Distance Estimation Using Kalman Filter

Ahmed Razim
Electrial Engineering Department
National Institute of Technology Silchar

February 2023

1 Introduction

Kalman filter is an optimal estimation algorithm that is used to estimate the system state when it cannot be measured directly or when the measurements are noisy. Kalman filters are well suited for estimating the desired signal from noisy measurements. Here we discuss about the estimation of distance of a moving object with constant velocity. The sensor used here is HC-SR04 which has a range of $2cm$–$400cm$. The discrete KF algorithm is implemented using ATMega328p microcontroller on Arduino UNO board.

2 The Kalman Filter

In this project we use an Ultrasonic sensor to measure the distance of a moving object from a reference point. We assume constant velocity dynamics.

$$s = ut \tag{1}$$

The state space model,

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{2}$$

$$y_k = Hx_k + v_k \tag{3}$$

where,
A is the state transition matrix
B is the control matrix
H is the observation matrix
w_k is the process noise
v_k is the measurement noise

Kalman filter is a recursive filter which works in two steps: The prediction step and the Update step. The prediction step equations are:
The state extrapolation equation,

$$\hat{x}_{k+1,k} = A\hat{x}_{k,k} + Bu_k + w_k \tag{4}$$

where,
$\hat{x}_{k+1,k}$ is the predicted state at time step k+1
$\hat{x}_{k,k}$ is the estimated system state at time step k
u_k is the control input

and the covariance extrapolation equation,

$$P_{k+1,k} = AP_{k,k}A^{-1} + Q \tag{5}$$

where,
$P_{k+1,k}$ is the predicted state error covariance matrix
$P_{k,k}$ is the current estimate of the state error covariance matrix
Q is the process noise covariance matrix

The Updation step equations are:
The Kalman Gain,

$$K_k = P_{k,k-1}H^T(HP_{k,k-1}H^T + R_k)^T \tag{6}$$

where,
K_k is the Kalman Gain
$Pn, n-1$ is the prior estimate of the covariance matrix
R is the Measurement noise covariance matrix
The state update equation:

$$\hat{x}_{k,k} = \hat{x}_{k,k-1} + K_k(y_k - H\hat{x}_{k,k-1}) \tag{7}$$

The covariance update equation:

$$P_{k,k} = (I - K_kH)P_{k,k-1}(I - K_kH)^T + K_kR_kK_k^T \tag{8}$$

The equations (4), (5), (6), (7) and (8) are the Kalman filter equations.

3 KF Algorithm Design

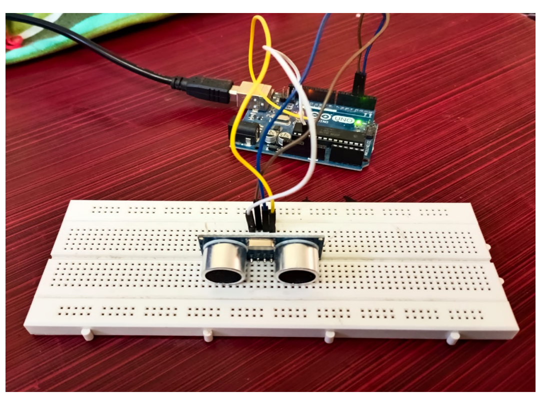

Figure 1: Sensor Setup

Our model is a constant velocity model in one dimension as given in (1). Considering the distance $s = x$ and using the motion equations,

$$x_{k+1} = x_k + \dot{x}_k \Delta t \tag{9}$$

$$\dot{x_{k+1}} = \dot{x_k} \tag{10}$$

From (9) and (10),

State transition matrix, $A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$

Control matrix $B = 0$

Observation matrix, $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$

Let $Q = 200$ and $R = 40$

We use the above values in Kalman filter equations.

4 Result

In Arduino IDE, we coded the Kalman filter equations of our model, extracted the result to a csv file and plotted using matplotlib. For a specific set of initial values, errors and sampling time, we got the following graph.

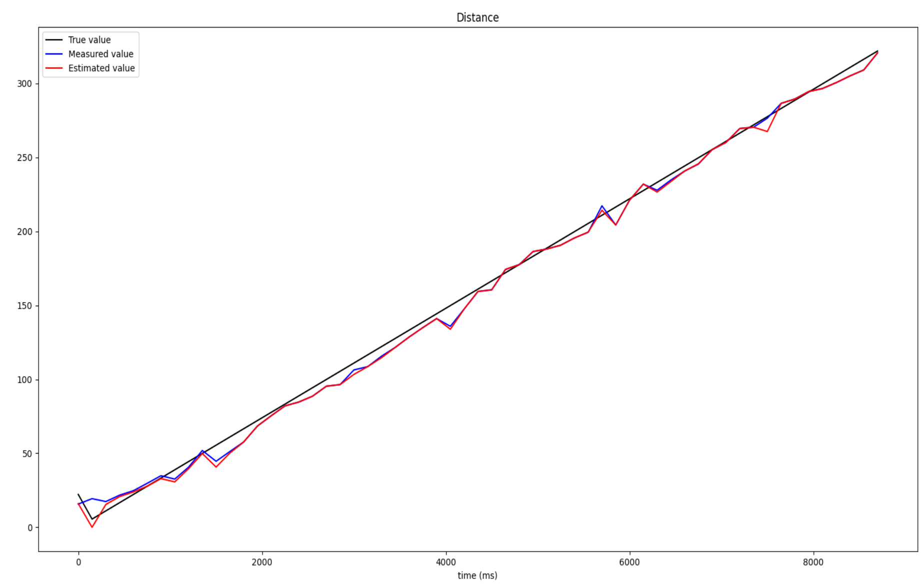

Figure 2: Object Displacement

The graph is obtained for a small time interval. We can see in figure.2 that the estimated value remains closer to the true value, irrespective of the noisy measurement. Hence the distance for linear motion with constant velocity has been estimated successfully using Kalman Filter.