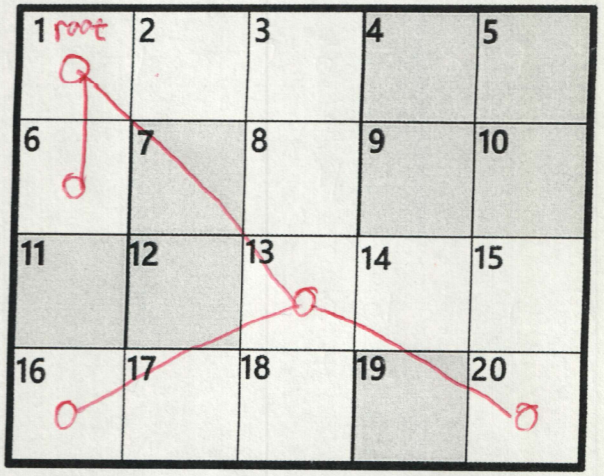
10 PTS

- 2 PTS Start numbering at Goal (or state searching backwards)

- 5 PTS correct numbering method

- 3 PTS correct path

3. [20 points] You are provided the following 4-connected occupancy grid (shaded areas represent obstructions):

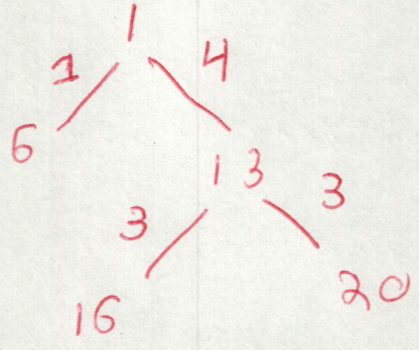


Represent this occupancy grid as a graph using grid location 1 as the root node. When creating this graph, each traversal from one grid to another increases the cost of the path by one unit. For example, going from node 1 to node 3 has a cost of 2 units because there are two traversals. Further, when creating this graph, in cases where there is no choice of the path that can be followed, don't illustrate each intermediate node – only show the node at which the decision is made.

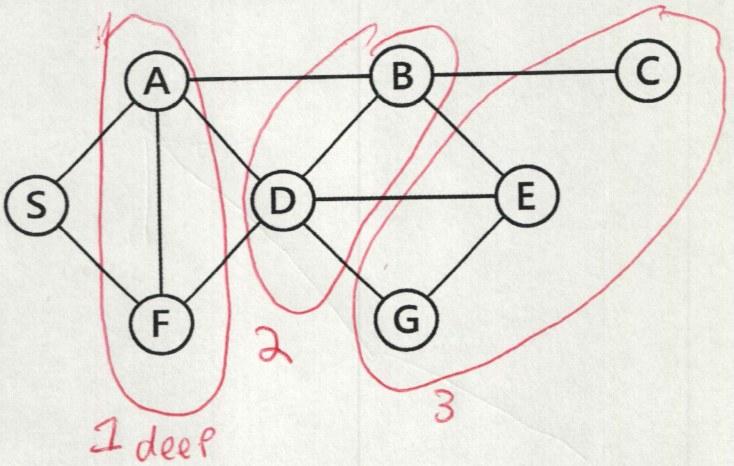
a) Illustrate the occupancy grid as a graph using a tree diagram, showing the nodes and the weights on the edges between nodes. (hint: there will be 5 nodes).

10 PTS Tree

10 PTS weights



4. [20 points] For the following graph, determine the order in which nodes are expanded, starting with node "S," if a breadth-first search strategy is used (ties must be broken alphabetically):



BFS

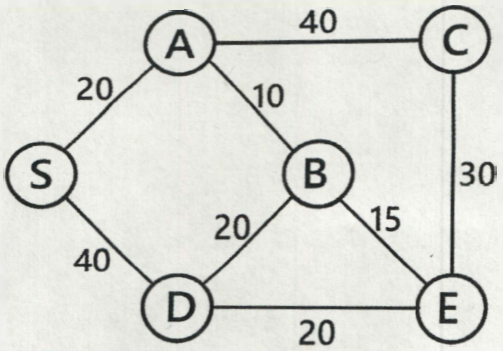
S
A
F
B
D
C
E
G

DFs has many solutions as graph is not directed.

1 example

S
A
B
C
E
G
D
F

5. [20 points] Given the following graph and the provided list of straight-line distances, determine the optimal route from S to E using the A* algorithm:



Straight-Line Distance Between Nodes	
S → E	35
A → E	25
B → E	10
C → E	20
D → E	15

1.) S→A = 20 + 25 = 45

S→D = 40 + 15 = 55

A Chosen 5 PTS

2.) A→B = (20+10) + 10 = 40

A→C = (40+20) + 20 = 80

B Chosen 5 PTS

past frontier

S→D = 40 + 15 = 55

3.) B→E = (20+10+15) + 0 = 45

B→D = (20+10+20) + 15 = 65

E Chosen 5PTS

Past frontier

S→D = 40 + 15 = 55

A→C = (40+20) + 20 = 80

Path:

S → A → B → E

Localization – gMapping and Particle Filters I

Prof. Michalson

What is gMapping?

- gMapping is a popular SLAM algorithm implemented in ROS
  - SLAM – Simultaneous Localization and Mapping
  - gMapping implementation details can be found at openslam.org
  - gMapping uses a “Rao-Blackwellized Particle Filter” to create maps
    - “Rao-Blackwellized” refers to the math used to select “particles”
    - A “particle filter” is a technique for localizing relative to a map.

The Problem

- You are given:
  - Observations $z_{1:t} = z_1, \ldots, z_t$
  - Odometry measurements $u_{2:t} = u_1, \ldots, u_t$
- We need to find:
  - Location with (posterior) probability
    - $P(x_{1:t}, m \mid z_{1:t}, u_{2:t})$
    - m is the map (occupancy grid)

Example Particle Filter Animation

(https://youtu.be/aUkBa1zMKv4)

Matlab code is on-line

Two localization problems

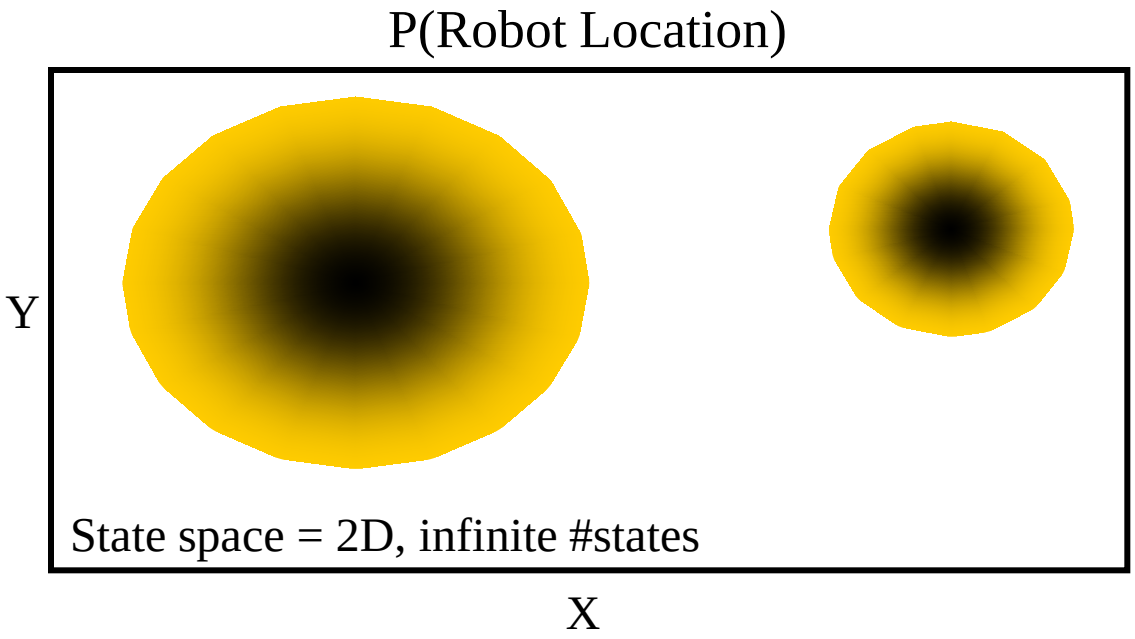
- In general, there are at least two localization problems we need to address:

- "Global" localization
  - figure out where the robot is, but we don't know where the robot started
  - Sometimes called the "kidnapped robot problem"

- "Position tracking"
  - figure out where the robot is, given that we know where the robot started

Markov Localization

- Key idea: compute a probability distribution over all possible positions in the environment.
  - This probability distribution represents the likelihood that the robot is in a particular location.



Markov Localization

- Perception (or sensing) model: represents likelihood that robot senses a particular reading at a particular position (related to our discussion last class)

$$P(x) = \alpha P(o|x)P(x)$$

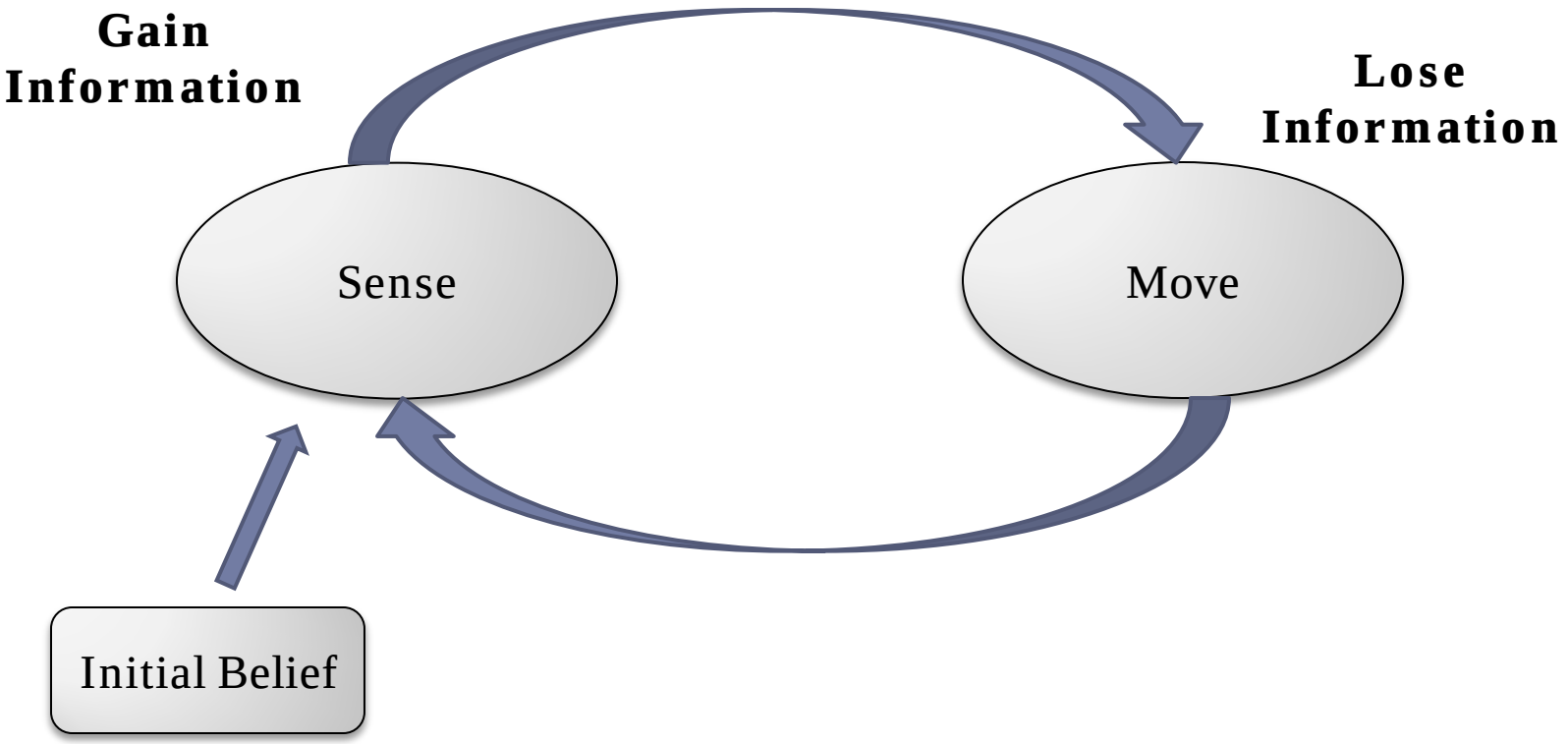
"Probability that robot will perceive o, given that the robot is in position x, times the likelihood the robot is in position x"

- Action (or motion) model: represents movements of robot

$$P(x) = \sum P(x|a, x')P(x')$$

Probability that action a from position x' moves the robot to position x, times the likelihood the robot is in position x', summed over all possible ways robot could have reached position x"

Localization



Particle Filter

- Represent the belief that a robot is at a particular location by a set of "samples", or "particles"

  Represent $p(x)$ by set of N weighted, random samples, called *particles*:

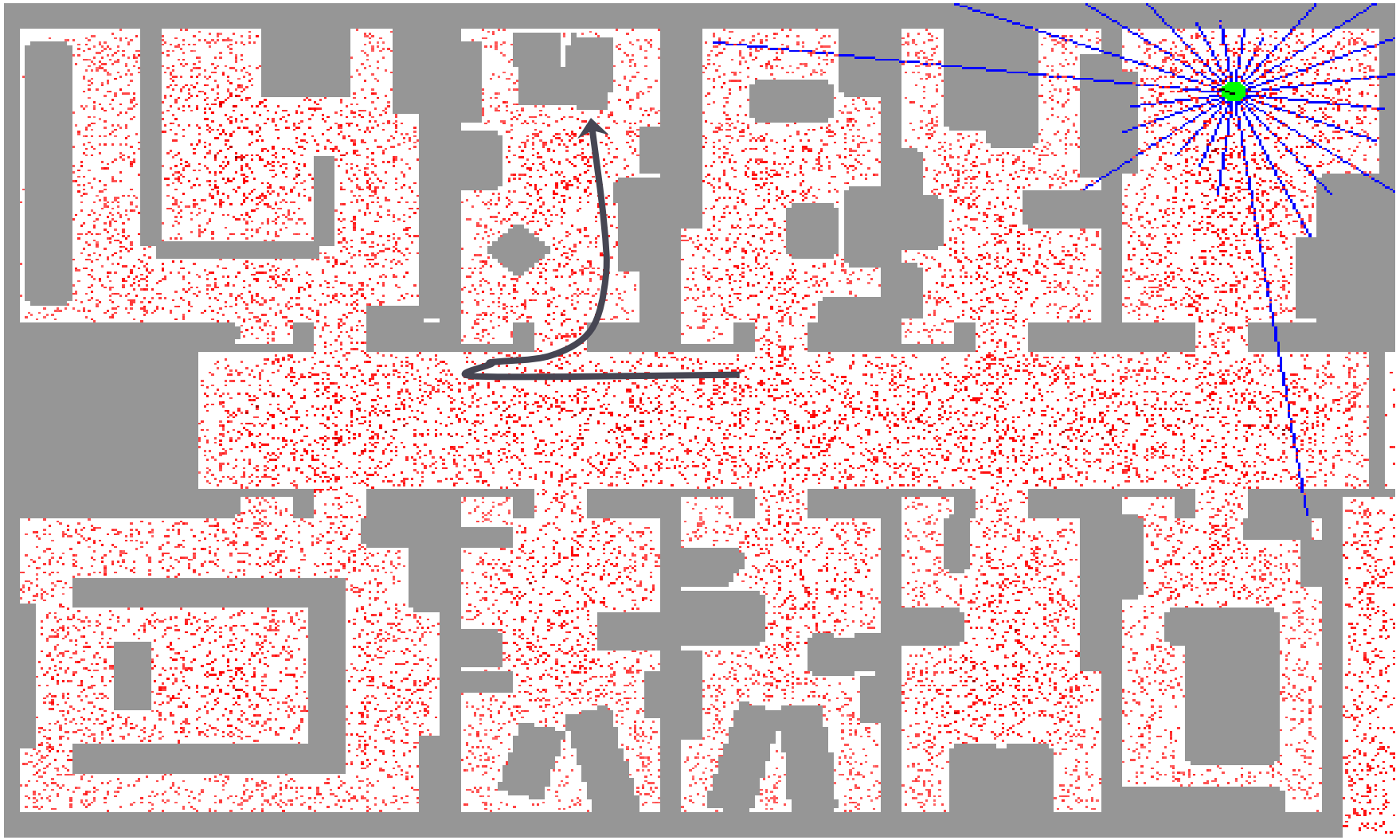
  where a sample, s_i, is of the form: $\langle\langle x, y, \theta\rangle, p\rangle$

  $\langle x, y, \theta\rangle$ represents robot's position

  p represents a weight, where sum of all p's is 1 (analogous to discrete probability)

- Allows us to:
  - Maintain multiple estimates of robot's location
  - Track possible robot positions, given all previous measurements

Particle Filter Localization (using sonar)



http://www.cs.washington.edu/ai/Mobile_Robotics//mcl/animations/global-floor-start.gif

Kalman Filter Examples I
(by the numbers)

Prof. Michalson

Kalman Filter at a Glance

- x =state estimate
- P =uncertainty covariance
- A =state transition matrix
- B =control input function
- u =motion vector
- Q = motion noise

- z =measurement
- H =measurement function
- R = measurement noise

- I =identity matrix

Time Update (Motion)

$$\bar{x}_k = Ax_{k-1} + Bu_k$$
$$\bar{P}_k = AP_{k-1}A^T + Q$$

Measurement Update

$$K_k = \bar{P}_k H^T (H\bar{P}_k H^T + R)^{-1}$$

$$x_k = \bar{x}_k + K(z_k - H\bar{x}_k)$$

$$P_k = (I - K_k H)\bar{P}_k$$

Building a Kalman Filter

- The Kalman Filter relies on an adequate model of the system. When approaching filter design:
  - Understand your problem – what are the important dynamics to model?
  - Create a state transition model – Start simple!
  - Create a measurement model – How do measurements indicate state?
  - Create a noise model – Basic equations assume white noise.
  - Test the filter – Evaluate performance.
  - Refine – Tune noise parameters or state/measurement models.

Estimating Position from a Wall

- Step 1 – Understand the Problem:
  - Consider one-dimensional problem with a robot using a laser to measure distance from a wall.
    - The robot can be stationary, is moving towards, or is moving away from the wall.
    - There is noise in the sensor that will result in changes in the range measurement even if the robot is stationary.

Estimating Position from a Wall

- Step 2 – Create a state transition model:
  - Let's start with the simplest possible model:
    - Assume the robot is stationary.
    - This means the distance, r_W, from the wall is constant. i.e. $r_W = c$.
    - If the state is defined as the position, x, this means that the state transition is $x_k = x_{k-1}$.
    - This also means x is scalar, the system matrix, A, is scalar and that A = 1
    - Stationary means no drive input so B = u = 0.
    - Thus, $\bar{x}_k = Ax_{k-1} + Bu_k = x_{k-1}$

Estimating Position from a Wall

- Step 3 – Create a measurement model:
  - In this case we only have a measurement of distance to the wall:
    - The measurement is represented by z_k.
    - In this case it is the scalar value of the quantity we want to measure.
    - We will assume that it is also of the same scale and units of the state variable, x_k, that we want to determine, so no modification is needed.
    - This means H =1, so: $z_k = H\bar{x}_k = \bar{x}_k$

Estimating Position from a Wall

- Step 4 – Create a noise model:
  - Assume that the laser range finder provides a noisy measurement:
    - The measurement noise has variance R, which is a scalar value.
    - The process noise will also be a scalar value, Q, which is generally not available.
    - Since the problem is scalar, the noise error covariance, P, is also a scalar.

Estimating Position from a Wall

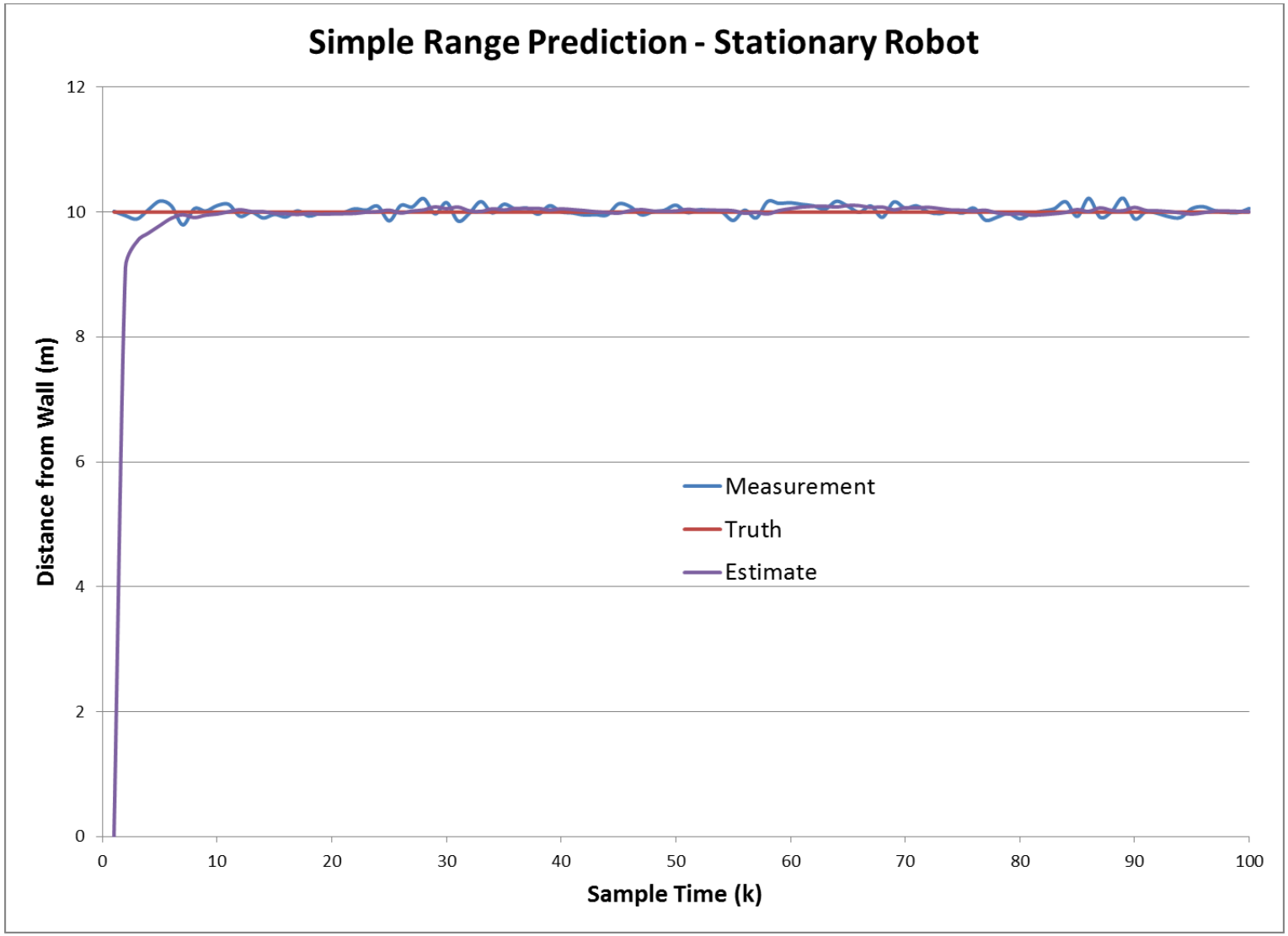
- Step 5 – Testing the resultant filter:
  - Based on our model, the time update equations are:

$$\bar{x}_k = x_{k-1}$$
$$\bar{P}_k = P_{k-1} + Q$$

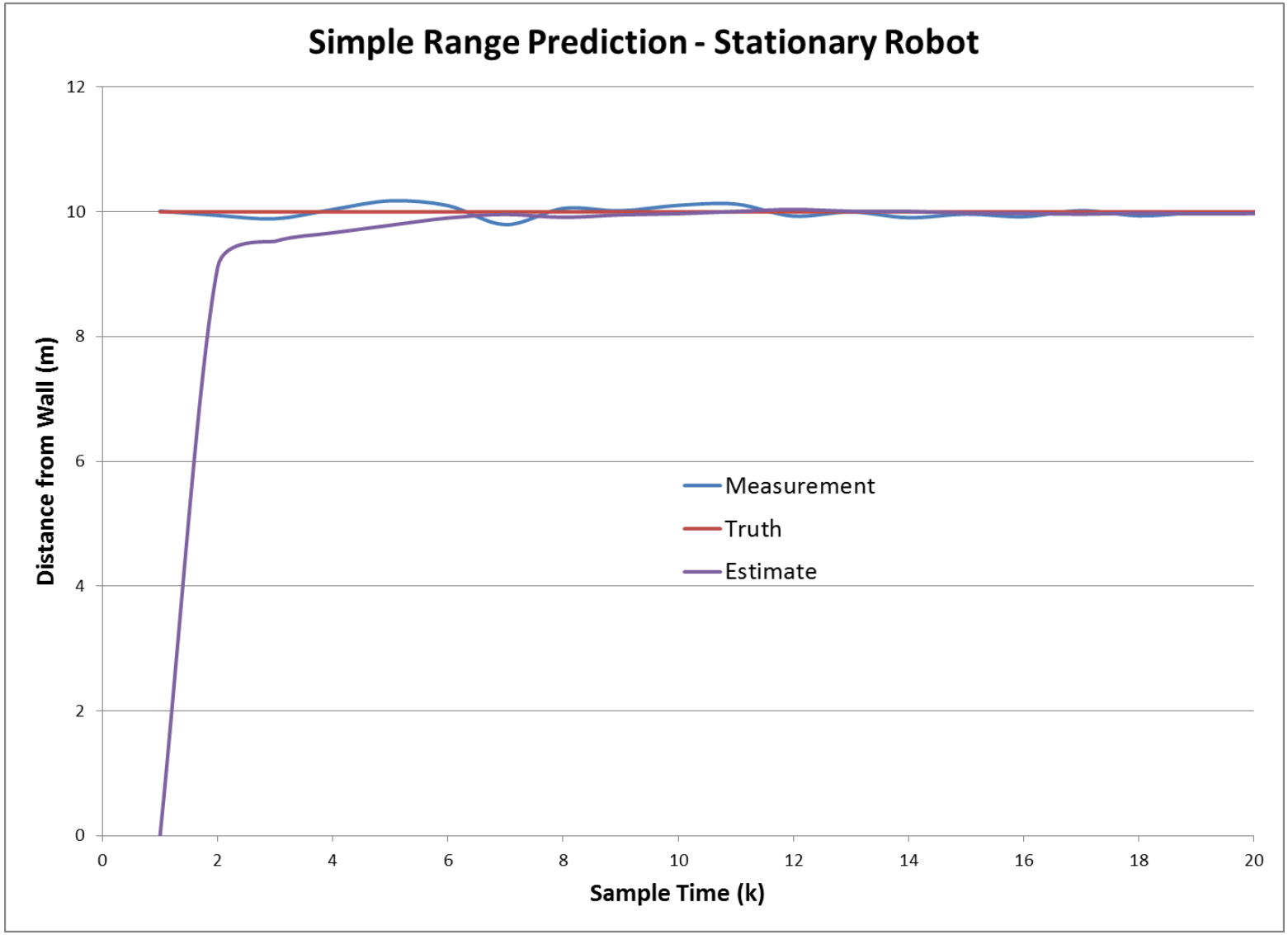
  - The measurement update equations are:

$$K_k = \bar{P}_k(\bar{P}_k + R\ )^{-1}$$
$$x_k = \bar{x}_k + K(z_k - \bar{x}_k)$$
$$P_k = (I - K_k)\bar{P}_k$$

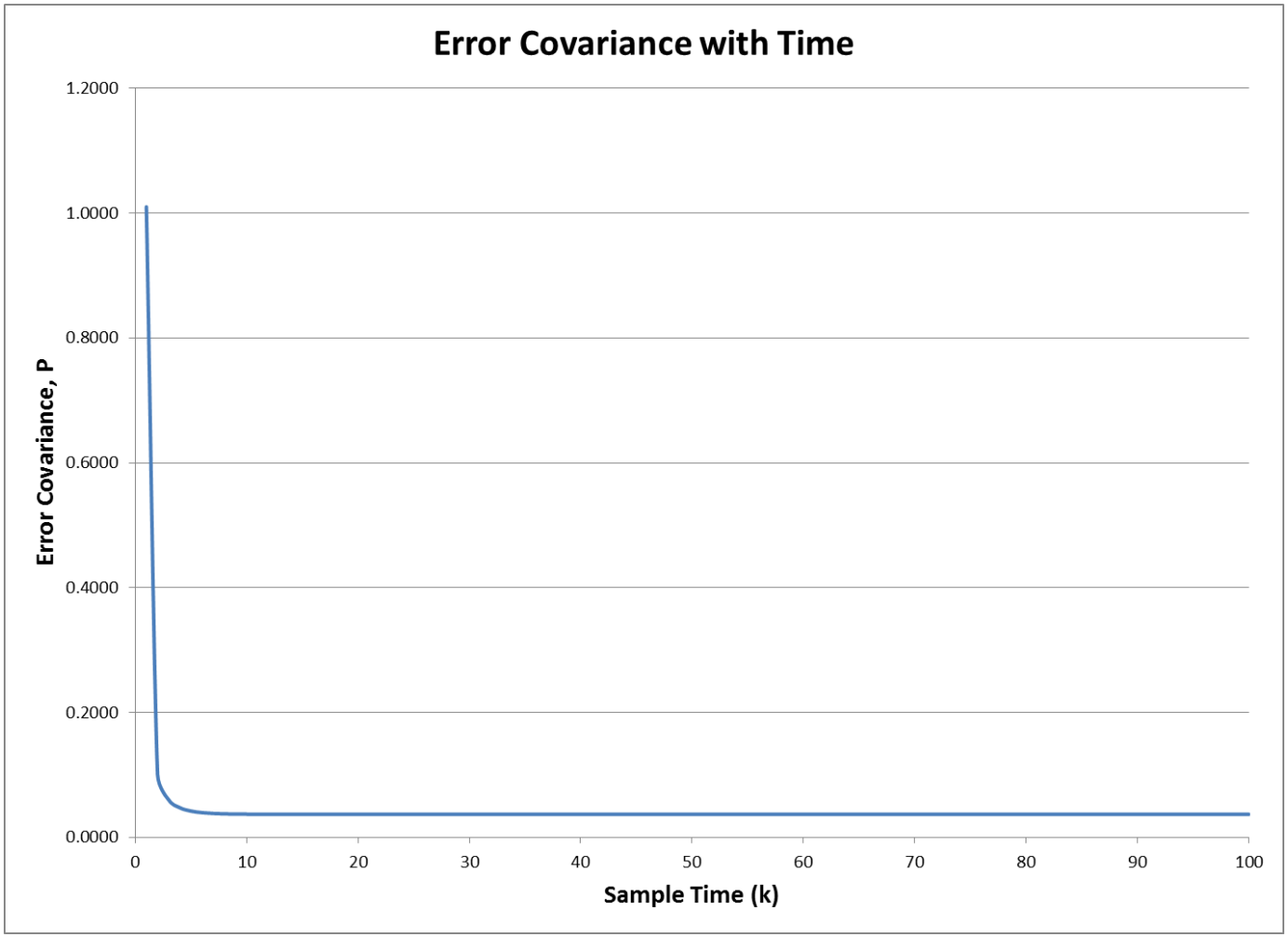
Simple Filter Performance



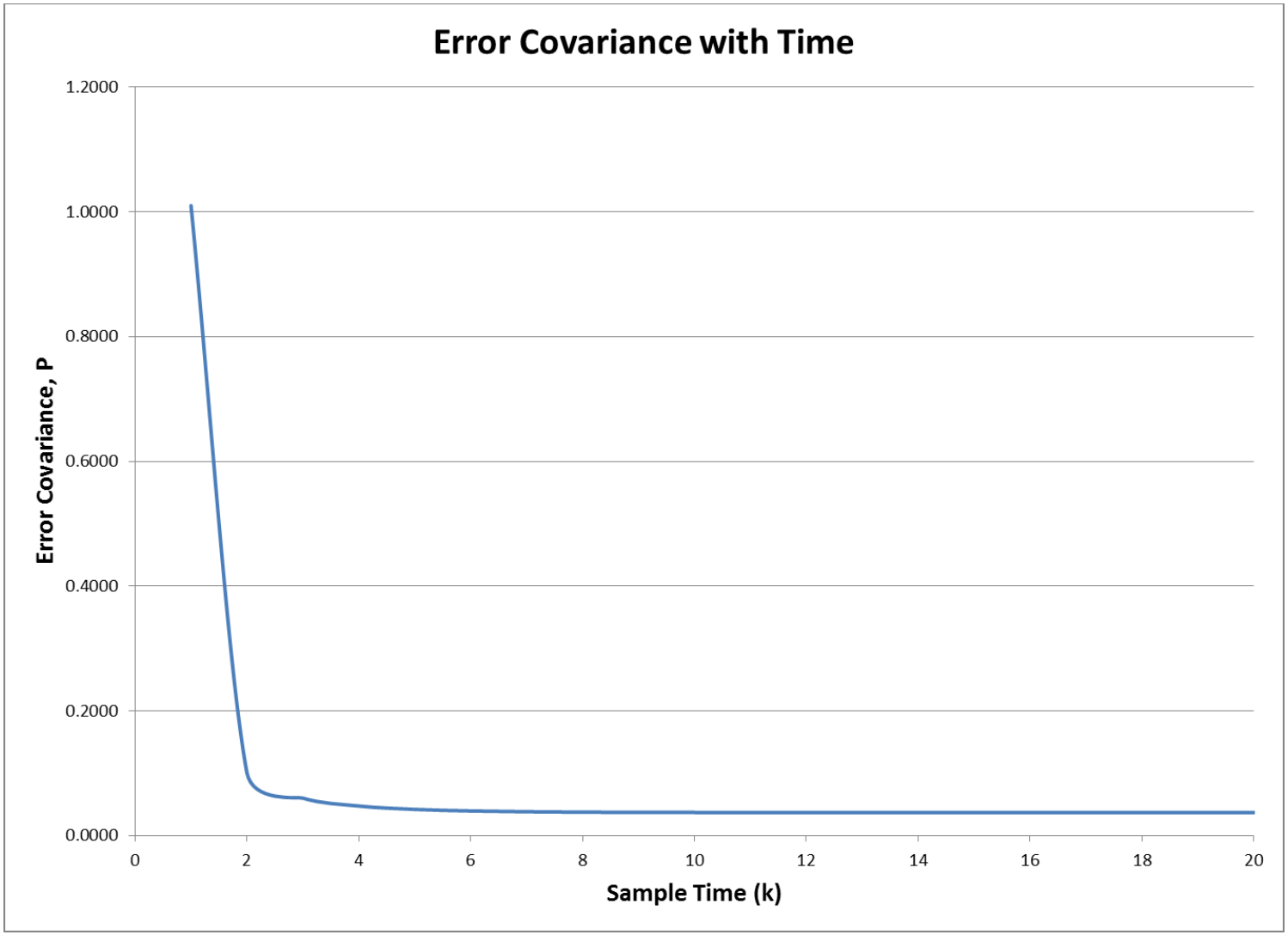
Simple Filter Performance - Closeup



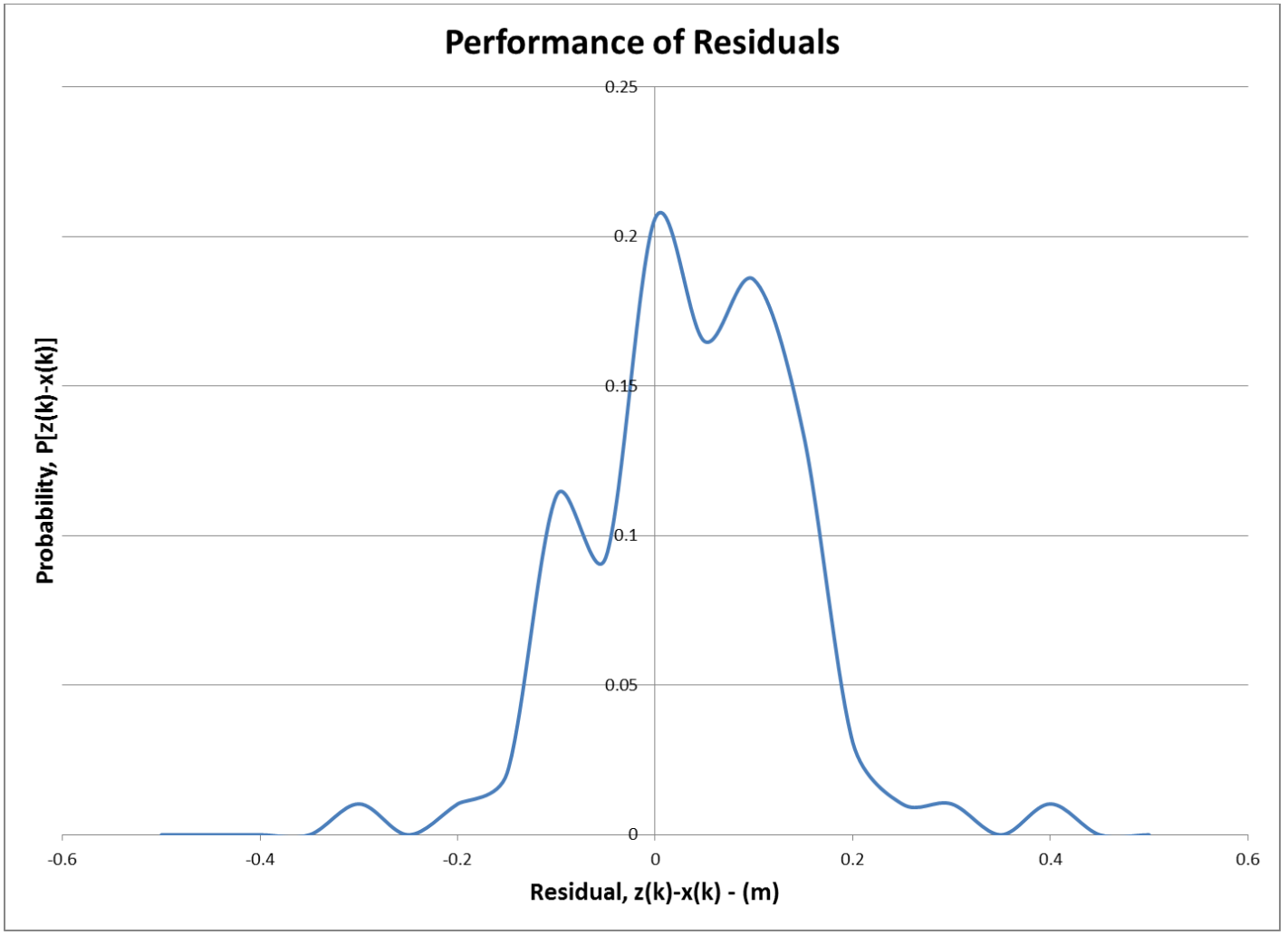
Convergence of Error Variance



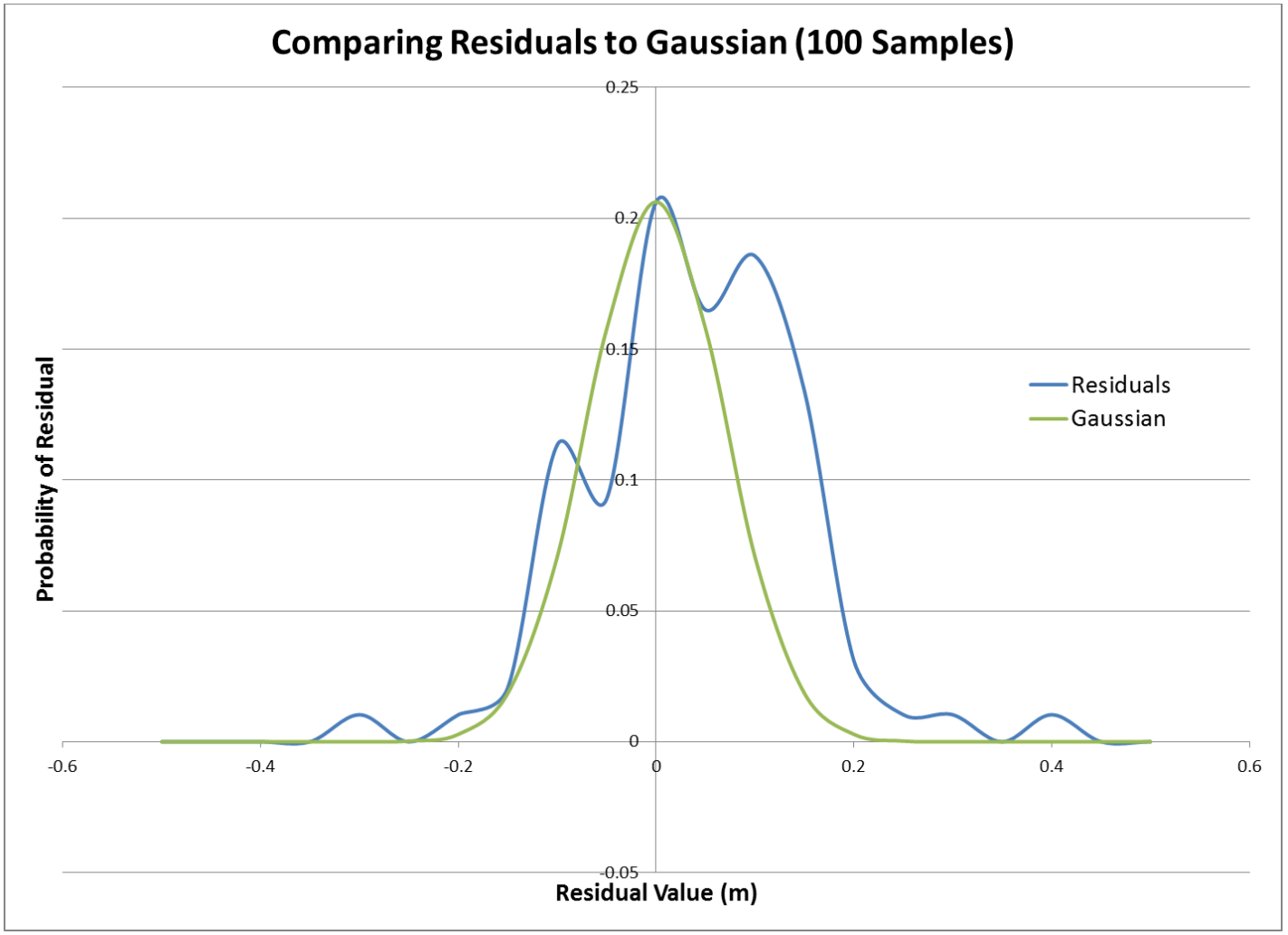
Convergence of Error Variance - Close-up



Residual Behavior



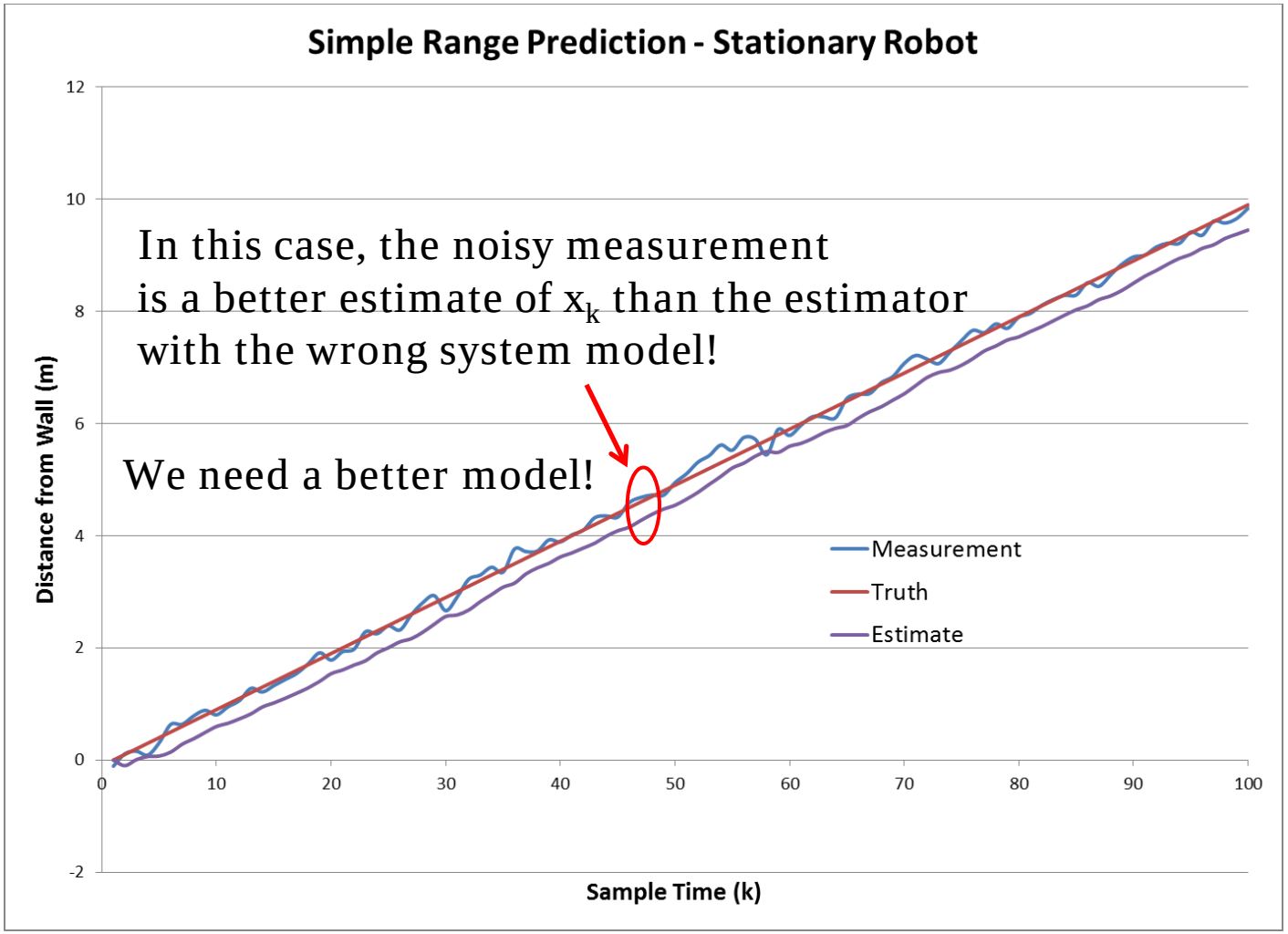
Analyzing Residual Behavior



What if the Model is Wrong?

- If the system model is wrong, then the filter can't be expected to follow the actual system dynamics.
- However, it will try.
- There's no need to complicate matters if the simple solution is adequate – test the filter and decide.
- If better accuracy is needed, the system and/or measurement model needs to better model the actual system.

What happens if the Model is Wrong?



HOMEWORK ASSIGNMENT 5

(Due: Friday, April 29, 2016; 8pm)

1. Assume a simple model of a robot with a range sensor mounted on a stationary platform. Assume that the range is initialized to 8 meters, that the error covariance is initialized to 50, the process noise is assumed to be 0.01 and the measurement noise is r = 0.1. Calculate the value of the Kalman Filter estimate of state given the following measurements:

Z_k	X_k
7.75	
7.28	
7.83	
7.55	
7.68	

2. The file RBE3002_D16_homework_5_range_data.csv contains 200 range measurements, measured every 100 ms in units of centimeters. Using a Kalman Filter based on the PV model, process this data to determine an estimate of position and velocity for the robot. In this model, assume that the initial position and velocity are zero, that the variance of the measurement noise is 0.1, and that the initial P and Q matrices are given by:

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } Q = \begin{bmatrix} 0.000333 & 0.0005 \\ 0.0005 & 0.001 \end{bmatrix}$$

   a. Plot the estimate of range versus time for the 200 samples
   b. Comment on the plot of the position (range) estimate – what is the behavior around the time of initialization of the filter and at the discontinuity?
   c. What are the final position and velocity estimates of the robot?

3. Consider an empty room that measures 20m x 10m. The origin of the room is at the origin of an X, Y coordinate system and the room is positioned such that the 20m side lies along the X axis. Robot location is given as a 3-tuple, (X, Y, θ), where X and Y are the position relative to the origin in meters and θ is the rotation where the Y axis represents 0 degrees and the X axis represents 90 degrees. The robot makes four range measurements, one each at 0, 90, 180, and 270 degrees (in that order). Localization of the robot is to be performed using a particle filter in which the metric of "goodness of fit" is | measured range – particle range |. The measured range at 0, 90, 180 and 270 is 6.5m, 18m, 3.5m and 4.25m. The particles are located at P1 = (5, 5, 45), P2 = (2, 8, 335), P3 = (15, 5, 90), and P4 = (3, 3, 345).
   a. What are the "goodness of fit" values for each particle?
   b. Which particle is most likely near the robot location?
   c. If you were going to create 100 more particles, approximately how many would you locate in the vicinity of each of P1, P2, P3 and P4?