The Kalman Filter

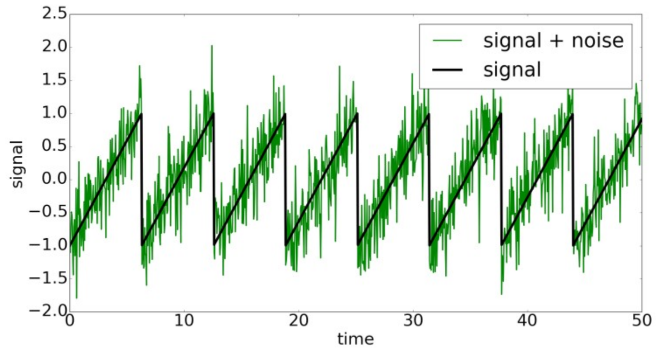
Eli Chertkov

Algorithms interest group talk

Source: nasa.go

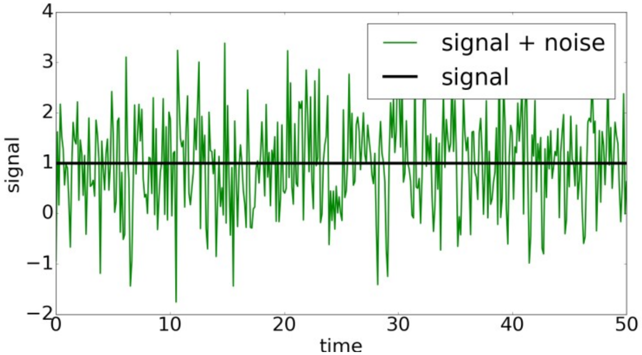
Some history of filtering

A central problem in signal processing is **filtering**, finding a signal in noise



Source: Wikipedia

The **Wiener Filter** was proposed by Norbert Wiener and Andrey Kolomogorov independently in the 1940’s. Wiener worked on applying the filter to the

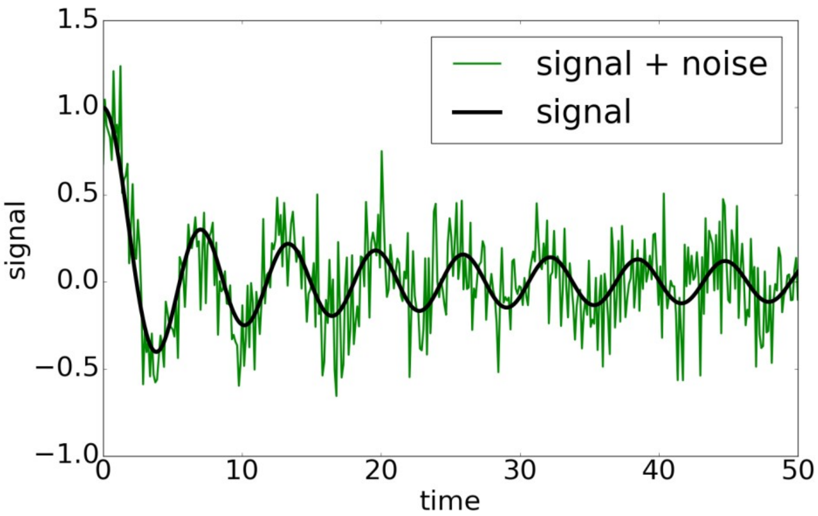


craft guns in

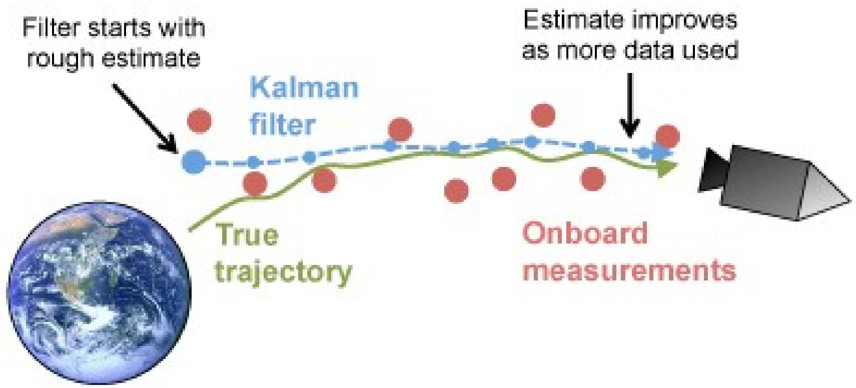
The Wiener Filter was designed for filtering continuous, stationary signals.

Some history of filtering

The Kalman Filter is an extension of the Wiener Filter to non-stationary (and usually discrete) signals.



Rudolf Kalman developed the **Kalman Filter** in the 1960’s. The filter was used to help control spacecraft navigation systems. It was even used in the NASA Apollo missions to the moon!



Source: https://plus.maths.org/content/understanding-

The problem the Kalman Filter solves

The goal of the Kalman Filter is to estimate the time evolution of a system, whose **dynamics** obey a linear stochastic finite-difference equation

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t$$

- $\mathbf{x}_t$: Current state of system
- $\mathbf{F}_t$: State transition matrix
- $\mathbf{x}_{t-1}$: Previous system state
- $\mathbf{B}_t$: Control to state conversion matrix
- $\mathbf{u}_t$: Control (driving forces)
- $\mathbf{w}_t$: Process noise

This estimate is based on **measurements**, which are linearly proportional to the system's state

$$\mathbf{z}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t$$

- $\mathbf{z}_t$: Measurement
- $\mathbf{H}_t$: Conversion matrix
- $\mathbf{x}_t$: System state
- $\mathbf{v}_t$: Measurement noise

Noise is assumed to be Gaussian distributed.

Sketch of derivation of Kalman Filter equations

The Kalman Filter algorithm involves iteratively computing a Gaussian estimate for the current state based on **1) previous estimates** and **2) measurements**.

1) Compute prediction from previous estimates

Estimated mean → $\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} + B_t u_t$

Estimated covariance → $\hat{P}_{t|t-1} = F_t \hat{P}_{t-1|t-1} F_t^T + Q_t$ ← Covariance of process noise

These equations come from inserting the equation of motion into definitions of mean and covariance.

2) Combine with current measurements

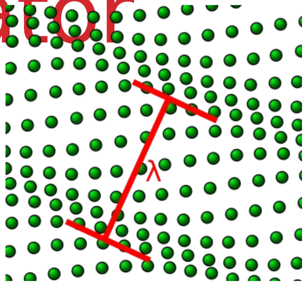
$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - H_t \hat{x}_{t|t-1})$$

$$\hat{P}_{t|t} = \hat{P}_{t|t-1} - K_t H_t \hat{P}_{t|t-1}$$

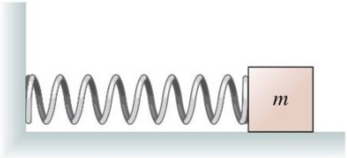
where $K_t = \hat{P}_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}$ is the Kalman gain.

These equations come from adding process and measurement Gaussians together and completing the square.

Example: Simple Harmonic Oscillator



We all know the ubiquitous simple harmonic oscillator and its equation of motion:



$$\ddot{x}(t) = -\omega^2 x(t)$$

The SHO equation is a second-order linear ODE. We can write it as a system of first-order linear ODEs.

$$x_1 \equiv x \qquad \dot{x}_1 = x_2$$

$$x_2 \equiv \dot{x} \qquad \dot{x}_2 = -\omega^2 x_1$$

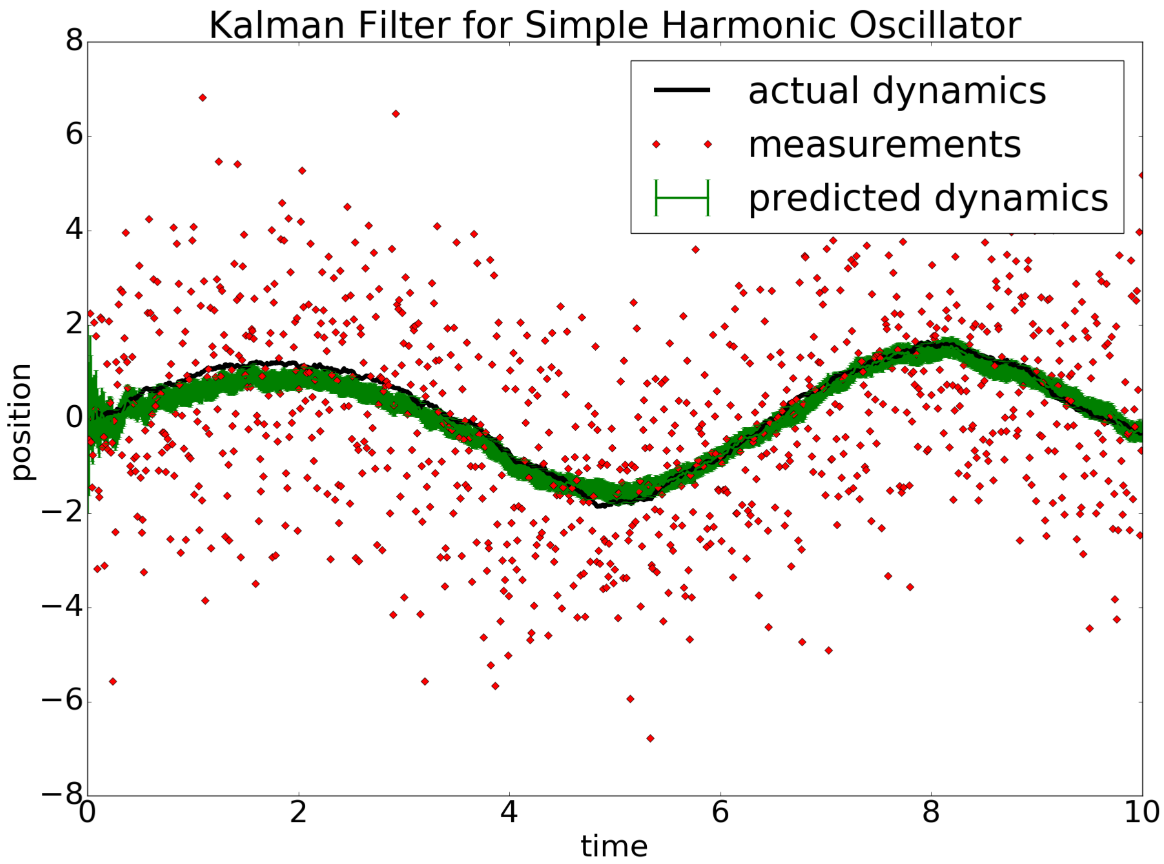
We can now discretize the ODEs with finite-differences to obtain

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_t = \begin{pmatrix} 1 & \Delta t \\ -\omega_0^2 \Delta t & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{t-1}$$

Now we can apply the Kalman Filter.

Example: Simple Harmonic Oscillator

Suppose that we have a SHO subject to random perturbations whose position we can measure with a noisy measuring device. How well can a Kalman Filter pick up the signal?



Example: Launching a satellite into orbit

Suppose that we are a telecommunications company that wants to launch a satellite into geosynchronous orbit.

We know the equations of motion of the satellite under the influence of gravity.

Source: http://www.space.com/19593-amazing-rocket-launches-photos-2013.html

They are (in polar coordinates)

$$\ddot{r} + r\dot{\phi}^2 - \frac{u_r}{m} + \frac{GM}{r^2} = 0 \qquad 2\dot{r}\dot{\phi} + r\ddot{\phi} - \frac{u_\phi}{m} = 0$$

These are non-linear! Does the Kalman Filter apply?

Yes! We can linearize it at each time step by computing the Jacobian. Then we can use finite-differences as before. This non-linear filter, which combines Jacobians with the Kalman Filter is called the Extended Kalman Filter.

Great, but what about the control parameters? How do we pick them?

Digression: PID control

Proportional-Integral-Derivative (PID) control is a simple, general purpose control scheme. The control signal (driving force) has three separate terms, each of which $x(t)$ attempts to force the $y(t)$ system state into a target state

$$u(t) = u_P(t) + u_I(t) + u_D(t)$$

Proportional control

$$u_P(t) = -K_P e(t)$$

Integrated control

$$u_I(t) = -K_I \int_0^t e(t')dt'$$

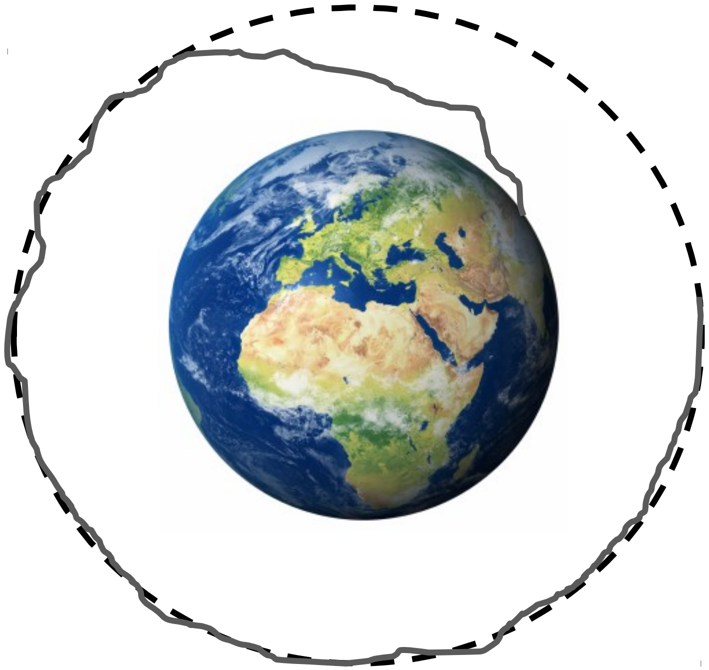
Derivative control

$$u_D(t) = -K_D \frac{de(t)}{dt}$$

where $e(t) = y(t) - x(t)$ is there error.

Example continued: Launching a satellite into orbit

Using PID control along with the Kalman Filter, we can guide a satellite from launch to a geosynchronous orbit, even with a noisy measuring apparatus.



Sketch of the result

Thank you!

You can find example scripts implementing the Kalman Filter in the ipython notebook posted on the group github.