

A3M33MKR

Kalman Filter

Ing. Karel Košnar PhD.

Intelligent and Mobile Robotics Group
Decision Making and Control
Czech Technical University in Prague

October 22, 2014

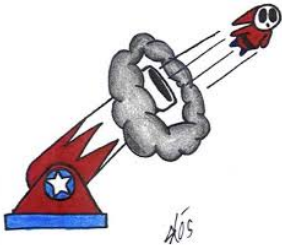
Assignment

We will use a common physics problem.

This assignment will involve firing a ball from a cannon at an unknown angle at an unknown muzzle velocity.

However, we will have a camera that will record the ball's position from the side. The positions measured from the camera have significant measurement error.

We want to know position of the ball as precise as possible.



Mathematical Description

Position $\mathbf{x} = (x, y)$ of the ball is given by initial velocities $\mathbf{v}_0 = (v_{0x}, v_{0y})$ and gravitational acceleration g (we assume the initial position $\mathbf{x} = (x_0, y_0)$)

$$x(t) = x_0 + v_{0x}t$$

$$y(t) = y_0 - v_{0y}t - \frac{1}{2}gt^2 \quad (1)$$

$$v_x(t) = v_{0x}$$

$$v_y(t) = v_{0y} - gt$$

Mathematical Description

In discrete case, the system can be described by

$$\begin{aligned}x_n &= x_{n-1} + v_{n-1x} \Delta t \\y_n &= y_{n-1} - v_{n-1y} \Delta t - \frac{1}{2} g \Delta t^2 \\v_{nx} &= v_{n-1x} \\v_{ny} &= v_{n-1y} - g \Delta t\end{aligned}\tag{2}$$

State description

$$x_n = \mathbf{A}x_{n-1} + \mathbf{B}u \quad (3)$$

u is the control inputs, in this case the control input is the influence of the gravitational acceleration

Kalman filter

used 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.

R = Estimated measurement error covariance.

Kalman filter

inputs

u = Control vector. Constant in this case.

z_n = Measurement vector. Position of the cannon ball measured in this time step. Measured values contain the noise.

Outputs:

x_n = Newest estimate of the current "true" state.

P_n = Newest estimate of the average error for each part of the state.

Kalman filter - How to compute it

Predict, where the cannon ball gonna be

$$x_{predicted} = \mathbf{A}x_{n-1} + \mathbf{B}u \quad (4)$$

Predict the error

$$P_{predicted} = \mathbf{A}P_{n-1}\mathbf{A}^T + \mathbf{Q} \quad (5)$$

Compare reality against prediction, we call it innovation

$$\tilde{y} = z_n - \mathbf{H}x_{predicted} \quad (6)$$

Compare error against prediction

$$S = \mathbf{H}P_{predicted}\mathbf{H}^T + \mathbf{R} \quad (7)$$

Kalman filter - How to compute it cont.

now the key part the Kalman Gain

$$K = \mathbf{P}_{predicted} \mathbf{H}^T \mathbf{S}^{-1} \quad (8)$$

and finally we update the state

$$x_n = x_{predicted} + \mathbf{K} \tilde{y} \quad (9)$$

and covariance

$$P_n = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_{predicted} \quad (10)$$

code explanation

Initialization of the matrices, the size of the state vector is expected to be 4 and the size of measurement vector is expected to be 2.

```
Matrix4f A,B,P,Q;  
Matrix2f R;  
Matrix<float,2,4> H;  
Vector4f x;  
float dt = 0.1;
```

The matrices can be filled like this

```
P << 1, 0, 0, 0,  
     0, 1, 0, 0,  
     0, 0, 1, 0,  
     0, 0, 0, 1;
```

Code description cont.

initialization of the GUI and the system simulator

```
Gui gui;  
System system;
```

Point has x and y. A variable

measurement holds the measured position of the ball

truth holds the true position (used for painting)

kfPosition holds the position estimated by the Kalman filter

Code description cont.

the main loop simulates the one step of the system, stores the position in proper variables

```
for(int i=1; i<nSteps; i++) {  
    system.makeStep(); \\  
    truth = system.getTruthPosition();  
    measurement = system.getMeasurement();
```

estimates the position by Kalman Filter

```
    kfPosition = KalmanFilter(measurement);
```

and plot all position by calling

```
    gui.setPoints(truth, measurement, kfPosition);  
}  
gui.startInteractor();
```