Kalman Filter design

1) Theory

We assume that our model in continuos time is:

$$\begin{cases} \dot{x} = Ax + Bu + w \\ y = Cx + Du + v \end{cases} \tag{1.}$$

$$x = \begin{bmatrix} \theta \\ \omega \end{bmatrix}, u = v \tag{2.}$$

With A,B,C all fully identified and D = 0

With all the independence assumptions about w and v. (Z = 0)

$$\begin{aligned} w &\sim \mathcal{N}(0, Q) \\ v &\sim \mathcal{N}(0, R) \\ Q &= \begin{bmatrix} \sigma_\theta^2 & \sigma_{\theta,\omega}^2 \\ \sigma_{\theta,\omega}^2 & \sigma_\omega^2 \end{bmatrix} \\ R &= \begin{bmatrix} \sigma_{\text{encoder}}^2 \end{bmatrix} \end{aligned} \tag{3.}$$

The system in discrete time is:

$$\begin{cases} x(k+1) = \overline{A}x(k) + \overline{B}u(k) + w \\ y(k) \quad\quad = \overline{C}x(k) \quad\quad\quad\quad + v \end{cases} \tag{4.}$$

With

$$\begin{aligned} \overline{A} &= I + A \cdot \Delta t \\ \overline{B} &= B \cdot \Delta t \\ \overline{C} &= C \end{aligned} \tag{5.}$$

1.1) Problem

How do you calculate Q and R?

1.1.a) R matrix

Assuming that the discretization from the encoder is the only noise corrupting our "true" θ we can see v as a gaussian noise with zero mean and standard deviation of $\sigma_{\text{enc}} = \frac{\text{LSB}}{\sqrt{12}} = 2\frac{\pi}{4096\sqrt{12}}$.

2) The literature

Taken from "Kalman and Bayesian Filters in Python" (Link).

2.1) Chapter 7.3.2

Ignoring the control we can rewrite our state equation as:

$$\dot{x} = Ax + \Gamma\alpha \tag{6.}$$

With $\Gamma = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $\alpha = \dot{\omega}$ describing the acceleration of the angle caused by "something" (control, disturbances, noise, whatever).

Assuming to have a constant acceleration in our sampling period we can discretize it as:

$$x(k+1) = \overline{A}x(k) + \overline{\Gamma}\alpha(k) \tag{7.}$$

With

$$\overline{\Gamma} = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} \tag{8.}$$

The covariance of this process will be:

$$\overline{Q} = \overline{\Gamma}\sigma_\omega^2\overline{\Gamma}^T \tag{9.}$$

For σ_ω just use the rule of thumb:

$$\sigma_\omega \in \left[\frac{1}{2}, 1\right]\alpha_{\max} \tag{10.}$$

$\alpha_{\max}$ can be estimated from the model, simply assume feeding the max voltage to the motor.

This a good starting point but it has a big drawback: it doesn't use the experimental data.

2.1.a) Approximation from 7.3.4

We can see that Q will have the form:

$$\overline{Q} = \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 \\ \frac{1}{2}\Delta t^3 & \Delta t \end{bmatrix}\sigma_\omega^2 \approx \begin{bmatrix} 0 & 0 \\ 0 & \sigma_\omega^2\Delta t \end{bmatrix} \tag{11.}$$

In continuos form:

$$Q \approx \begin{bmatrix} 0 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} \tag{12.}$$

This is suggesting to us that instead of a diagonal matrix we can focus on a lower triangular matrix. And that we need to estimate only one parameter.

Intuitively this make sense, in fact every source of noise will primarily affect the speed (disturbances on the voltage source, disturbances due to the nonlinearity of friction, disturbances due to the motion of the arm etc. all can be seen as a torque disturbance on the shaft, which will in turn act as a noise for the speed). Disturbance for the angle do exists (think about backslash or flexibility of the shaft) but they are not a problem for our case. They are also already accounted for in the R matrix.

2.2) "Autocovariance Least-Squares" (ALS) & "Optimized Kalman Filter" (OKF)

From: [https://en.wikipedia.org/wiki/Kalman_filter#Estimation_of_the_noise_covariances_Qk_and_Rk]

These two methods are respectively described as:

> Extensive research has been done to estimate these covariances from data. One practical method of doing this is the autocovariance least-squares (ALS) technique that uses the time-lagged autocovariances of routine operating data to estimate the covariances.
>
> — Wikipedia

> Another approach is the Optimized Kalman Filter (OKF), which considers the covariance matrices not as representatives of the noise, but rather, as parameters aimed to achieve the most accurate state estimation.
>
> — Wikipedia

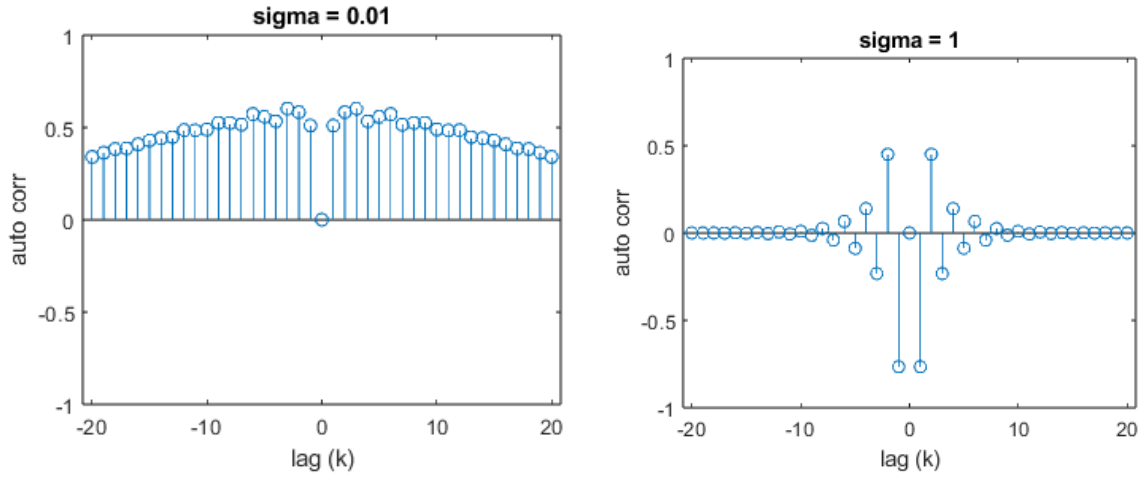
Given these "general outlines" I propose this method:

1 take a reasonable range for σ_ω^2
2 **iterate over the possible range** and for each σ_ω^2:
3 | Build Q as in Equation 12
4 | Calculate the optimal Kalman Gain given the system + R + Q
5 | Simulate the discrete time Kalman Filter with the data from the open loop experiments
6 | Calculate the error of the prediction
7 | Calculate the auto-covariance of the prediction error (residuals)
8 **end**
9 Check if any of the autocovariances is "acceptable"
10 Pick the σ_ω^2 with the "best" autocovariances

A lot of stuff going on there:

- How do you define an autocovariance as "acceptable"
- How do you compare them to find the best?

For now let's see what the error autocovariance looks like:



The auto-covariance (like the autocorrelation) has value 1 for lag $= 0$ and < 1 for everything else. It's also symmetric.

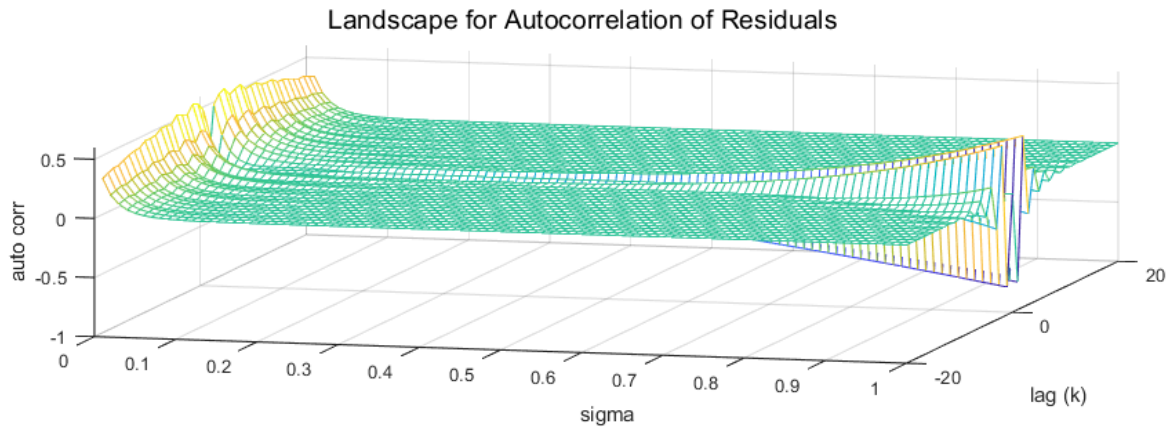
For clarity I forced the 0-lag to 0 (it will also be useful later when calculating the norm).

A signal is a white-noise if the autocorrelation is 0 for every lag value different from 0. This is unreasonable in the real world so a generally accepted whiteness test usually checks if the autocorrelation is bounded betwen ± 0.2 for lags different from 0.

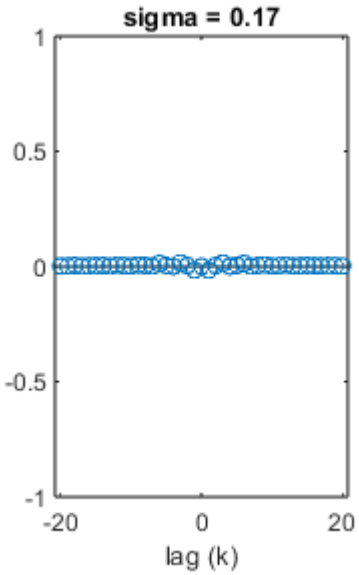
- For $\sigma^2 = 0.01$ we can see how the Kalman Filter does a bad job at predicting, with an almost flat autocovariance.

- For $\sigma^2 = 1$ the quality improves but it's still not acceptable.

We can plot the results in between:

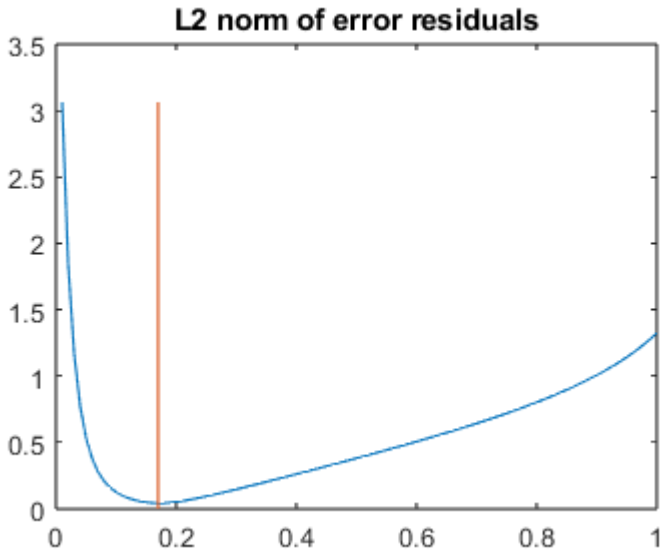


We see that at around $\sigma^2 = 0.1$ the shape is almost flat:

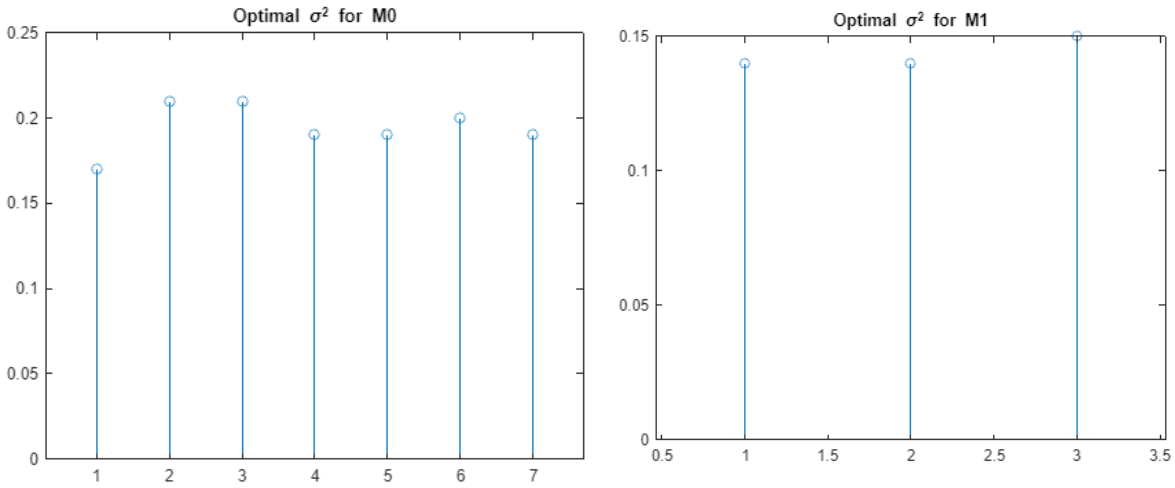


This indicates that our Kalman Filter is able to extract as much information as possible from the data for this specific Q.

We can further optimize it by calculating the L2 norm of the residuals and picking the lowest value.



Repeating this for multiple dataset we see that the "optimal" σ^2 doesn't change that much.



But we still have a small difference between motors.

This was expected due to the effects noticed while experimenting (drifting).

2.2.a) Example of Kalman Filter with optimal Q: