Kalman Filter Theory and Applications Equation Drilldown

Michael L. Carroll

November 27, 2020

Part I
The Basic Kalman Equations
Topics

- 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

- **Mathematical Formulation of the Problem**

- **Mathematical Formulation of the Problem**
- Drilldown on State Dynamics and Covariance Extrapolation Equations

- **Mathematical Formulation of the Problem**
- Drilldown on State Dynamics and Covariance Extrapolation Equations
- The Five Kalman Filter Equations

- **Mathematical Formulation of the Problem**
- Drilldown on State Dynamics and Covariance Extrapolation Equations
- The Five Kalman Filter Equations
- Examples

- **Mathematical Formulation of the Problem**
- Drilldown on State Dynamics and Covariance Extrapolation Equations
- The Five Kalman Filter Equations
- Examples
- Exercises

- **Mathematical Formulation of the Problem**
- Drilldown on State Dynamics and Covariance Extrapolation Equations
- The Five Kalman Filter Equations
- Examples
- Exercises

- Given two models:

- Given two models:
  - State Dynamics Model: Produces predictions

- Given two models:
    - State Dynamics Model: Produces predictions
    - Measurement Model: Translates predictions into predicted measurements

- Given two models:
  - State Dynamics Model: Produces predictions
  - Measurement Model: Translates predictions into predicted measurements
- Calculate:

- Given two models:
  - State Dynamics Model: Produces predictions
  - Measurement Model: Translates predictions into predicted measurements
- Calculate:
  - Residual, subtract the predicted measurements from the actual measurements

- Given two models:
  - State Dynamics Model: Produces predictions
  - Measurement Model: Translates predictions into predicted measurements
- Calculate:
  - Residual, subtract the predicted measurements from the actual measurements
  - Optimal **linear** gain matrix $K(k)$ for weighting measurement residual and update state estimate *recursively and incrementally* by adding gain-weighted residual

- Given two models:
  - State Dynamics Model: Produces predictions
  - Measurement Model: Translates predictions into predicted measurements
- Calculate:
  - Residual, subtract the predicted measurements from the actual measurements
  - Optimal **linear** gain matrix $K(k)$ for weighting measurement residual and update state estimate *recursively and incrementally* by adding gain-weighted residual

---

Given:

Given:

- State Dynamics Model: $x(k) = f[x(k-1)] + w(k)$

Given:

- State Dynamics Model: $x(k) = f[x(k-1)] + w(k)$
  - where x=state vector, f=vector-valued dynamics function, w=process noise vector

Given:

- State Dynamics Model: $x(k) = f[x(k-1)] + w(k)$
  - where x=state vector, f=vector-valued dynamics function, w=process noise vector
- Measurement Model: $z(k) = h[x(k)] + v(k)$

Given:

- State Dynamics Model: $x(k) = f[x(k-1)] + w(k)$
  - where x=state vector, f=vector-valued dynamics function, w=process noise vector
- Measurement Model: $z(k) = h[x(k)] + v(k)$
  - where z=measurement vector, h=vector-valued measurement function, v=measurement noise

Given:

- State Dynamics Model: $x(k) = f[x(k-1)] + w(k)$
  - where x=state vector, f=vector-valued dynamics function, w=process noise vector
- Measurement Model: $z(k) = h[x(k)] + v(k)$
  - where z=measurement vector, h=vector-valued measurement function, v=measurement noise

Calculate:

Given:

- State Dynamics Model: $x(k) = f[x(k-1)] + w(k)$
  - where x=state vector, f=vector-valued dynamics function, w=process noise vector
- Measurement Model: $z(k) = h[x(k)] + v(k)$
  - where z=measurement vector, h=vector-valued measurement function, v=measurement noise

Calculate:

- Optimal gain $K(k)$ in order to update state estimate:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[\, z(k) - h(\hat{x}^{-}(k))\,\right]$$

Given:

- State Dynamics Model: $x(k) = f[x(k-1)] + w(k)$
  - where x=state vector, f=vector-valued dynamics function, w=process noise vector
- Measurement Model: $z(k) = h[x(k)] + v(k)$
  - where z=measurement vector, h=vector-valued measurement function, v=measurement noise

Calculate:

- Optimal gain $K(k)$ in order to update state estimate:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\,[\,z(k) - h(\hat{x}^{-}(k))\,]$$

  - where $\hat{x}^{-}(k) = f[\hat{x}^{+}(k-1)]$=extrapolated/predicted estimate, K=gain matrix, $h(\hat{x}^{-}(k)])$=expected/predicted measurement

Given:

- State Dynamics Model: $x(k) = f[x(k-1)] + w(k)$
  - where x=state vector, f=vector-valued dynamics function, w=process noise vector
- Measurement Model: $z(k) = h[x(k)] + v(k)$
  - where z=measurement vector, h=vector-valued measurement function, v=measurement noise

Calculate:

- Optimal gain $K(k)$ in order to update state estimate:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[z(k) - h(\hat{x}^{-}(k))\right]$$

  - where $\hat{x}^{-}(k) = f[\hat{x}^{+}(k-1)]$=extrapolated/predicted estimate, K=gain matrix, $h(\hat{x}^{-}(k)])$=expected/predicted measurement

- Kalman filter originally derived for linear system dynamics and linear measurement model

- Kalman filter originally derived for linear system dynamics and linear measurement model
  - Assume vector-valued f realized as multiplication by time-invariant matrix Φ called **state transition matrix**

- Kalman filter originally derived for linear system dynamics and linear measurement model
  - Assume vector-valued f realized as multiplication by time-invariant matrix Φ called **state transition matrix**
  - Assume h realized as multiplication by matrix $H(k)$

- Kalman filter originally derived for linear system dynamics and linear measurement model
  - Assume vector-valued f realized as multiplication by time-invariant matrix Φ called **state transition matrix**
  - Assume h realized as multiplication by matrix $H(k)$
- Find: Gain multiplier matrix $K(k)$

- Kalman filter originally derived for linear system dynamics and linear measurement model
  - Assume vector-valued f realized as multiplication by time-invariant matrix Φ called **state transition matrix**
  - Assume h realized as multiplication by matrix $H(k)$
- Find: Gain multiplier matrix $K(k)$

Given:

Given:

- Discrete-Time State Dynamics Model:

$$x(k) = \Phi x(k-1) + w(k)$$

Given:

- Discrete-Time State Dynamics Model:

$$x(k) = \Phi x(k-1) + w(k)$$

  - x=state vector, Φ=state transition matrix, w=process noise vector

Given:

- Discrete-Time State Dynamics Model:

$$x(k) = \Phi x(k-1) + w(k)$$

  - x=state vector, Φ=state transition matrix, w=process noise vector
- Measurement Model: $z(k) = H\,x(k) + v(k)$
  - where z=measurement vector, H=measurement matrix (assumed constant), v=measurement noise vector

Given:

- Discrete-Time State Dynamics Model:

$$x(k) = \Phi x(k-1) + w(k)$$

  - x=state vector, Φ=state transition matrix, w=process noise vector
- Measurement Model: $z(k) = H\,x(k) + v(k)$
  - where z=measurement vector, H=measurement matrix (assumed constant), v=measurement noise vector

Calculate:

Given:

- Discrete-Time State Dynamics Model:
$$x(k) = \Phi x(k-1) + w(k)$$
  - x=state vector, Φ=state transition matrix, w=process noise vector
- Measurement Model: $z(k) = H\, x(k) + v(k)$
  - where z=measurement vector, H=measurement matrix (assumed constant), v=measurement noise vector

Calculate:

- An optimal gain matrix $K(k)$ and new state estimate:
$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[\, z(k) - H\hat{x}^{-}(k)\,\right]$$

Given:

- Discrete-Time State Dynamics Model:

$$x(k) = \Phi x(k-1) + w(k)$$

  - x=state vector, Φ=state transition matrix, w=process noise vector
- Measurement Model: $z(k) = H\,x(k) + v(k)$
  - where z=measurement vector, H=measurement matrix (assumed constant), v=measurement noise vector

Calculate:

- An optimal gain matrix $K(k)$ and new state estimate:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[z(k) - H\hat{x}^{-}(k)\right]$$

  - where $\hat{x}^{-}(k) = \Phi\hat{x}^{+}(k-1)$ extrapolates previous estimate, and $H\hat{x}^{-}(k)$ is the expected/predicted measurement

Given:

- Discrete-Time State Dynamics Model:

$$x(k) = \Phi x(k-1) + w(k)$$

  - x=state vector, Φ=state transition matrix, w=process noise vector
- Measurement Model: $z(k) = H\,x(k) + v(k)$
  - where z=measurement vector, H=measurement matrix (assumed constant), v=measurement noise vector

Calculate:

- An optimal gain matrix $K(k)$ and new state estimate:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[\,z(k) - H\hat{x}^{-}(k)\,\right]$$

  - where $\hat{x}^{-}(k) = \Phi\hat{x}^{+}(k-1)$ extrapolates previous estimate, and $H\hat{x}^{-}(k)$ is the expected/predicted measurement

- Reminder: To distinguish prior (extrapolated) estimate from updated estimate, we use $+/$- hat notation:

- Reminder: To distinguish prior (extrapolated) estimate from updated estimate, we use $+/$- hat notation:
- $\hat{x}^{-}(k) := \Phi \hat{x}^{+}(k-1)$ is predicted state estimate

- Reminder: To distinguish prior (extrapolated) estimate from updated estimate, we use $+/$- hat notation:
- $\hat{x}^{-}(k) := \Phi\hat{x}^{+}(k-1)$ is predicted state estimate
- $\hat{x}^{+}(k) =$ updated estimate

- Reminder: To distinguish prior (extrapolated) estimate from updated estimate, we use $+/$- hat notation:
- $\hat{x}^{-}(k) := \Phi\hat{x}^{+}(k-1)$ is predicted state estimate
- $\hat{x}^{+}(k) =$ updated estimate
- $\hat{z}(k) := H(k)\hat{x}^{-}(k)$ sometimes used to denote expected/predicted measurment

- Reminder: To distinguish prior (extrapolated) estimate from updated estimate, we use $+/-$ hat notation:
- $\hat{x}^-(k) := \Phi\hat{x}^+(k-1)$ is predicted state estimate
- $\hat{x}^+(k) =$ updated estimate
- $\hat{z}(k) := H(k)\hat{x}^-(k)$ sometimes used to denote expected/predicted measurment

- Solution should look like this:

- Solution should look like this:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[\, z(k) - H\hat{x}^{-}(k) \,\right] \tag{1}$$

- Solution should look like this:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[\, z(k) - H\hat{x}^{-}(k) \,\right] \tag{1}$$

where $\hat{x}^{-}(k) = \Phi\hat{x}^{+}(k-1)$

- Solution should look like this:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[\, z(k) - H\hat{x}^{-}(k) \,\right] \tag{1}$$

where $\hat{x}^{-}(k) = \Phi\hat{x}^{+}(k-1)$

- Looks like running average!

- Solution should look like this:

$$\hat{x}^{+}(k) = \hat{x}^{-}(k) + K(k)\left[\, z(k) - H\hat{x}^{-}(k) \,\right] \qquad (1)$$

where $\hat{x}^{-}(k) = \Phi\hat{x}^{+}(k-1)$

- Looks like running average!

- But where are process noise $w(k)$ and measurement noise $v(k)$ used?

- But where are process noise $w(k)$ and measurement noise $v(k)$ used?
- Statistics (covariances) associated with noise processes are inputs to Kalman gain computation

- But where are process noise $w(k)$ and measurement noise $v(k)$ used?
- Statistics (covariances) associated with noise processes are inputs to Kalman gain computation
- Kalman gain will also be used to update **estimation uncertainty**, known as the **estimation error covariance**

- But where are process noise $w(k)$ and measurement noise $v(k)$ used?
- Statistics (covariances) associated with noise processes are inputs to Kalman gain computation
- Kalman gain will also be used to update **estimation uncertainty**, known as the **estimation error covariance**
- Φ be used to predict change in estimation uncertainty over time

---

- Where are we?

- Where are we?
  - We've formulated the linear discrete problem

- Where are we?
  - We've formulated the linear discrete problem
  - We've indicated the form of the solution: the recursive estimating algorithm with a matrix-valued Kalman gain

- Where are we?
  - We've formulated the linear discrete problem
  - We've indicated the form of the solution: the recursive estimating algorithm with a matrix-valued Kalman gain
- What's next?

- Where are we?
  - We've formulated the linear discrete problem
  - We've indicated the form of the solution: the recursive estimating algorithm with a matrix-valued Kalman gain
- What's next?
  - Drill down on state dynamics and determine how noise/uncertainty statistics change over time

- Where are we?
  - We've formulated the linear discrete problem
  - We've indicated the form of the solution: the recursive estimating algorithm with a matrix-valued Kalman gain
- What's next?
  - Drill down on state dynamics and determine how noise/uncertainty statistics change over time
  - Use both estimation uncertainty and measurment noise uncertainty to calculate the Kalman gain

- Where are we?
  - We've formulated the linear discrete problem
  - We've indicated the form of the solution: the recursive estimating algorithm with a matrix-valued Kalman gain
- What's next?
  - Drill down on state dynamics and determine how noise/uncertainty statistics change over time
  - Use both estimation uncertainty and measurment noise uncertainty to calculate the Kalman gain
  - Use Kalman gain to update estimation uncertainty

- Where are we?
  - We've formulated the linear discrete problem
  - We've indicated the form of the solution: the recursive estimating algorithm with a matrix-valued Kalman gain
- What's next?
  - Drill down on state dynamics and determine how noise/uncertainty statistics change over time
  - Use both estimation uncertainty and measurment noise uncertainty to calculate the Kalman gain
  - Use Kalman gain to update estimation uncertainty