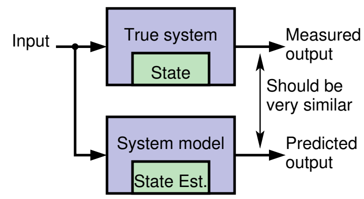
Model-based state estimation

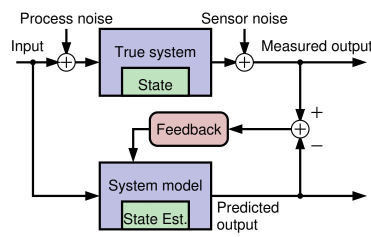
- KFs use sensed measurements and a mathematical model of a dynamic system to estimate its internal hidden state.
- A model of the system and its state dynamics is assumed to be known.
- A system's state is a vector of values that completely summarizes the effects of the past on the system.
- Model's state should mirror system's state.



- An example of a system state vector is the set of values comprising the present position, orientation, linear and angular velocity of an aircraft.
  - How the aircraft came to be in this state is not relevant to predicting future behavior, which depends only on the present state and future inputs (engine thrust, wind, etc.).

Why do we need a model?

- We cannot generally measure the state of a dynamic system directly. Even when we can, we often choose not to do so because it is too expensive or too complex.
- Instead, we measure the system input and then propagate those measurements through the model, updating the model's prediction of the true state.
- We make measurements that are linear or nonlinear functions of members of the state.
- The measured and predicted outputs are compared.



- The KF is an algorithm that updates the model's state estimate using this prediction error as feedback regarding the quality of the present state estimate.

What kind of model do we assume?

- Linear KFs use discrete-time state-space models of the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ z_k &= Cx_k + Du_k + v_k. \end{aligned}$$

- 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 that describe the specific system we are observing.
- First equation ("state equation" or "process equation") describes state evolution.
- Second equation ("output equation" or "measurement equation") describes how the measured output relates to the state.

A simple example model

- Concrete example: Consider the 1-d 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.
  - u_k is equal to force divided by mass; w_k is a vector that perturbs both p_k and s_k.
  - The measurement could be a noisy position estimate:

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

- Example illustrates how the state-space form can model a specific dynamic system.
- The form is extremely flexible: can be applied to *any* finite-dimensional linear system.

1.1.2: What are some key Kalman-filter concepts?

Why do we need feedback?

- Our goal is to make an optimal estimate of x_k in:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ z_k &= Cx_k + Du_k + v_k. \end{aligned}$$

- If we knew x_0, w_k, and v_k perfectly, and if our model were exact, there would be no need for feedback to estimate x_k at any point in time. We simply simulate the model!
  - But, we rarely know x_0 exactly; and, we never know w_k and/or v_k (by definition).
  - Also, no physical system is truly linear and even if one were, we would never know A, B, C, and D exactly.
- So, simulating the model (specifically, simulating the state equation "open loop") is not sufficient to make a robust estimate of x_k.
- Feedback allows us to compare predicted z_k with measured z_k to adjust x_k.

1.1.2: What are some key Kalman-filter concepts?

How does the feedback work?

- Discrete-time Kalman filters repeatedly execute two steps:
  1. Predict the present state-vector values based on all past available data. For example, a linear KF computes ($\hat{x}_k^-$ is the prediction of x_k):

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

  2. Estimate the present state value by updating the prediction based on all presently available data. For example, a linear KF computes ($\hat{x}_k^+$ is the estimate of x_k):

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

- A very straightforward idea. But...
  - What should be the feedback gain matrix L_k?
    - That is, how do we make this feedback optimal in some meaningful sense?
  - Can we generalize this feedback concept to nonlinear systems?
  - What if we don't know u_k (as in the tracking application)?

Summary

- KFs use sensed measurements and a mathematical model of a dynamic system to estimate its internal hidden state.
- For the kind of KFs we will study, the mathematical model must be formulated in a discrete-time state-space format.
- This form is very general, and can apply to nearly any dynamic system of interest.
- KFs operate by repeatedly predicting the present state, and then updating that prediction using a measured system output to make an estimate of the state.
- This process is optimized by computing an optimal feedback gain matrix L_k at every timestep that blends the prediction and the new information in the measurement.
- There is a lot to learn, and the next topic will present our roadmap for doing so.