Kalman Filter Theory and Applications
Heuristic Overview

Michael L. Carroll

August 17, 2020

- Understanding the Equations: Heuristic Introduction

- Understanding the Equations: Heuristic Introduction
- Equation Drilldown: Taking the Equations Apart

- Understanding the Equations: Heuristic Introduction
- Equation Drilldown: Taking the Equations Apart
- State Space Concepts

- Understanding the Equations: Heuristic Introduction
- Equation Drilldown: Taking the Equations Apart
- State Space Concepts

- Recursive Predictor-Corrector Algorithms

- Recursive Predictor-Corrector Algorithms
- Running Averages

- Recursive Predictor-Corrector Algorithms
- Running Averages
- **Prediction (Extrapolation)**

- Recursive Predictor-Corrector Algorithms
- Running Averages
- **Prediction (Extrapolation)**
- Correction (Measurement Update)

- Recursive Predictor-Corrector Algorithms
- Running Averages
- **Prediction (Extrapolation)**
- Correction (Measurement Update)
- Gain Computation

- Recursive Predictor-Corrector Algorithms
- Running Averages
- **Prediction (Extrapolation)**
- Correction (Measurement Update)
- Gain Computation

Prediction (Extrapolation)

Prediction (Extrapolation)

- In the running average example, the dynamics were trivial, i.e., $x(k+1) = x(k)$

Prediction (Extrapolation)

- In the running average example, the dynamics were trivial, i.e., $x(k+1) = x(k)$
- This means that the state never really changes. It's always equal to its previous value

Prediction (Extrapolation)

- In the running average example, the dynamics were trivial, i.e., $x(k+1) = x(k)$
- This means that the state never really changes. It's always equal to its previous value
- We just don't know what the exact value is

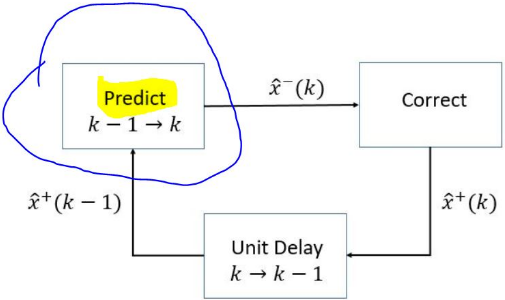
Prediction (Extrapolation)

- In the running average example, the dynamics were trivial, i.e., $x(k+1) = x(k)$
- This means that the state never really changes. It's always equal to its previous value
- We just don't know what the exact value is
- Example: Measuring the same resistor over and over again with a voltmeter to estimate the resistor's resistance value.

Prediction (Extrapolation)

- In the running average example, the dynamics were trivial, i.e., $x(k+1) = x(k)$
- This means that the state never really changes. It's always equal to its previous value
- We just don't know what the exact value is
- Example: Measuring the same resistor over and over again with a voltmeter to estimate the resistor's resistance value.

Prediction (Extrapolation)



Prediction (Extrapolation)

Prediction (Extrapolation)

- In more general Kalman filtering, the situation becomes more complicated by adding more elements:

- In more general Kalman filtering, the situation becomes more complicated by adding more elements:
  - Non-trivial state dynamics, possible time dependence and **process noise**:

$$x(k+1) = f\left[k, x(k), w(k)\right]$$

Prediction (Extrapolation)

- In more general Kalman filtering, the situation becomes more complicated by adding more elements:
  - Non-trivial state dynamics, possible time dependence and **process noise**:
    $$x(k+1) = f\left[k, x(k), w(k)\right]$$
  - Typically, the process noise is additive:
    $$x(k+1) = f\left[k, x(k)\right] + w(k)$$
    Or, equivalently,
    $$x(k) = f\left[k-1, x(k-1)\right] + w(k-1)$$

Prediction (Extrapolation)

- In more general Kalman filtering, the situation becomes more complicated by adding more elements:
  - Non-trivial state dynamics, possible time dependence and **process noise**:
  $$x(k+1) = f\left[k, x(k), w(k)\right]$$
  - Typically, the process noise is additive:
  $$x(k+1) = f\left[k, x(k)\right] + w(k)$$
  Or, equivalently,
  $$x(k) = f\left[k-1, x(k-1)\right] + w(k-1)$$
- Continuous-time systems are often specified as differential equations: $\dot{x}(t) = f(t, x(t), w(t))$,
  where the 'dot' over the x refers to the derivative of x with respect to time

Prediction (Extrapolation)

Prediction (Extrapolation)

- Form of function f can be simple or complicated, linear or nonlinear (but Kalman filtering requires linearization)

Prediction (Extrapolation)

- Form of function f can be simple or complicated, linear or nonlinear (but Kalman filtering requires linearization)
- Independent variable k indicates possible explicit time dependence

Prediction (Extrapolation)

- Form of function f can be simple or complicated, linear or nonlinear (but Kalman filtering requires linearization)
- Independent variable k indicates possible explicit time dependence
- Process noise assumed to be Gaussian white noise (to be defined later)

Prediction (Extrapolation)

- Form of function f can be simple or complicated, linear or nonlinear (but Kalman filtering requires linearization)
- Independent variable k indicates possible explicit time dependence
- Process noise assumed to be Gaussian white noise (to be defined later)

Prediction (Extrapolation)

Prediction (Extrapolation)

- General Kalman filtering further complicates things by allowing the state variable to be **vector-valued**

Prediction (Extrapolation)

- General Kalman filtering further complicates things by allowing the state variable to be **vector-valued**
- f thus becomes a vector-valued function, often a matrix (possibly time-dependent)

Prediction (Extrapolation)

- General Kalman filtering further complicates things by allowing the state variable to be **vector-valued**
- f thus becomes a vector-valued function, often a matrix (possibly time-dependent)
- Process noise must therefore also become a vector-valued variable
  (actually, a **random process**)

Prediction (Extrapolation)

- General Kalman filtering further complicates things by allowing the state variable to be **vector-valued**
- f thus becomes a vector-valued function, often a matrix (possibly time-dependent)
- Process noise must therefore also become a vector-valued variable
  (actually, a **random process**)

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$
  - $\dot{x}(t) = f(x(t), w(t)) = 0$, a constant function; if initial condition is a random variable then this is called a random constant

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$
  - $\dot{x}(t) = f(x(t), w(t)) = 0$, a constant function; if initial condition is a random variable then this is called a random constant
  - $\dot{x}(t) = f(x(t), w(t)) = x(t) + w(t)$, a continuous random walk

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$
  - $\dot{x}(t) = f(x(t), w(t)) = 0$, a constant function; if initial condition is a random variable then this is called a random constant
  - $\dot{x}(t) = f(x(t), w(t)) = x(t) + w(t)$, a continuous random walk
  - $\dot{x}(t) = f(x(t), w(t)) = -\beta x(t) + w(t)$, exponentially correlated random process

Prediction (Extrapolation)

Some Examples of *Scalar* Dynamical System Functions

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$
  - $\dot{x}(t) = f(x(t), w(t)) = 0$, a constant function; if initial condition is a random variable then this is called a random constant
  - $\dot{x}(t) = f(x(t), w(t)) = x(t) + w(t)$, a continuous random walk
  - $\dot{x}(t) = f(x(t), w(t)) = -\beta x(t) + w(t)$, exponentially correlated random process
- Discrete, Time-Invariant

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$
  - $\dot{x}(t) = f(x(t), w(t)) = 0$, a constant function; if initial condition is a random variable then this is called a random constant
  - $\dot{x}(t) = f(x(t), w(t)) = x(t) + w(t)$, a continuous random walk
  - $\dot{x}(t) = f(x(t), w(t)) = -\beta x(t) + w(t)$, exponentially correlated random process
- Discrete, Time-Invariant
  - $x(k+1) = f(x(k), w(k)) = x(k) + w(k)$, a discrete random walk sequence

Prediction (Extrapolation)

Some Examples of *Scalar* Dynamical System Functions

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$
  - $\dot{x}(t) = f(x(t), w(t)) = 0$, a constant function; if initial condition is a random variable then this is called a random constant
  - $\dot{x}(t) = f(x(t), w(t)) = x(t) + w(t)$, a continuous random walk
  - $\dot{x}(t) = f(x(t), w(t)) = -\beta x(t) + w(t)$, exponentially correlated random process
- Discrete, Time-Invariant
  - $x(k+1) = f(x(k), w(k)) = x(k) + w(k)$, a discrete random walk sequence
  - $x(k+1) = f(x(k), w(k) = x^2(k-1) \cdot w(k)$, discrete, nonlinear, multiplicative noise

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$
  - $\dot{x}(t) = f(x(t), w(t)) = 0$, a constant function; if initial condition is a random variable then this is called a random constant
  - $\dot{x}(t) = f(x(t), w(t)) = x(t) + w(t)$, a continuous random walk
  - $\dot{x}(t) = f(x(t), w(t)) = -\beta x(t) + w(t)$, exponentially correlated random process
- Discrete, Time-Invariant
  - $x(k+1) = f(x(k), w(k)) = x(k) + w(k)$, a discrete random walk sequence
  - $x(k+1) = f(x(k), w(k) = x^2(k-1) \cdot w(k)$, discrete, nonlinear, multiplicative noise
  - $x(k+1) = f(x(k), w(k)) = ax(k) + bx(k-1) + w(k)$, autoregressive (AR) difference equation plus noise input

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$
  - $\dot{x}(t) = f(x(t), w(t)) = 0$, a constant function; if initial condition is a random variable then this is called a random constant
  - $\dot{x}(t) = f(x(t), w(t)) = x(t) + w(t)$, a continuous random walk
  - $\dot{x}(t) = f(x(t), w(t)) = -\beta x(t) + w(t)$, exponentially correlated random process
- Discrete, Time-Invariant
  - $x(k+1) = f(x(k), w(k)) = x(k) + w(k)$, a discrete random walk sequence
  - $x(k+1) = f(x(k), w(k) = x^2(k-1) \cdot w(k)$, discrete, nonlinear, multiplicative noise
  - $x(k+1) = f(x(k), w(k)) = ax(k) + bx(k-1) + w(k)$, autoregressive (AR) difference equation plus noise input

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = f(\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}) = \begin{bmatrix} 0 & 1.0 \\ -2.0 & 1.5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$$

a constant-coefficient differential equation driven by noise

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = f(\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}) = \begin{bmatrix} 0 & 1.0 \\ -2.0 & 1.5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$$

a constant-coefficient differential equation driven by noise

- Discrete, but Time-varying

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = f(\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}) = \begin{bmatrix} 0 & 1.0 \\ -2.0 & 1.5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$$

a constant-coefficient differential equation driven by noise

- Discrete, but Time-varying

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = f(\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}) = \begin{bmatrix} 0 & \sin(\omega k) \\ \cos(\omega k) & k^2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

a **difference** equation with time-dependent coefficients and driven by noise

- Continuous, Time-Invariant and Differentiable over some bounded time interval $[0, T]$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = f\left(\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}\right) = \begin{bmatrix} 0 & 1.0 \\ -2.0 & 1.5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$$

a constant-coefficient differential equation driven by noise

- Discrete, but Time-varying

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = f\left(\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}\right) = \begin{bmatrix} 0 & \sin(\omega k) \\ \cos(\omega k) & k^2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

a **difference** equation with time-dependent coefficients and driven by noise

- Discrete, Time-Invariant: A car moving in one direction along a road at constant speed (under cruise control)

- Discrete, Time-Invariant: A car moving in one direction along a road at constant speed (under cruise control)

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = f\left(\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}\right) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

- Discrete, Time-Invariant: A car moving in one direction along a road at constant speed (under cruise control)

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = f(\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

  - where $\Delta T := t_{k+1} - t_k$ is the time difference between each epoch (assumed constant)

- Discrete, Time-Invariant: A car moving in one direction along a road at constant speed (under cruise control)

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = f(\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

  - where $\Delta T := t_{k+1} - t_k$ is the time difference between each epoch (assumed constant)
  - The first state x_1 is the **position** of the car

- Discrete, Time-Invariant: A car moving in one direction along a road at constant speed (under cruise control)

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = f\left(\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} \right) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

  - where $\Delta T := t_{k+1} - t_k$ is the time difference between each epoch (assumed constant)
  - The first state x_1 is the **position** of the car
  - The second state x_2 is the (constant) **velocity**

- Discrete, Time-Invariant: A car moving in one direction along a road at constant speed (under cruise control)

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = f(\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

  - where $\Delta T := t_{k+1} - t_k$ is the time difference between each epoch (assumed constant)
  - The first state x_1 is the **position** of the car
  - The second state x_2 is the (constant) **velocity**

Prediction (Extrapolation)

Some **Physical** Examples of Vector Dynamical System Functions

- Note that the position is primarily a function of the velocity $x_2(k)$ and the time increment ΔT

- Note that the position is primarily a function of the velocity $x_2(k)$ and the time increment ΔT
- Note also that there are noise terms associated with both the position and velocity states:

$$x_1(k+1) = x_1(k) + x_2(k) \cdot \Delta T + w_1(k)$$

and

$$x_2(k+1) = x_2(k) + w_2(k)$$

- Note that the position is primarily a function of the velocity $x_2(k)$ and the time increment ΔT
- Note also that there are noise terms associated with both the position and velocity states:

$$x_1(k+1) = x_1(k) + x_2(k) \cdot \Delta T + w_1(k)$$

and

$$x_2(k+1) = x_2(k) + w_2(k)$$

- w_1 is noise due perhaps to an inaccurate clock (whose dynamics we are not explicitly modeling)

- Note that the position is primarily a function of the velocity $x_2(k)$ and the time increment ΔT
- Note also that there are noise terms associated with both the position and velocity states:

$$x_1(k+1) = x_1(k) + x_2(k) \cdot \Delta T + w_1(k)$$

and

$$x_2(k+1) = x_2(k) + w_2(k)$$

- w_1 is noise due perhaps to an inaccurate clock (whose dynamics we are not explicitly modeling)
- w_2 is noise due to unmodeled, short term accelerations that the cruise contral is trying to overcome

- Note that the position is primarily a function of the velocity $x_2(k)$ and the time increment ΔT
- Note also that there are noise terms associated with both the position and velocity states:

$$x_1(k+1) = x_1(k) + x_2(k) \cdot \Delta T + w_1(k)$$

and

$$x_2(k+1) = x_2(k) + w_2(k)$$

- w_1 is noise due perhaps to an inaccurate clock (whose dynamics we are not explicitly modeling)
- w_2 is noise due to unmodeled, short term accelerations that the cruise contral is trying to overcome

- Note that presence of a random forcing function like Gaussian white noise makes the state vector process a stochastic process (aka random process)

- Note that presence of a random forcing function like Gaussian white noise makes the state vector process a stochastic process (aka random process)
  - Technically, this means we can't use ordinary calculus to solve the stochastic differential equations

- Note that presence of a random forcing function like Gaussian white noise makes the state vector process a stochastic process (aka random process)
    - Technically, this means we can't use ordinary calculus to solve the stochastic differential equations
- Even with the same initial conditions the sample trajectory will be different each time, due to randomly changing forcing functions

- Note that presence of a random forcing function like Gaussian white noise makes the state vector process a stochastic process (aka random process)
  - Technically, this means we can't use ordinary calculus to solve the stochastic differential equations
- Even with the same initial conditions the sample trajectory will be different each time, due to randomly changing forcing functions
- But many sample trajectories can be simulated/generated on a computer using Monte Carlo methods

- Note that presence of a random forcing function like Gaussian white noise makes the state vector process a stochastic process (aka random process)
  - Technically, this means we can't use ordinary calculus to solve the stochastic differential equations
- Even with the same initial conditions the sample trajectory will be different each time, due to randomly changing forcing functions
- But many sample trajectories can be simulated/generated on a computer using Monte Carlo methods
- We'll look at this more closely when we discuss random variables and random/stochastic processes

- Note that presence of a random forcing function like Gaussian white noise makes the state vector process a stochastic process (aka random process)
  - Technically, this means we can't use ordinary calculus to solve the stochastic differential equations
- Even with the same initial conditions the sample trajectory will be different each time, due to randomly changing forcing functions
- But many sample trajectories can be simulated/generated on a computer using Monte Carlo methods
- We'll look at this more closely when we discuss random variables and random/stochastic processes

- Note that we did not mention a **control function** $u(t)$ input that could also be part of the forcing function

- Note that we did not mention a **control function** $u(t)$ input that could also be part of the forcing function
  - Control functions are usually considered to be deterministic (although in reality you can't control things perfectly)

- Note that we did not mention a **control function** $u(t)$ input that could also be part of the forcing function
    - Control functions are usually considered to be deterministic (although in reality you can't control things perfectly)
        - E.g., aircraft actuator (e.g., elevator or flaps) deflection angle might be a control variable and yet its true value may not be known precisely!

- Note that we did not mention a **control function** $u(t)$ input that could also be part of the forcing function
    - Control functions are usually considered to be deterministic (although in reality you can't control things perfectly)
        - E.g., aircraft actuator (e.g., elevator or flaps) deflection angle might be a control variable and yet its true value may not be known precisely!
- To keep things simple and to focus on state estimation, we are going to ignore the presence of deterministic control

- Note that we did not mention a **control function** $u(t)$ input that could also be part of the forcing function
  - Control functions are usually considered to be deterministic (although in reality you can't control things perfectly)
    - E.g., aircraft actuator (e.g., elevator or flaps) deflection angle might be a control variable and yet its true value may not be known precisely!
- To keep things simple and to focus on state estimation, we are going to ignore the presence of deterministic control
- Thus, the only forcing functions considered will be uncontrollable disturbance functions like noise

- Note that we did not mention a **control function** $u(t)$ input that could also be part of the forcing function
  - Control functions are usually considered to be deterministic (although in reality you can't control things perfectly)
    - E.g., aircraft actuator (e.g., elevator or flaps) deflection angle might be a control variable and yet its true value may not be known precisely!
- To keep things simple and to focus on state estimation, we are going to ignore the presence of deterministic control
- Thus, the only forcing functions considered will be uncontrollable disturbance functions like noise
- And that is precisely what we want to filter out: Noise

- Note that we did not mention a **control function** $u(t)$ input that could also be part of the forcing function
  - Control functions are usually considered to be deterministic (although in reality you can't control things perfectly)
    - E.g., aircraft actuator (e.g., elevator or flaps) deflection angle might be a control variable and yet its true value may not be known precisely!
- To keep things simple and to focus on state estimation, we are going to ignore the presence of deterministic control
- Thus, the only forcing functions considered will be uncontrollable disturbance functions like noise
- And that is precisely what we want to filter out: Noise

- So, what is noise?

- So, what is noise?
- Key characteristic: Uncorrelated in time

- So, what is noise?
- Key characteristic: Uncorrelated in time
    - E.g., $w(k)$ and $w(k+1)$ are statistically uncorrelated

- So, what is noise?
- Key characteristic: Uncorrelated in time
  - E.g., $w(k)$ and $w(k+1)$ are statistically uncorrelated
  - As a function of time, it can only be characterized **statistically**

- So, what is noise?
- Key characteristic: Uncorrelated in time
  - E.g., $w(k)$ and $w(k+1)$ are statistically uncorrelated
  - As a function of time, it can only be characterized **statistically**
  - Process noise is distinct from measurement noise and may result from real physical disturbances or from modeling errors (just don't know all the underlying perturbations)

- So, what is noise?
- Key characteristic: Uncorrelated in time
  - E.g., $w(k)$ and $w(k+1)$ are statistically uncorrelated
  - As a function of time, it can only be characterized **statistically**
  - Process noise is distinct from measurement noise and may result from real physical disturbances or from modeling errors (just don't know all the underlying perturbations)
- More to come on this topic after we've drilled down more on probability and statistics

- So, what is noise?
- Key characteristic: Uncorrelated in time
  - E.g., $w(k)$ and $w(k+1)$ are statistically uncorrelated
  - As a function of time, it can only be characterized **statistically**
  - Process noise is distinct from measurement noise and may result from real physical disturbances or from modeling errors (just don't know all the underlying perturbations)
- More to come on this topic after we've drilled down more on probability and statistics

- When we apply state dynamics to the previous time step to bring the previous state estimate forward to the present time, we ignore the process noise

- When we apply state dynamics to the previous time step to bring the previous state estimate forward to the present time, we ignore the process noise
- That's because our estimate is a sample mean (average) and we expect the noise – on average – to be zero

- When we apply state dynamics to the previous time step to bring the previous state estimate forward to the present time, we ignore the process noise
- That's because our estimate is a sample mean (average) and we expect the noise – on average – to be zero
- Therefore, $\hat{x}^{-}(k) = f(k-1, \hat{x}^{+}(k-1), 0)$

- When we apply state dynamics to the previous time step to bring the previous state estimate forward to the present time, we ignore the process noise
- That's because our estimate is a sample mean (average) and we expect the noise – on average – to be zero
- Therefore, $\hat{x}^-(k) = f(k-1, \hat{x}^+(k-1), 0)$
  - Makes implicit use of the fact that process noise is additive and zero-mean

- When we apply state dynamics to the previous time step to bring the previous state estimate forward to the present time, we ignore the process noise
- That's because our estimate is a sample mean (average) and we expect the noise – on average – to be zero
- Therefore, $\hat{x}^-(k) = f(k-1, \hat{x}^+(k-1), 0)$
  - Makes implicit use of the fact that process noise is additive and zero-mean
- The quantity $\hat{x}^-(k)$ is basis for the **predicted measurement** which results when we apply the measurement function to the predicted state (to be discussed in next video)

- When we apply state dynamics to the previous time step to bring the previous state estimate forward to the present time, we ignore the process noise
- That's because our estimate is a sample mean (average) and we expect the noise – on average – to be zero
- Therefore, $\hat{x}^{-}(k) = f(k-1, \hat{x}^{+}(k-1), 0)$
  - Makes implicit use of the fact that process noise is additive and zero-mean
- The quantity $\hat{x}^{-}(k)$ is basis for the **predicted measurement** which results when we apply the measurement function to the predicted state (to be discussed in next video)

- Recall that this prediction step was missing from the running average problem because the f function in this case was a simple identity: $\hat{x}^{-}(k) = \hat{x}^{+}(k-1)$

- Recall that this prediction step was missing from the running average problem because the f function in this case was a simple identity: $\hat{x}^{-}(k) = \hat{x}^{+}(k-1)$
- Recall also that the measurement function in the running average scenario was likewise the identify function

- Recall that this prediction step was missing from the running average problem because the f function in this case was a simple identity: $\hat{x}^{-}(k) = \hat{x}^{+}(k-1)$
- Recall also that the measurement function in the running average scenario was likewise the identify function
- Although process noise plays no role in predicting the new state, the process noise **statistics** play a very important role in predicting how the estimation uncertainty (or estimation error covariance) grows with time

Prediction (Extrapolation)

State Dynamics and Predicted Measurement

- Recall that this prediction step was missing from the running average problem because the f function in this case was a simple identity: $\hat{x}^{-}(k) = \hat{x}^{+}(k-1)$
- Recall also that the measurement function in the running average scenario was likewise the identify function
- Although process noise plays no role in predicting the new state, the process noise **statistics** play a very important role in predicting how the estimation uncertainty (or estimation error covariance) grows with time
- Process noise is also important and is used directly in simulating system behavior over time

- Recall that this prediction step was missing from the running average problem because the f function in this case was a simple identity: $\hat{x}^{-}(k) = \hat{x}^{+}(k-1)$
- Recall also that the measurement function in the running average scenario was likewise the identify function
- Although process noise plays no role in predicting the new state, the process noise **statistics** play a very important role in predicting how the estimation uncertainty (or estimation error covariance) grows with time
- Process noise is also important and is used directly in simulating system behavior over time
- We'll discuss this more when we drill down on the five Kalman equations

- Recall that this prediction step was missing from the running average problem because the f function in this case was a simple identity: $\hat{x}^{-}(k) = \hat{x}^{+}(k-1)$
- Recall also that the measurement function in the running average scenario was likewise the identify function
- Although process noise plays no role in predicting the new state, the process noise **statistics** play a very important role in predicting how the estimation uncertainty (or estimation error covariance) grows with time
- Process noise is also important and is used directly in simulating system behavior over time
- We'll discuss this more when we drill down on the five Kalman equations

- Where are we?

- Where are we?
  - Looked at some mathematical models of dynamic systems, i.e., models of how systems change state with time

- Where are we?
  - Looked at some mathematical models of dynamic systems, i.e., models of how systems change state with time
  - Considered concept of **process noise** as a random forcing function

- Where are we?
  - Looked at some mathematical models of dynamic systems, i.e., models of how systems change state with time
  - Considered concept of **process noise** as a random forcing function
  - Discussed how the state estimate is propagated in time and why the process noise is ignored
- What's next?

- Where are we?
  - Looked at some mathematical models of dynamic systems, i.e., models of how systems change state with time
  - Considered concept of **process noise** as a random forcing function
  - Discussed how the state estimate is propagated in time and why the process noise is ignored
- What's next?
  - In the next video we will consider the **Correction (measurement update)** step in more detail

- Where are we?
  - Looked at some mathematical models of dynamic systems, i.e., models of how systems change state with time
  - Considered concept of **process noise** as a random forcing function
  - Discussed how the state estimate is propagated in time and why the process noise is ignored
- What's next?
  - In the next video we will consider the **Correction (measurement update)** step in more detail
  - This will prepare us for a more detailed discussion of the Kalman gain

Prediction (Extrapolation) Summary

Vector Check

- Where are we?
  - Looked at some mathematical models of dynamic systems, i.e., models of how systems change state with time
  - Considered concept of **process noise** as a random forcing function
  - Discussed how the state estimate is propagated in time and why the process noise is ignored
- What's next?
  - In the next video we will consider the **Correction (measurement update)** step in more detail
  - This will prepare us for a more detailed discussion of the Kalman gain