Part 2: Kalman Filtering

Table of Contents

Description of Sensors

In order to adequately monitor GDP, we need to consider the prior GDP and prior rate of growth. We will also use two “sensors”. The net exports, which is the difference between exports and imports, will be used as a measure of GDP. The inflation rate of the currency will be considered a measure of the growth rate.

State Transition Model

Let the state at a given time step be represented by the variable $x_k = [x \quad \dot{x}]^T$ where x is the GDP and $\dot{x}$ is the rate of growth of the GDP.

The state transition model is given below with A defined like so, since the next time step’s GDP will be equal to the previous step plus the amount of growth, and the growth rate is assumed to be the same unless disturbed by outside influences. (In other words, there is no control input to set the growth rate, so it is assumed to be constant.)

For this step, we must estimate the process noise covariance, or the uncertainty, labeled as Q in the model. Since many hidden factors could affect GDP, like the state of the world economy outside of this country, the process noise is assumed to be moderately high relative to a GPD of $5,000,000. In other words, we inject some noise into the model to compensate for factors that are not considered.

```
syms delta_t last_x_k last_p_k mu_k;

A = [1 delta_t;
     0 1];

B = [0 0;
     0 0];

Q = [250000 0;
     0 250000];

% Kalman filter transition equations
x_k = A*last_x_k + B*mu_k;

P_k = A*last_p_k*A' + Q;
```

We will assume that a time period of one month (1/12 of a year) is the standard measurement time between intervals for the sensor.

```
A = subs(A, delta_t, 1/12);
```

Sensor Model

The sensor model, which draws upon net exports and inflation, is given below with H defined like so. H is simply an identity matrix because each “sensor” is assumed to have the same weight in this simplistic model. The sensor noise is added to the equation to estimate how accurate the net exports and inflation measurements are. Since the measurements of net exports and inflation are assumed to be pretty accurate, the sensor noise is low relative to the quantities measured.

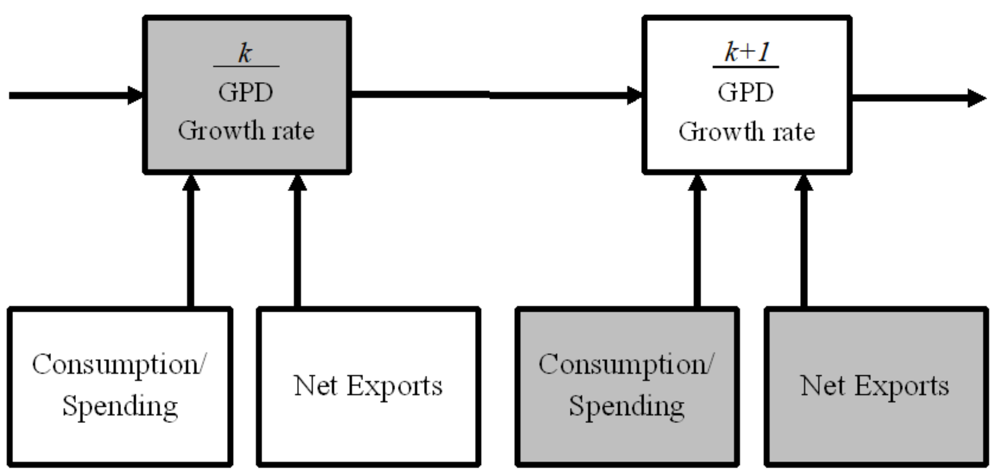
```
sensor_input = sym('sensor_input', [2 1]);
sensor_noise = [100000;
                10000];

H = [1 0
     0 1];

z_k = H*sensor_input + sensor_noise;
```

Network Representation

The network representing the Kalman Filter is shown below. The state is represented by GDP and Growth Rate, as depicted in two time steps (k and k+1). The Markov Blanket is represented by the shaded squares. The information in the future is not known, and the information in the past does not matter for estimating the current state, so these are excluded.



Conditional Probabilities

The conditional probability of the next state in the Kalman Filter is given by $P(x_{k+1}|x_k) = P(e_{k+1}|x_{k+1})P(x_{k+1}|x_k)P(x_t|e_{1:k})$. (Since we are not able to see into the future, the $P(e_{k+1}|x_{k+1})$ is neglected.) These probabilities follow a normal distribution such that $P(x_{k+1}|x_k) \sim N(x_k, P_k)$. To start running the filter, we must choose initial values of mean and variance, or x_0 and P_0.

To begin, the initial value of GDP for a country is assumed to be $5,000,000. The standard deviation of the estimate is 10,000, which means the variance is 100,000,000.

The initial growth rate is assumed to be 1.5% per year, with a standard deviation of 0.2%. For the initial value of GDP, a 1.5% growth rate corresponds to $75,000 with a standard deviation of 10,000 and a variance of 100,000,000. This intermediate computation is necessary because a percentage has no units and would not permit an accurate transition model. In a full implementation of the filter, this parameter would have to be continuously recomputed.

In order to create the initial probability distribution, we have the mean and covariance matrices shown in the following section. To compute the transition and subsequent distributions, we need this probability distribution as well as the information from the sensor model.

Transition Without Sensors

To compute the transition, the initial state estimate and error covariance are set to the values specified above:

```
x_0 = [5000000;
       75000];
P_0 = [100000000 0;
       0 100000000];
```

The state transition is computed first.

```
x_1 = double(A*x_0)
P_1 = double(A*P_0*A' + Q)


x_1 =

     5006250
       75000


P_1 =

   1.0e+08 *

    1.0094    0.0833
    0.0833    1.0025
```

Appropriately, the GDP saw a small increase while the growth rate remained the same. The covariance matrix adjusted accordingly as well.

Now, the Kalman Gain is calculated. Here we must include the measurement noise covariance, R, which is essentially the uncertainty in how well the sensors can model the process. In this case, the estimates of GDP and growth rate provided by the sensors are assumed to be reasonably accurate, so the measurement noise is rather low. In other words, we assume that the GDP and net exports are pretty similar, as are the growth rate and inflation rate.

```
R = [100000 0;
     0 10000];

K_1 = double((H*P_1*H' + R)\P_1*H')


K_1 =

    0.9990    0.0000
    0.0001    0.9999
```

Finally, the new state and covariance predictions are computed. This is done in the absence of sensor information, so the z matrix and associated terms are not used.

```
x_1 = double(x_1)
P_1 = double((eye(2) - K_1*H)*P_1)


x_1 =

     5006250
       75000


P_1 =

   1.0e+05  *

    1.0052    0.0747
   -0.0752    0.0938
```

Since no sensor information was used, the mean values, or state estimate, remained the same. The covariance matrix is slightly different because the Kalman Gain incorporated other sources of noise that shifted the levels of uncertainty surrounding the mean.

Transition With Sensors

The same steps above are repeated for a transition with sensors where the sensor input is measured with \$5,005,000 in net exports and a 1.25% inflation rate (of the original \$5,000,000) after one month. Since the transition model and Kalman gain are the same, only the sensor measurement is computed and the predicted state is recalculated.

```
z_1 = subs(z_k, sensor_input, [5005000; 62500]);

x_1_sensor = double(x_1 + K_1*(z_1 - H*x_1))
P_1


x_1_sensor =
```

```
   1.0e+06 *

    5.1049
    0.0725


P_1 =

   1.0e+05 *

    1.0052    0.0747
   -0.0752    0.0938
```

With a little bit of sensor information, as well as the noise and uncertainty of the sensors and model, the predicted GDP has risen. With a continuous flow of sensor data, this prediction would slowly even out with time to be very near the real GDP assuming all models are accurate.

Published with MATLAB® R2017a