Kalman Filtering

Matt Barnes

Initial Performance

I measured the baseline performance using the noisy measurement as the state estimate. Without any noise, the LQR controller successfully stabilizes the helicopter. Then I increased each error term (σ_x, σ_y) individually until the helicopter crashed. The LQR controller relies on a reliable state estimate to execute a proper control maneuver, and without this the controller may try to correct for non-existent error (i.e. it is only noise). The results for each case are shown in Figure 1. The maximum tolerable noise is one metric to measure system performance.

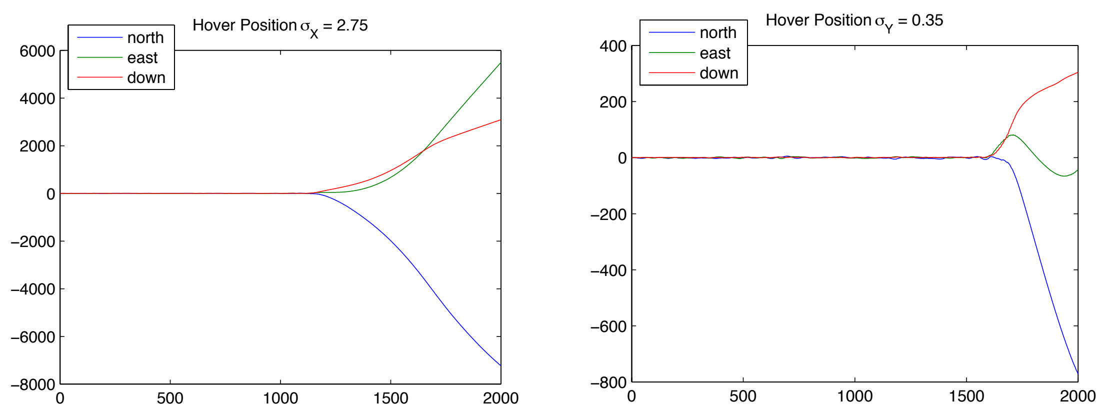

Figure 1: Without a Kalman filter, the helicopter is unstable at $\sigma_x = 2.75$ and $\sigma_y = 0.35$, respectively.

Kalman Filter

I implemented a Kalman filter to provide a better state estimate to the controller, detailed in Figure 2. The linearized system parameters A, B, and C were provided. Thus the only parameters to choose were the initial belief covariance Σ_0 and noise covariance terms R and Q. For Σ_0 I used 10 for every covariance term (off-diagonal entries) and 11 for every variance term (diagonal entries) to guarantee positive definitiveness. For R and Q I used identity matrices with entries of 0.1 and 1.0, respectively.

More complex methods of tuning R and Q certainly exist, such as observing the helicopter's and sensors' behaviors for back-calculation. However, I found these manually tuned parameters worked well enough. The results of the Kalman filter show it successfully enables stabilization at the default

Algorithm Kalman_filter($\boldsymbol{\mu_{t-1}}$, $\boldsymbol{\Sigma_{t-1}}$, $\boldsymbol{u_t}$, $\boldsymbol{y_t}$):

$$\bar{\mu}_t = A\mu_{t-1} + Bu_t$$
$$\bar{\Sigma}_t = A\Sigma_{t-1} + R$$

$$K_t = \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + Q)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(y_t - C\bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C)\bar{\Sigma}_t$$

return μ_t,Σ_t

Figure 2: The Kalman filter algorithm

parameters ($\sigma_x = 0.1$ and $\sigma_y = 0.5$), a feat the baseline system was not able to achieve. Test plots of the helicopter are shown in Figure 3. Unfortunately, my implementation did not work well with much more noise than these values.

Overall, the Kalman filter certainly improves the helicopter's stability. Sources of error could include the linearization of system dynamics (the A, B, and C will not accurately reflect the dynamics far from the normalization point) and the use of manually tuned covariance matrices.

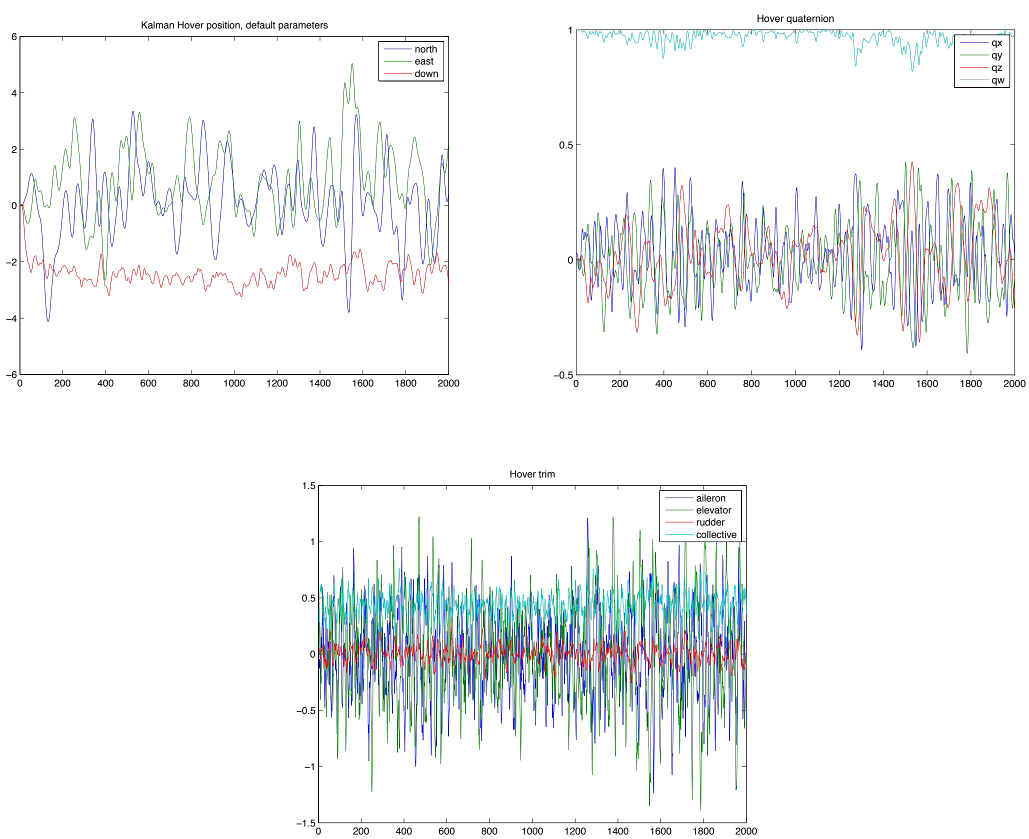

Figure 3: North East Down, quaternion, and trim plots for the helicopter with Kalman filter at default noise parameters ($\sigma_x = 0.1$ and $\sigma_y = 0.5$).