Underwater Navigation System

AUVSI Sub Support Team
Chris Scherr (EE) Gavin Lommatsch (EE) Sam Hooson (EE) Dr. Todd Kaiser (advisor)

Kalman Filter

Kalman filters are disctrete, recursive filters that allow the use of mathematical models to gain an estimate of a system state, despite the presense of significant error in real time measurements. By using a Kalman filter, noisy accelerometer, gyro, and magnetometer data can be combined to obtain an accurate representation of orientation and position. The following images provide some insight into how a Kalman filter operates.

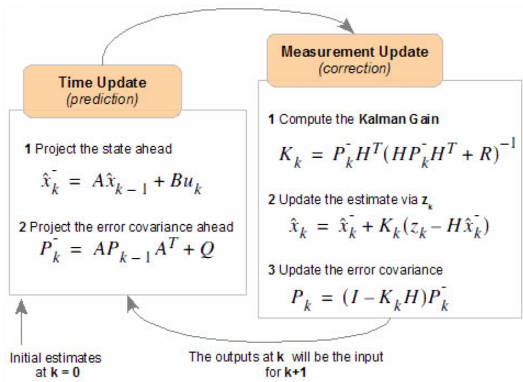

Figure: Stages of the Kalman Filter

The Kalman filter basically consists of two stages. In the first stage a mathematical state model is used to make a prediction about the system state. In the next stage this state prediction is compared to measured state values. The difference between the predicted and measured state is moderated based on estimated noise and error in the system and measurements, and a state estimation is output. The output estimation is then used in conjuction with the mathematical state model to predict the future state during the next time update, and the cycle begins again.

BLUE = inputs ORANGE = outputs BLACK = constants GRAY = intermediary variables

State Prediction
(Predict where we're gonna be)

$$x_{predicted} = \mathbf{A}x_{n-1} + \mathbf{B}\mathbf{u}_n$$

Covariance Prediction
(Predict how much error)

$$P_{predicted} = \mathbf{A}P_{n-1}\mathbf{A}^{\mathbf{T}} + \mathbf{Q}$$

Innovation
(Compare reality against prediction)

$$\tilde{y} = \mathbf{z}_n - \mathbf{H}x_{predicted}$$

Innovation Covariance
(Compare real error against prediction)

$$S = \mathbf{H}P_{predicted}\mathbf{H}^{\mathbf{T}} + \mathbf{R}$$

Kalman Gain
(Moderate the prediction)

$$K = P_{predicted}\mathbf{H}^{\mathbf{T}}S^{-1}$$

State Update
(New estimate of where we are)

$$\mathbf{x}_n = x_{predicted} + K\tilde{y}$$

Covariance Update
(New estimate of error)

$$\mathbf{P}_n = (I - K\mathbf{H})P_{predicted}$$

Inputs:
Un = Control vector. This indicates the magnitude of any control system's or user's control on the situation.
Zn = Measurement vector. This contains the real-world measurement we received in this time step.
Outputs:
Xn = Newest estimate of the current "true" state.
Pn = Newest estimate of the average error for each part of the state.
Constants:
A = State transition matrix. Basically, multiply state by this and add control factors, and you get a prediction of the state for the next time step.
B = Control matrix. This is used to define linear equations for any control factors.
H = Observation matrix. Multiply a state vector by H to translate it to a measurement vector.
Q = Estimated process error covariance. Finding precise values for Q and R are beyond the scope of this guide.
R = Estimated measurement error covariance. Finding precise values for Q and R are beyond the scope of this guide.

Figure: Kalman Filter Equations

The figure above displays the equations and variables that make up a basic Kalman filter. The meaning of most of the variables is fairly obvious. It should be noted that the patameters Q and R represent the estimated expected error in the system model and in the measurements. By changing these values, one can effectively "tune" the Kalman filter to obtain better results. If, for example, the measurements of a system are considered to be very accurate, a small value for R would be used. In this situation the Kalman filter output would follow the measure values more closely than the predicted state estimate.

Linear Kalman Filter for Positioning

```
A = [1 0 0 dt 0 0 0.5*(dt^2) 0 0;     x = [x_pos;
     0 1 0 0 dt 0 0 0.5*(dt^2) 0;          y_pos;
     0 0 1 0 0 dt 0 0 0.5*(dt^2);          z_pos;
     0 0 0 1 0 0 dt 0 0;                   x_vel;
     0 0 0 0 1 0 0 dt 0;                   y_vel;
     0 0 0 0 0 1 0 0 dt;                   z_vel;
     0 0 0 0 0 0 1 0 0;                    x_acc;
     0 0 0 0 0 0 0 1 0;                    y_acc;
     0 0 0 0 0 0 0 0 1];                   z_acc];


                    x = A*x;
```

Figure: State Transition and State Model Matrices for Positioning

The figure shown above is how the state transisition model takes the current input state and predicts a future state. These are some results from 1D generated accelerometer data correspoding to a 1 G acceleration followed by zero acceleration, and then followed by a -1G acceleration. This first example is with R = 0.3 and Q = 0.005

R =0.3, Q = 0.005
Figure: Kalman Filter Output for Acceleration

R=0.3, Q=0.005
Figure: Kalman Filter Output for Velocity

R=0.3, Q=0.005
Figure: Kalman Filter Output for Position

This next example is with the same accelerometer data and with R = 0.05 and Q = 0.3

R=0.05, Q=0.3
Figure: Kalman Filter Output for Acceleration

R=0.05, Q=0.3
Figure: Kalman Filter Output for Velocity

R=0.05, Q=0.3
Figure: Kalman Filter Output for Position

The above plots help to demonstrate the power of the kalman filter. Even with fairly noisy accelerometer data we can achieve accurate estimations of velocity and position. The second example also helps to demonstrate how Q and R affect the filter output. Comparing the two different plots of acceleration, it can be seen that when R is smaller the Kalman output follows the measured acceleration follows more closely.

Extended Kalman Filter (Quaternions)

$$X = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

$$q_{k+1} = q_k + \dot{q} * \Delta t$$

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & -w_1 & -w_2 & -w_3 \\ w_1 & 0 & -w_3 & -w_2 \\ w_2 & w_3 & 0 & -w_1 \\ w_3 & w_2 & w_1 & 0 \end{bmatrix} * \begin{bmatrix} q_{k1} \\ q_{k2} \\ q_{k3} \\ q_{k4} \end{bmatrix}$$

Figure: Kalman State Model for Quaternions & Orientation

The figure above shows the state X and the model used to predict the future quaternion value. Unlike the examples above for positioning, the relation between acceleration and positioning is a linear one and fairly easy to implement. The relation between a quaternion and its derivative is more complicated, so we need to use the above model. In this model q̇ represents the quaternion derivative, which is calculated using the current quaternion value and angular rotation rates. For our system this rotation rates are supplied by the IMU gyros. Below are some data plots showing the true, measured, and kalman filtered values for the four quaternion terms q1, q2, q3, and q4 corresponding to rotations about the three body frame axes. For this example synthetic gyro and magnetometer data was created. Noise was added to both data sets to create "measured" values.

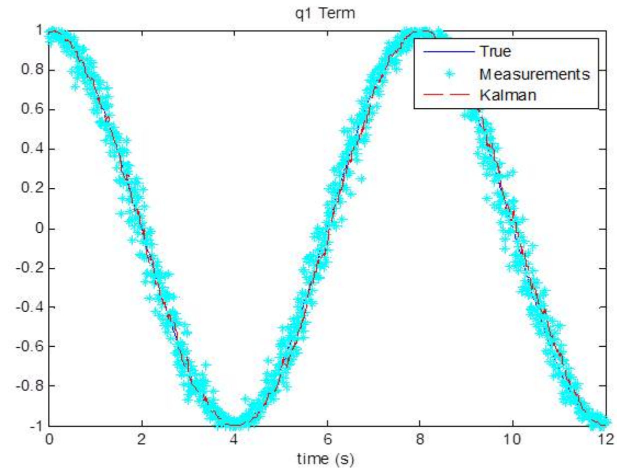

Figure: Kalman Output for Q1

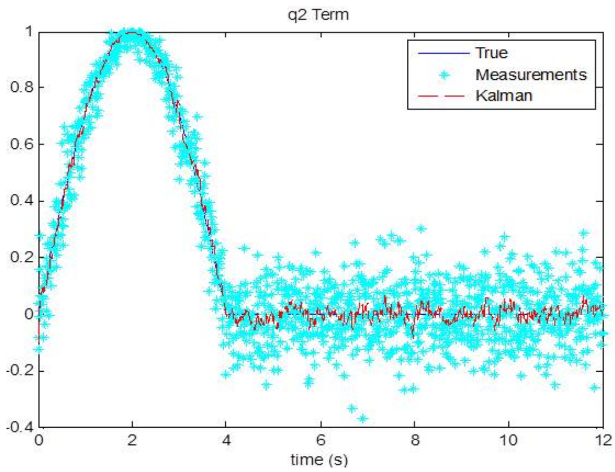

Figure: Kalman Output for Q2

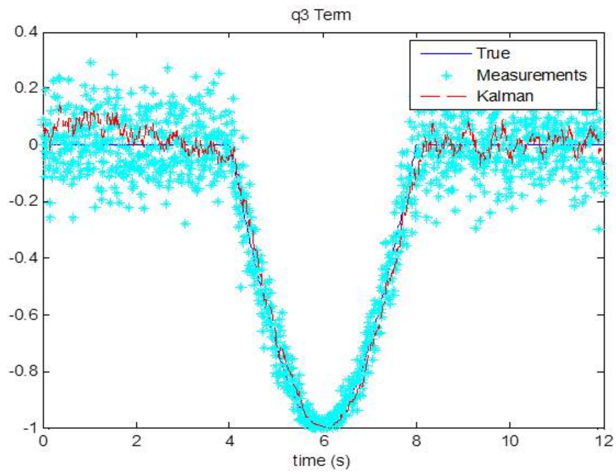

Figure: Kalman Output for Q3

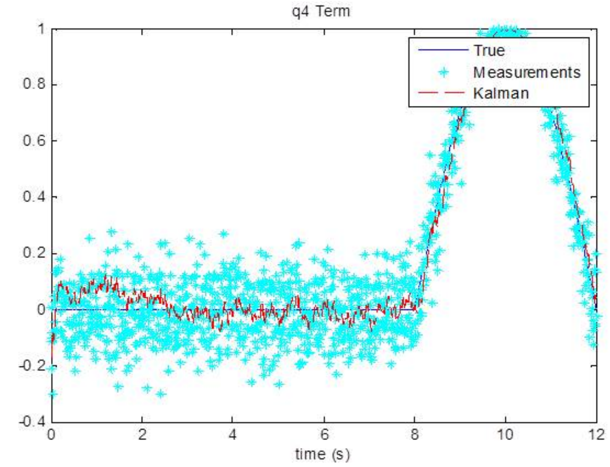

Figure: Kalman Output for Q4

As can be seen in the plots above, the Kalman filter is providing a better estimate of the system quaternion that defines orientation. These results will be furthur improved once proper values for Q and R have been determined. The team is currently in the process of finding these values, which involves tracking and combining sensor error values as the quaternions are calculated.

Even though the proper values of Q and R have yet to be determined, the estimated quaternion is still much improved over the measured values. The videos below show the visualization of the sensor with measurement values and with filtered quaternion values.

Measured Quaternian Visualization

Filtered Quaternian Visualization

From the videos it can be seen that the kalman filtered output is a lot smoother than the measured visualization. There is error in the exact orientation, but this will be fixed with improved Q and R estimates.