INTRODUCTION TO KALMAN FILTERS

1.1: What does a Kalman filter do?

- A Kalman filter is a tool—an algorithm usually implemented as a computer program—that uses sensor measurements to infer the internal hidden state of a dynamic system.

- Some examples help to give an idea of how this might be useful:

  - Using radar angle and range measurements in an air-traffic control system to determine the three-dimensional location and speed of aircraft approaching an airport (tracking);

  - Using live-motion video capture of marker dots placed on actors to track physical movement and aid creation of animation in films;

  - Using measurements of external electrical and thermal conditions of battery cells to determine internal "state of charge" and "state of health" of a hybrid electric vehicle battery pack;

  - Using measurements of the earth's magnetic field, GPS, and accelerations to determine my own position, orientation, and velocity in three dimensions (navigation);

  - Using financial market data to predict future prices of traded securities and commodities;

  and many others.

- Because the Kalman filter is a tool, it is very versatile. Its application areas are very diverse. Caution: If all you have is a hammer, everything looks like a nail! Same with Kalman filters!

- Kalman filters estimate the <u>state</u> of a dynamic system.

- A system's state is a vector of values that completely summarizes the effects of the past on the system.

- An example is the present position, orientation, linear and angular velocity of an aircraft: how the aircraft came to be in this state is not relevant; all future behavior depends only on the present state and future inputs (engine thrust, wind, etc.) to that aircraft.

- We cannot generally measure the state of a dynamic system directly. Even if we can, we often choose not to because it is too expensive or too complex.

- Instead, we make measurements that are somehow related to the state. They are linear or nonlinear functions of members of the state.

- Consider the following <u>model</u> of a linear dynamic system

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

$$z_k = Cx_k + Du_k + v_k.$$

- Here:

  - x_k is the system state vector at time index k.
  - u_k is the measurable (deterministic) input to the system.
  - w_k is the disturbance or process noise: an unmeasurable input that affects the state.
  - z_k is a sensor measurement, somehow related to x_k.

- v_k is sensor noise that corrupts the measurement.

- A, B, C, and D are matrices, specific to the system we are observing, that describe its dynamic behavior.

- The first equation is the "state equation" or "process equation" that describes how the state evolves over time.

- The second equation is the "output equation" that describes how the measured output relates to the state.

- We will see many models of this (and similar) form this semester.

- For example, consider the one-dimensional motion of a rigid object.

  - The state comprises position p_k and velocity (speed) s_k

$$\underbrace{\begin{bmatrix} p_k \\ s_k \end{bmatrix}}_{x_k} = \underbrace{\begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} p_{k-1} \\ s_{k-1} \end{bmatrix}}_{x_{k-1}} + \underbrace{\begin{bmatrix} 0 \\ \Delta T \end{bmatrix}}_{B} u_{k-1} + w_{k-1},$$

  where ΔT is the time interval between iterations $k-1$ and k.

  - The measurement may be a noisy position estimate

$$z_k = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} p_k \\ s_k \end{bmatrix} + v_k.$$

1.2: The basic idea of the Kalman filter

- Our goal is generally to estimate as well as we can (in some sense) the values of state vector x_k given all past measurements $z_0, z_1, \ldots z_k$.

- A helpful way to think about the problem is in terms of conditional probabilities
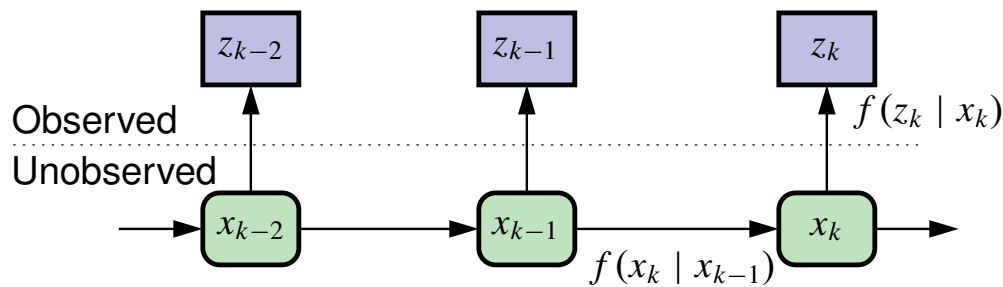
- The observations allow us to "peek" at what is happening in the true system. Based on observations and our model, we estimate the state.

- Discrete-time Kalman filters (the focus of this course) repeatedly execute two steps:

  1. <u>Predict</u> the present state value based on all past available data. For example, a linear Kalman filter computes

  $$\hat{x}_k^- = A\hat{x}_{k-1}^+ + Bu_{k-1},$$

  where $\hat{x}_k^-$ is the prediction of x_k.

  2. <u>Estimate</u> the present state value, by updating the prediction based on all presently available data. For example, a linear Kalman filter computes

  $$\hat{x}_k^+ = \hat{x}_k^- + L(z_k - (C\hat{x}_k^- + Du_k)).$$

- A very straightforward idea. But...

  - What should L be?

- How do we make this optimal?

- What about nonlinear systems?

- What if we don't know u_k (as in the tracking application)?

- There are lots of nuances. You can "learn" KF in a week, but you can also spend years exploring the breadth and depth of the subject.

- Our focus will be on learning the most useful forms of KF and how to apply them robustly to real problems.

The background to what we will study in this course

- Because a Kalman filter is a kind of estimator, we will need to study some ideas from estimation theory.

- Because disturbances and sensor noises are not deterministic, we will need to already know basic (undergraduate) probability (really well) and study some ideas from random process theory.

- Because the models we use are based very heavily on matrix math, we will need to already know the mechanics of linear algebra (really well) and we will need to study some ideas from linear and nonlinear system theory.
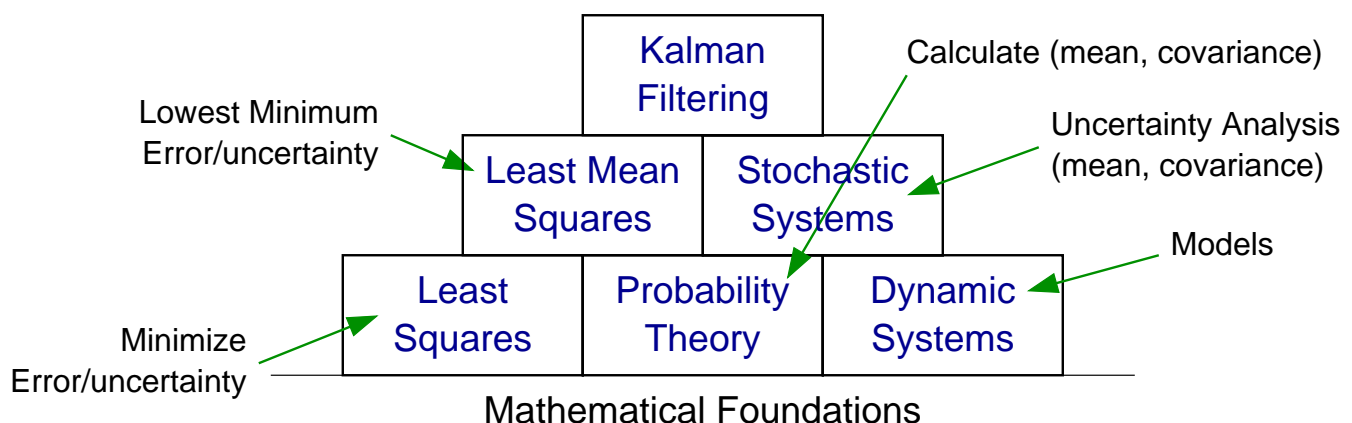
- It is possible to go really deep into any one of these individual areas—the more background you have the better—but our focus will be on developing methods that we can *apply* to real problems.

- The lectures will be primarily theoretical, and the homework/projects will be primarily applications.

- Some knowledge of the Laplace, z, and Fourier transforms is helpful to understand certain topics covered in lecture, but will not be critical for implementing Kalman filters and succeeding in this course.

- We will cover only the background material that we will ultimately apply. (The textbooks have much more information that can add to the richness of your experience in this course.)

- Be forewarned: You will see that I use different mathematical notation from the textbooks.

  - There are multiple conventions in common use. I choose the most compact one that is most compatible with other courses I teach.

  - I include a glossary of notation in each chapter of notes to aid your interpretation/translation of one system of notation versus another.

- We will make *extensive* use of MATLAB.

  - Prior experience with MATLAB is not necessary. We will teach what is needed as we go.

  - However, general programming knowledge is very important (variables, procedures, conditions, loops. . .).

1.3: Examples of applications of Kalman filtering

- Knowledge of how KF works is necessary to be able to adapt it to new situations.

- It is not generally sufficient to simply implement a toolbox function.

- The following examples are ones that my students and I have been directly involved with.

OBJECTIVE: Track marker dots on actors.

- State: x, y position and velocity of dots in frame.

- Observation: x, y positions of dots in frame (unlabeled).

- Issues: Data association, tracking when dots are obscured.
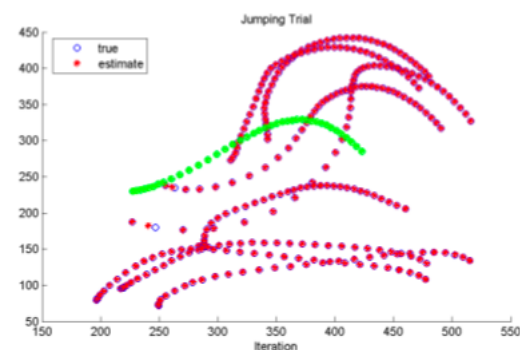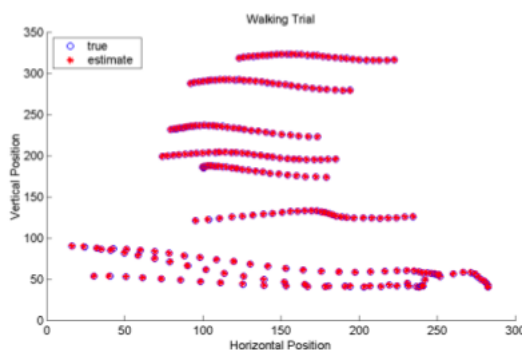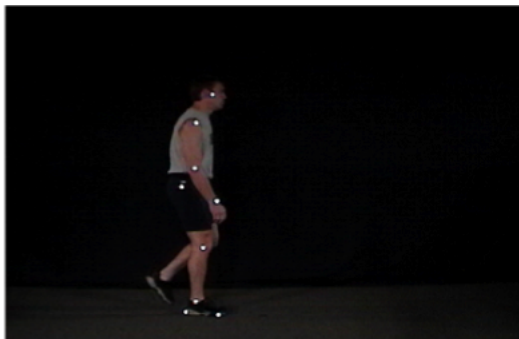
OBJECTIVE: Localize bad guys.

- State: x, y position and velocity.

- Observation: Radar azimuth/elevation (maybe range).

- Issues: Nonlinear relationship between measurements and position; measurements arriving to KF out-of-sequence.
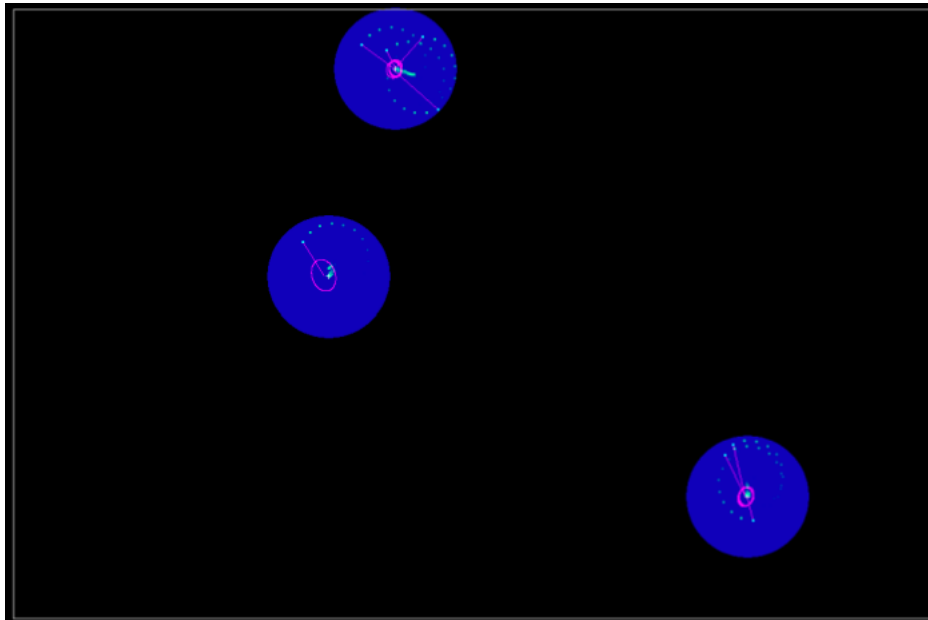
OBJECTIVE: SOC estimation for battery cells.

- State: SOC, polarization voltages, hysteresis voltages.
- Observations: Current, temperature, and voltage under load.
- Issues: Lack of available simple battery model, nonlinear dynamics and measurement.

OBJECTIVE: SOH estimation for battery cells.

- State: Resistance and capacity of battery cells.
- Observations: Current, temperature, and voltage under load.
- Issues: Parameter estimation; not state estimation.

OBJECTIVE: Localize myself (navigation).

- State: Position, orientation.
- Observations: Accelerations, rotations, GPS fixes.
- Issues: Correct for drift of IMU using GPS.

Next steps

- Review of matrix algebra (see Appendix A).

- Modeling systems in "state space form."

- Review of random processes.

- Kalman filter derivation.

- Then, more advanced topics: How do we make this thing actually work?