Kalman Filter Theory and Applications Equation Drilldown

https://github.com/musicarroll/kalman_course

Michael L. Carroll

June 22, 2023

- 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

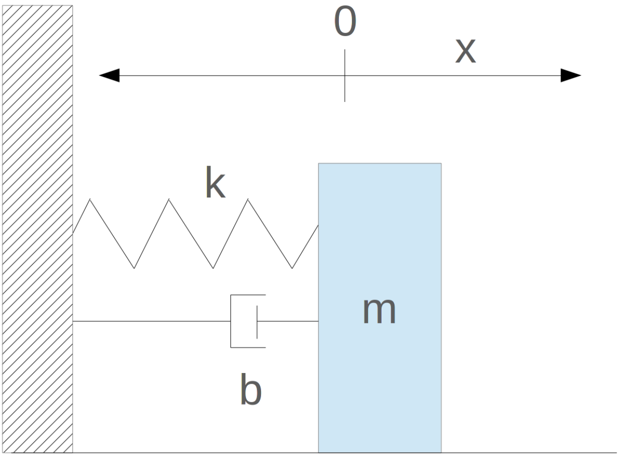
- Mathematical Formulation of the Problem
- Drilldown on State Dynamics and Covariance Extrapolation Equations
- The Five Kalman Filter Equations
- **Examples: Damped Harmonic Oscillator**

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

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

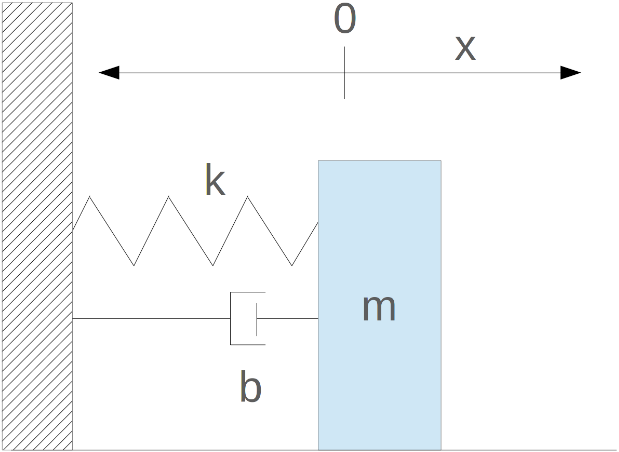
Damped Harmonic Oscillator

Spring, Mass, Damper System



Damped Harmonic Oscillator

Spring, Mass, Damper System



- System model: Number of states $= 2$ (position and velocity)

- System model: Number of states $= 2$ (position and velocity)
  Derived from differential equation model: $m\ddot{x} + b\dot{x} + kx = 0$

- System model: Number of states $= 2$ (position and velocity)
  Derived from differential equation model: $m\ddot{x} + b\dot{x} + kx = 0$

$$\begin{bmatrix} x_1(j) \\ x_2(j) \end{bmatrix} = \exp\left(\begin{bmatrix} 0 & 1 \\ -b/m & -k/m \end{bmatrix} \Delta t \right) \begin{bmatrix} x_1(j-1) \\ x_2(j-1) \end{bmatrix} + \begin{bmatrix} w_1(j) \\ w_2(j) \end{bmatrix}$$

- Thus, $\Phi(\Delta t) = \exp\left(\begin{bmatrix} 0 & 1 \\ -b/m & -k/m \end{bmatrix} \Delta t \right)$.

- System model: Number of states $= 2$ (position and velocity)
  Derived from differential equation model: $m\ddot{x} + b\dot{x} + kx = 0$

$$\begin{bmatrix} x_1(j) \\ x_2(j) \end{bmatrix} = \exp\left(\begin{bmatrix} 0 & 1 \\ -b/m & -k/m \end{bmatrix} \Delta t\right) \begin{bmatrix} x_1(j-1) \\ x_2(j-1) \end{bmatrix} + \begin{bmatrix} w_1(j) \\ w_2(j) \end{bmatrix}$$

- Thus, $\Phi(\Delta t) = \exp\left(\begin{bmatrix} 0 & 1 \\ -b/m & -k/m \end{bmatrix} \Delta t\right)$.
- Note that exp is a matrix exponential. We'll explain later how to derive state transition matrices from the underlying differential equations

- System model: Number of states = 2 (position and velocity)
  Derived from differential equation model: $m\ddot{x} + b\dot{x} + kx = 0$

$$\begin{bmatrix} x_1(j) \\ x_2(j) \end{bmatrix} = \exp\left(\begin{bmatrix} 0 & 1 \\ -b/m & -k/m \end{bmatrix} \Delta t \right) \begin{bmatrix} x_1(j-1) \\ x_2(j-1) \end{bmatrix} + \begin{bmatrix} w_1(j) \\ w_2(j) \end{bmatrix}$$

- Thus, $\Phi(\Delta t) = \exp\left(\begin{bmatrix} 0 & 1 \\ -b/m & -k/m \end{bmatrix} \Delta t \right)$.
- Note that exp is a matrix exponential. We'll explain later how to derive state transition matrices from the underlying differential equations
- Here $\Delta t = t_j - t_{j-1}$, m is the mass, k is the spring constant, and b is the damping constant; we are using t for the time step index, to avoid clash with spring constant

- System model: Number of states $= 2$ (position and velocity) Derived from differential equation model: $m\ddot{x} + b\dot{x} + kx = 0$

$$\begin{bmatrix} x_1(j) \\ x_2(j) \end{bmatrix} = \exp\left(\begin{bmatrix} 0 & 1 \\ -b/m & -k/m \end{bmatrix} \Delta t \right) \begin{bmatrix} x_1(j-1) \\ x_2(j-1) \end{bmatrix} + \begin{bmatrix} w_1(j) \\ w_2(j) \end{bmatrix}$$

- Thus, $\Phi(\Delta t) = \exp\left(\begin{bmatrix} 0 & 1 \\ -b/m & -k/m \end{bmatrix} \Delta t \right)$.
- Note that exp is a matrix exponential. We'll explain later how to derive state transition matrices from the underlying differential equations
- Here $\Delta t = t_j - t_{j-1}$, m is the mass, k is the spring constant, and b is the damping constant; we are using t for the time step index, to avoid clash with spring constant

- Measurement Model:

$$z(j) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(j) \\ x_2(j) \end{bmatrix} + v(j)$$

where the measurement noise $v(j)$ has variance σ^2

- Measurement Model:

$$z(j) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(j) \\ x_2(j) \end{bmatrix} + v(j)$$

  where the measurement noise $v(j)$ has variance σ^2
- Thus $H_t = \begin{bmatrix} 1 & 0 \end{bmatrix}$

- Measurement Model:

$$z(j) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(j) \\ x_2(j) \end{bmatrix} + v(j)$$

  where the measurement noise $v(j)$ has variance σ^2
- Thus $H_t = \begin{bmatrix} 1 & 0 \end{bmatrix}$
- Note that we only have a position measurement (similar to constant velocity example).

- Measurement Model:

$$z(j) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(j) \\ x_2(j) \end{bmatrix} + v(j)$$

  where the measurement noise $v(j)$ has variance σ^2
- Thus $H_t = \begin{bmatrix} 1 & 0 \end{bmatrix}$
- Note that we only have a position measurement (similar to constant velocity example).

- $\Phi(\Delta t)$ can be approximated through series expansion of

$$\exp \begin{bmatrix} 0 & \Delta t \\ -\frac{b}{m}\Delta t & -\frac{k}{m}\Delta t \end{bmatrix}$$

- $\Phi(\Delta t)$ can be approximated through series expansion of

$$\exp\begin{bmatrix} 0 & \Delta t \\ -\frac{b}{m}\Delta t & -\frac{k}{m}\Delta t \end{bmatrix}$$

- Or we can use Matlab's (or scipy's) expm function for specific values of k,b,m, and Δt.

- $\Phi(\Delta t)$ can be approximated through series expansion of

$$\exp \begin{bmatrix} 0 & \Delta t \\ -\frac{b}{m}\Delta t & -\frac{k}{m}\Delta t \end{bmatrix}$$

- Or we can use Matlab's (or scipy's) expm function for specific values of k, b, m, and Δt.

- Because Φ and Q are matrices, the state and covariance extrapolation equations do not simplify from their general matrix form

- Because Φ and Q are matrices, the state and covariance extrapolation equations do not simplify from their general matrix form
- Moreover, like Φ, which is derived from a continuous model, Q is also dependent on Δt and q, the continuous process noise variance:

$$Q = q \begin{bmatrix} \frac{(\Delta t)^3}{3} & \frac{(\Delta t)^2}{2} \\ \frac{(\Delta t)^2}{2} & \Delta t \end{bmatrix}$$

- Because Φ and Q are matrices, the state and covariance extrapolation equations do not simplify from their general matrix form
- Moreover, like Φ, which is derived from a continuous model, Q is also dependent on Δt and q, the continuous process noise variance:

$$Q = q \begin{bmatrix} \frac{(\Delta t)^3}{3} & \frac{(\Delta t)^2}{2} \\ \frac{(\Delta t)^2}{2} & \Delta t \end{bmatrix}$$

- The details of how this is derived from the continuous model are beyond the scope of the current presentation. See, e.g., Grewal and Andrews.

- Because Φ and Q are matrices, the state and covariance extrapolation equations do not simplify from their general matrix form
- Moreover, like Φ, which is derived from a continuous model, Q is also dependent on Δt and q, the continuous process noise variance:

$$Q = q \begin{bmatrix} \frac{(\Delta t)^3}{3} & \frac{(\Delta t)^2}{2} \\ \frac{(\Delta t)^2}{2} & \Delta t \end{bmatrix}$$

- The details of how this is derived from the continuous model are beyond the scope of the current presentation. See, e.g., Grewal and Andrews.

- $P^{-}(j) = \Phi(j) P^{+}(j-1)\, \Phi(j)^{\top} + Q(j)$ becomes:

- $P^{-}(j) = \Phi(j) P^{+}(j-1) \Phi(j)^{\top} + Q(j)$ becomes:

$$P^{-}(j) = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{21} \\ \phi_{12} & \phi_{22} \end{bmatrix} + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

- $P^{-}(j) = \Phi(j) P^{+}(j-1)\, \Phi(j)^{\top} + Q(j)$ becomes:

$$
\begin{aligned}
P^{-}(j) &= \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{21} \\ \phi_{12} & \phi_{22} \end{bmatrix} + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \\
&= \begin{bmatrix} (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{11} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{12} & (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{21} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{22} \\ (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{11} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{12} & (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{21} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{22} \end{bmatrix} \\
&\quad + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}
\end{aligned}
$$

- $P^-(j) = \Phi(j) P^+(j-1)\, \Phi(j)^\top + Q(j)$ becomes:

$$P^-(j) = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{21} \\ \phi_{12} & \phi_{22} \end{bmatrix} + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

$$= \begin{bmatrix} (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{11} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{12} & (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{21} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{22} \\ (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{11} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{12} & (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{21} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{22} \end{bmatrix}$$

$$+ \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

$$= \begin{bmatrix} \phi_{12}p_{12}\phi_{12} + q_{11} & \phi_{12}p_{12}\phi_{21} + \phi_{12}p_{22}\phi_{22} + q_{12} \\ \phi_{22}p_{21}\phi_{12} + q_{21} & \phi_{22}p_{21}\phi_{21} + \phi_{22}p_{22}\phi_{22} + q_{22} \end{bmatrix}$$

- $P^{-}(j) = \Phi(j) P^{+}(j-1)\,\Phi(j)^{\top} + Q(j)$ becomes:

$$
\begin{aligned}
P^{-}(j) &= \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{21} \\ \phi_{12} & \phi_{22} \end{bmatrix} + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \\
&= \begin{bmatrix} (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{11} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{12} & (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{21} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{22} \\ (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{11} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{12} & (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{21} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{22} \end{bmatrix} \\
&\quad + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \\
&= \begin{bmatrix} \phi_{12}p_{12}\phi_{12} + q_{11} & \phi_{12}p_{12}\phi_{21} + \phi_{12}p_{22}\phi_{22} + q_{12} \\ \phi_{22}p_{21}\phi_{12} + q_{21} & \phi_{22}p_{21}\phi_{21} + \phi_{22}p_{22}\phi_{22} + q_{22} \end{bmatrix}
\end{aligned}
$$

- Note that the last matrix results when we simplify using $\phi_{11} = 0$ (See previous slide)

- $P^-(j) = \Phi(j)P^+(j-1)\,\Phi(j)^\top + Q(j)$ becomes:

$$
\begin{aligned}
P^-(j) &= \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{21} \\ \phi_{12} & \phi_{22} \end{bmatrix} + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \\
&= \begin{bmatrix} (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{11} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{12} & (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{21} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{22} \\ (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{11} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{12} & (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{21} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{22} \end{bmatrix} \\
&\quad + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \\
&= \begin{bmatrix} \phi_{12}p_{12}\phi_{12} + q_{11} & \phi_{12}p_{12}\phi_{21} + \phi_{12}p_{22}\phi_{22} + q_{12} \\ \phi_{22}p_{21}\phi_{12} + q_{21} & \phi_{22}p_{21}\phi_{21} + \phi_{22}p_{22}\phi_{22} + q_{22} \end{bmatrix}
\end{aligned}
$$

- Note that the last matrix results when we simplify using $\phi_{11} = 0$ (See previous slide)
- We've also decluttered by suppressing the $+$ superscripts and $j-1$ indices on the right-hand sides

- $P^-(j) = \Phi(j)P^+(j-1)\,\Phi(j)^\top + Q(j)$ becomes:

$$
\begin{aligned}
P^-(j) &= \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{21} \\ \phi_{12} & \phi_{22} \end{bmatrix} + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \\
&= \begin{bmatrix} (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{11} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{12} & (\phi_{11}p_{11} + \phi_{12}p_{21})\phi_{21} + (\phi_{11}p_{12} + \phi_{12}p_{22})\phi_{22} \\ (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{11} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{12} & (\phi_{12}p_{11} + \phi_{22}p_{21})\phi_{21} + (\phi_{12}p_{12} + \phi_{22}p_{22})\phi_{22} \end{bmatrix} \\
&\quad + \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \\
&= \begin{bmatrix} \phi_{12}p_{12}\phi_{12} + q_{11} & \phi_{12}p_{12}\phi_{21} + \phi_{12}p_{22}\phi_{22} + q_{12} \\ \phi_{22}p_{21}\phi_{12} + q_{21} & \phi_{22}p_{21}\phi_{21} + \phi_{22}p_{22}\phi_{22} + q_{22} \end{bmatrix}
\end{aligned}
$$

- Note that the last matrix results when we simplify using $\phi_{11} = 0$ (See previous slide)
- We've also decluttered by suppressing the $+$ superscripts and $j-1$ indices on the right-hand sides

- The gain equation $K = P^{-}H^{\top}\left[HP^{-}H^{\top} + R\right]^{-1}$ looks like this:

- The gain equation $K = P^{-}H^{\top}\left[HP^{-}H^{\top} + R\right]^{-1}$ looks like this:

$$K = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma^2 \right)^{-1}$$

- The gain equation $K = P^- H^\top \left[H P^- H^\top + R \right]^{-1}$ looks like this:

$$K = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma^2 \right)^{-1}$$

$$= \begin{bmatrix} \dfrac{p_{11}}{p_{11} + \sigma^2} \\ \dfrac{p_{21}}{p_{11} + \sigma^2} \end{bmatrix},$$

- The gain equation $K = P^{-}H^{\top}\left[HP^{-}H^{\top} + R\right]^{-1}$ looks like this:

$$K = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma^2 \right)^{-1}$$
$$= \begin{bmatrix} \dfrac{p_{11}}{p_{11} + \sigma^2} \\ \dfrac{p_{21}}{p_{11} + \sigma^2} \end{bmatrix},$$

where we have once again suppressed the minus(-) superscripts and time step variable j on the right hand side

- The gain equation $K = P^{-}H^{\top}\left[HP^{-}H^{\top} + R\right]^{-1}$ looks like this:

$$K = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma^2 \right)^{-1}$$
$$= \begin{bmatrix} \dfrac{p_{11}}{p_{11} + \sigma^2} \\ \dfrac{p_{21}}{p_{11} + \sigma^2} \end{bmatrix},$$

where we have once again suppressed the minus(-) superscripts and time step variable j on the right hand side

- $\Delta t = 0.1$ sec, 50 samples,
  50% process noise assumption factor

- $\Delta t = 0.1$ sec, 50 samples,
  50% process noise assumption factor
- $m = 1$ kg

- $\Delta t = 0.1$ sec, 50 samples,
  50% process noise assumption factor
- $m = 1$ kg
- Light spring: $k = 1$ N/m

- $\Delta t = 0.1$ sec, 50 samples,
  50% process noise assumption factor
- $m = 1$ kg
- Light spring: $k = 1$ N/m
- Medium damping: $b = 5$ Ns/m

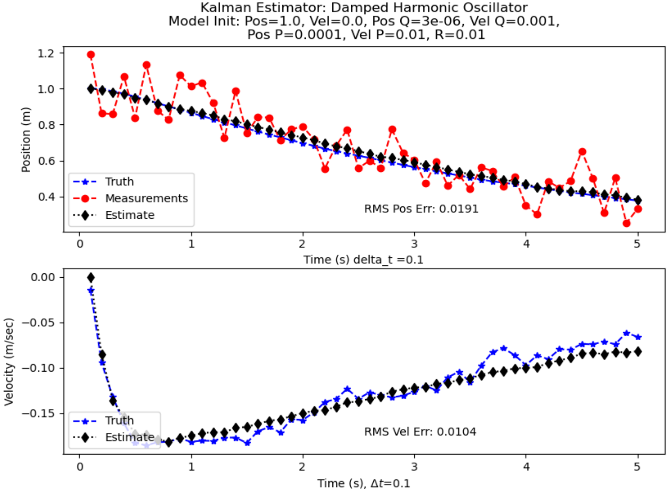
- $\Delta t = 0.1$ sec, 50 samples,
  50% process noise assumption factor
- $m = 1$ kg
- Light spring: $k = 1$ N/m
- Medium damping: $b = 5$ Ns/m

Damped Harmonic Oscillator Estimator

MATLAB/Python Run

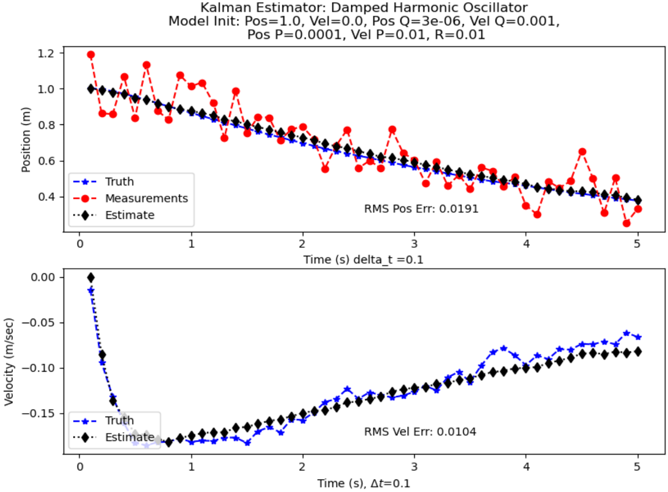
Damped Harmonic Oscillator Estimator

MATLAB/Python Run



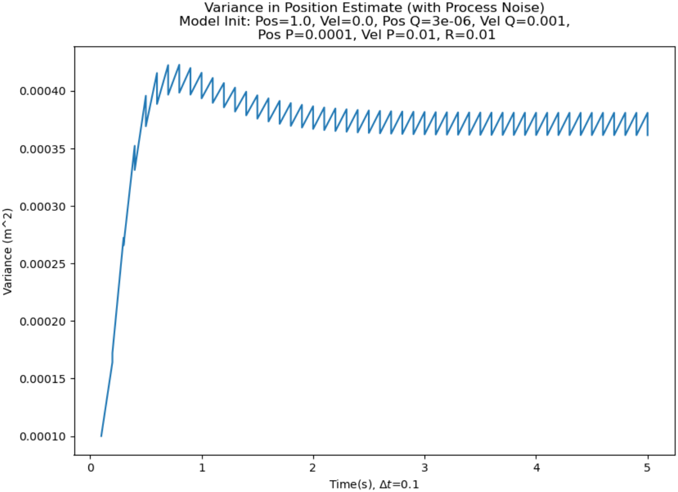
Damped Harmonic Oscillator Estimator

MATLAB/Python Run



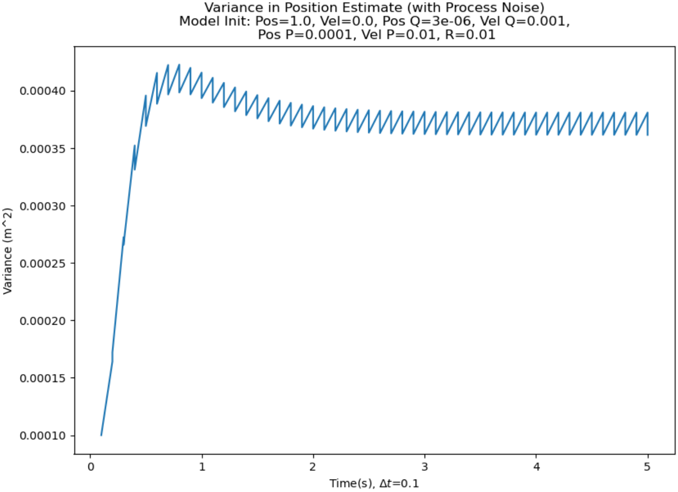
Damped Harmonic Oscillator Estimator

Sawtooth Plots



Damped Harmonic Oscillator Estimator

Sawtooth Plots



- $\Delta t = 1$ sec, 50 samples,
  50% process noise assumption factor

- $\Delta t = 1$ sec, 50 samples,
  50% process noise assumption factor
- $m = 1$ kg

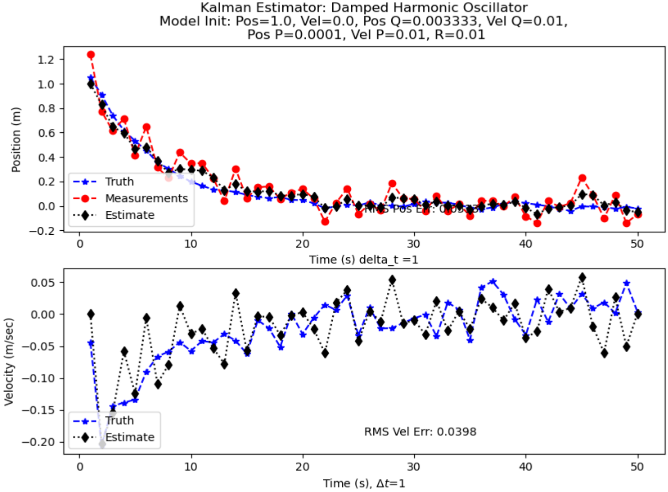
- $\Delta t = 1$ sec, 50 samples,
  50% process noise assumption factor
- $m = 1$ kg
- Light spring: $k = 1$ N/m

- $\Delta t = 1$ sec, 50 samples,
  50% process noise assumption factor
- $m = 1$ kg
- Light spring: $k = 1$ N/m
- Medium damping: $b = 5$ Ns/m

- $\Delta t = 1$ sec, 50 samples,
  50% process noise assumption factor
- $m = 1$ kg
- Light spring: $k = 1$ N/m
- Medium damping: $b = 5$ Ns/m

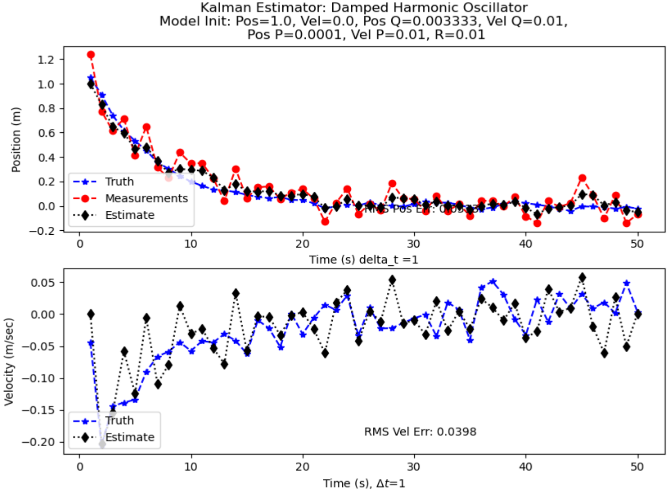
Damped Harmonic Oscillator Estimator

MATLAB/Python Run



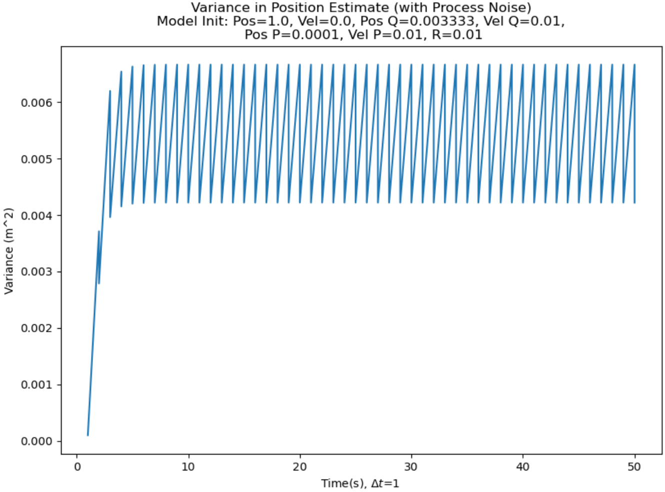
Damped Harmonic Oscillator Estimator

MATLAB/Python Run



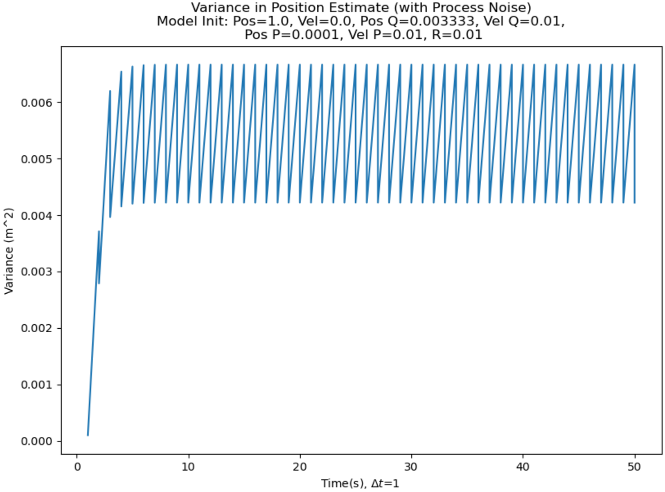
Damped Harmonic Oscillator Estimator

Sawtooth Plots



Damped Harmonic Oscillator Estimator

Sawtooth Plots



- Where are we?

- Where are we?
  - Examined the Damped Harmonic Oscillator and how the discrete model depends on the continuous model from which it was derived

- Where are we?
  - Examined the Damped Harmonic Oscillator and how the discrete model depends on the continuous model from which it was derived
  - Showed results of some Python simulations: See damposc.py on github: `https://github.com/musicarroll/kalman_course`

- Where are we?
  - Examined the Damped Harmonic Oscillator and how the discrete model depends on the continuous model from which it was derived
  - Showed results of some Python simulations: See damposc.py on github: `https://github.com/musicarroll/kalman_course`

- Where are we?
  - Examined the Damped Harmonic Oscillator and how the discrete model depends on the continuous model from which it was derived
  - Showed results of some Python simulations: See damposc.py on github: `https://github.com/musicarroll/kalman_course`
- What's next?

- Where are we?
  - Examined the Damped Harmonic Oscillator and how the discrete model depends on the continuous model from which it was derived
  - Showed results of some Python simulations: See damposc.py on github: `https://github.com/musicarroll/kalman_course`
- What's next?
  - In the next video, we will present some exercises for you to solve.

- Where are we?
  - Examined the Damped Harmonic Oscillator and how the discrete model depends on the continuous model from which it was derived
  - Showed results of some Python simulations: See damposc.py on github: `https://github.com/musicarroll/kalman_course`
- What's next?
  - In the next video, we will present some exercises for you to solve.