Noise filtering algorithm for control of motion systems

Hnin Lae Wah, Aung Myo Thant Sin
Department of Mechatronic Engineering, Yangon Technological University
Yangon, Myanmar
hninlae.gtc@gmail.com

Abstract. Measurement noise and external disruptions are problematic in real world applications and every control system design. These unwanted disturbances affect the measured output signal and cause the system to provide an inaccurate output. We need to eliminate these noises for stabilizing of the control system design. In this paper, we propose how to eliminate effectively the noises in motion control systems using the Kalman filter. We investigate practically the Kalman filtering algorithm based on the measurements generated by the encoder sensor fixed on DC motor. The Kalman algorithm is used for noise filtering because it is a recursive algorithm and it can handle the noises that occur when the sensor is used to read data. The performance of KF method is enhanced by adjusting its parameters according to the innovation requirements. In the experimental results, the velocity estimation of the Finite-difference (FD) method is demonstrated to compare with the velocity estimation of the Kalman filtering (KF) method.

Keywords - Kalman filter, finite-difference, motion control, Process Noise Covariance, Measurement Noise Covariance

1. Introduction

Nowadays in many applications such as robot manipulator, hundreds of assorted areas, including all navigation forms (aviation, land and sea), moving object tracking, control motors, controlled motion systems, channel tracking in communications, traffic monitoring, fuzzy logic and neural network training, the detection of underground radioactivity and others control, we need to know the precise information and data as position, velocity, and acceleration. However, it is a challenge to obtain precise information in practice. Sensors and GPS are commercially available and are used to monitor the process and to obtain the information, and data, this data from monitoring system is important for decision or control system. However, in practice, the massive data and signals from the sensors and GPS are inaccurate, unsatisfactory and never available in the pure or noiseless form and it is a challenging task to obtain precise information. Every signal faces some distortion due to background noises. Hence, every data and signal must be handled with noise reduction tools. There are available several noise filtering algorithms. However, many of the algorithms have two major weaknesses. Firstly, some algorithms are genuinely complex and hard to implement real world events. Secondly, some algorithms filter the noises or unwanted parts of a signal, but can change the characteristics of the original signal in the process.

One of the most widely applied algorithms in real-time noise filtering and reduction is the Kalman filtering algorithm that can be utilized for stochastic estimation. Kalman filter is a common algorithm to estimate the system state from noisy data and signals, first released in 1960 [1]. Kalman filter has ideal accuracy in linear estimation, so it has been widely accepted and has received attention since it has been theorized and utilized in different areas [2]. Kalman filter uses a set Mathematical equation for predicting the actual value from the noisy measured information and estimates recursively the state of a
dynamic system with two diverse models (i.e. dynamic model and observation model) [3]. While the dynamic system model describes the behaviour of the state vector, the observation model establishes the relationship between measurements and the state vector. Both models are related with statistical properties to portray the exactness of the models. The Kalman filter performs iteratively and recursively some set of mathematical equation on considering that the noise is Gaussian [4].

The performance of the KF method is limited by the numerical accuracy of the calculations and the common problem associated with Kalman filter is how to choose the appropriate process noise covariance and measurement noise variance. Proper set up of the noise variances makes a converged filter [5]. No appropriate set up of the noise variances may obviously decrease Kalman filter's performance and even diverge the filter [6]. Almost all the previous studies, the noise variances are kept constant during the estimation. In the Kalman filtering algorithm, the gain K is computed the model parameters and the noise covariance of process and measurement. So the process and measurement noise variances are usually measured priorly during filter operation. Most publications on noise filtering present simulation results to demonstrate the performance of the KF method as there are not known experimental results. In this paper, we try to gather a deeper insight on how the noise variances affect the filtering results and we test practically the performance of Kalman filter for motion control of a shaft of DC motor which is fixed by encoder sensor. The incremental shaft encoder is one of the most commonly used sensors for obtaining position measurements in electric motors [7]. The results in this paper are experimented, rather than simulated. The performance of the KF method has been evaluated experimentally using a DC motor and encoder sensor fixed on the motor that provides the position measurement only.

This paper is organized as follows: section II reviews the basics of the Kalman filter when the measurement and process noise variances are priorly needed and then implements the system and observation models with transition matrices and Kalman filter equation. Section III implements a Kalman filter to estimate the state of the motor shaft by removing the periodical disturbances introduced by using the position measurement only from the incremental shaft encoder sensor. In section IV we evaluate the experimented data from section III and the estimation results are presented. The experimental results demonstrate the performance of the Kalman filter in motion controlled system. Section V makes the concluding statements.

2. Background Knowledge of Kalman filter

In order to filter the noises or unwanted parts of uncertainty input or noisy data, Kalman filtering algorithm is widely used for real time data processing and is a recursive estimator means that No need to store all past measurements and doesn’t need to reprocess all data each time step and only the current measurement data and the estimated state of the previous time step are required. The system state estimation provided by the Kalman filter can minimize the mean square estimation error within a given range of observations. The Kalman filter estimates the state of a dynamic system, based on two different models, namely the dynamic system model and observation model, that tend to be more precise that than the measurements alone. For using a Kalman filter, the appropriate system model must first be selected. After that, it needs observation model to get a sequence of noisy observations.

2.1. System Model and Observation Model

To estimate the state of the system, various models can be used depending on the filter considered problems, such as second order and third order kinematic models. Using basic equations of velocity, distance and time, the position of a moving object at time $t$ is at $x$ location, then at time $t+1$ the object will be at location $x + ((t + 1) - t) \times \dot{x}$, where $\dot{x}$ is velocity of moving object. Moving in further, acceleration value is available and can add it to above equation and update new position of a moving object to $x + ((t + 1) - t) \times \dot{x} + \frac{1}{2} \ddot{x}((t + 1) - t)^2$. And then jerk, snap, etc. are available in further movement. Formulizing above equations, current state of a moving object consists of its current position, velocity, acceleration, jerk, etc. as a vector $X$. So vector $X$ will have elements which are
\( x, \dot{x}, \ddot{x}, \ldots \) representing position, velocity, acceleration, jerk, etc. with change in time as \( \text{delta}_t = (t + 1) - t \). Using Taylor series and derivatives of position vector for motion paths with respect to time, the following equations for tracking motion paths become:

**Position:** \( x = x_0 + \dot{x}_0 \Delta t + \frac{1}{2} \ddot{x}_0 \Delta t^2 + \frac{1}{6} \dddot{x}_0 \Delta t^3 + \frac{1}{24} \ddddot{x}_0 \Delta t^4 + \frac{1}{120} \dddot{x}_0 \Delta t^5 + \ldots \) (1)

**Velocity:** \( \dot{x} = \frac{dx}{dt} = \dot{x}_0 + \ddot{x}_0 \Delta t + \frac{1}{2} \dddot{x}_0 \Delta t^2 + \frac{1}{6} \ddddot{x}_0 \Delta t^3 + \frac{1}{24} \ddddddot{x}_0 \Delta t^4 + \ldots \) (2)

**Acceleration:** \( \ddot{x} = \frac{d\dot{x}}{dt} = \ddot{x}_0 + \dddot{x}_0 \Delta t + \frac{1}{2} \ddddot{x}_0 \Delta t^2 + \frac{1}{6} \ddddddot{x}_0 \Delta t^3 + \ldots \) (3)

**Jerk:** \( \ddddot{x} = \frac{d\dddot{x}}{dt} = \ddddot{x}_0 + \ddddddot{x}_0 \Delta t + \frac{1}{2} \dddddddot{x}_0 \Delta t^2 + \ldots \) (4)

For second-order kinematic model, the position and velocity of the moving paths are expressed by the linear state space \( \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \). The position, velocity and acceleration of the moving objects are described by the linear state space \( \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} \) for third-order and \( \begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \end{bmatrix} \) for fourth-order model respectively. Calculation with matrix operation is more effective than solving individual equation.

2.1.1. For second-order kinematic model. The position and velocity equations of the moving object ‘\( i \)’ in equation (1) and (2)’ are used with the first and second derivatives being velocity and acceleration. So,

\[
\begin{bmatrix}
[\text{position}] \\
[\text{velocity}]
\end{bmatrix} =
\begin{bmatrix}
x_k \\
\dot{x}_k
\end{bmatrix} =
\begin{bmatrix}
x_0 + \dot{x}_0 \Delta t + \frac{1}{2} \ddot{x}_0 \Delta t^2 \\
\dot{x}_0 + \ddot{x}_0 \Delta t
\end{bmatrix}
\]

\[
\begin{bmatrix}
x_k \\
\dot{x}_k
\end{bmatrix} =
\begin{bmatrix}
1 & \Delta t \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
x_0 \\
\dot{x}_0
\end{bmatrix} +
\begin{bmatrix}
\frac{1}{2} \Delta t^2 \\
\frac{1}{2} \Delta t
\end{bmatrix}
\begin{bmatrix}
\ddot{x}
\end{bmatrix}
\]

(5)

Comparing with ‘equation (5)’ and Newton’s laws of motion concluding that \( X_k = AX_0 + Bu \), will get \( X_k, A \) and \( B \)

\[
X_k = \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix}, A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} \frac{1}{2} \Delta t^2 \\ \frac{1}{2} \Delta t \end{bmatrix}
\]

The initial state value of the moving objects with perfect precision is

\[
\begin{bmatrix}
x_0 \\
\dot{x}_0
\end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

(6)

where \( x_0 \) and \( \dot{x}_0 \) are the initial position and velocity. In the real world, the velocity will be disturbed by noise caused by wind gusts, potholes and other terrible realities. The noise of velocity is a random variable that changes with time. Therefore, a more realistic velocity equation would be:

\[
\text{Velocity: } \ddot{x}_k = \dot{x}_0 + \ddot{x} \Delta t + \nu_k
\]

A comparable condition can be determined for the position:
Position: \( x_k = x_0 + \ddot{x}_0 \Delta t + \frac{1}{2} \dddot{x} \Delta t^2 + p_k \)

where \( v_k \) is the velocity noise and \( p_k \) is the position noise.

The system and observation models were set up agreeing to the kinematic movement characteristics within the normal scenarios (‘see in equation (7) and (8)’). The observation model is established by a measurement system, which can be represented by a linear equation in the following form,

\[
X_k = A_k X_{k-1} + B_k u_k + w_k , \tag{7}
\]

\[
z_k = H x_k + \nu_k , \tag{8}
\]

where \( A \in \mathbb{R}^{n \times n} \) is the state transition matrix related to the previous state \( X_{k-1} \), \( B \in \mathbb{R}^{n \times 1} \) is the control input matrix that relates the control input \( u_k \) and the subscripts \( k \) and \( k-1 \) is the time index. The random variables ( \( w_k \) and \( v_k \) ) represent the process and measurement noises (due to such things as instrumentation errors) respectively, which are assumed to be zero mean Gaussian white noise with normal probability distributions, (i.e. \( w_k \sim N(0, Q_k) \) ) and \( \nu_k \sim N(0, R_k) \), where \( Q_k \) and \( R_k \) are the corresponding covariance matrices). \( z_k \) is the observation or measurement made at time \( k \), \( x_k \) is the state at time \( k \) and \( H \) is the observation model which maps the state space into the observed space. The vector \( X_k \) contains all of the information about the present state of the system, but cannot measure \( X_k \) directly [8]. We measure \( z_k \) instead of \( X_k \), which is a function of \( X_k \) that is corrupted by the noise \( \nu_k \). Using \( z_k \) to obtain an estimate of \( X_k \), but we don't have to extract information from the face value of \( z_k \), because it will be destroyed by noise. We can utilize the information that it provides to a certain extent, but we cannot completely believe it. The first thing we have to do is to calculate the difference between the measurement state \( z_k \) and the model state \( X_k \), this is also called the innovation:

\[
y_k = z_k - H X_k \tag{9}
\]

The innovation, \( y_k \) is defined as the difference between the measurement (observation) \( z_k \) and its prediction \( X_k \) using the information available at time \( k \). The Kalman filter contains the propagation of state estimation and the error covariance matrices at one time step to next one. \( X_k \) state variable was having 2 variables (position and velocity) so its covariance matrix is a 2 x 2 matrix.

Predicted Uncertainty or Process Covariance matrix:

\[
P_k = A_k P_{k-1} A_k^T + Q_k \tag{10}
\]

where \( P_{k-1} \) is previous estimated covariance matrix or state uncertainty. If the initial values of position and velocity are known completely, the error covariance matrix should be initialized with zero covariance matrix:

\[
P_{k-1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{11}
\]

where \( Q_k = B_k B_k^T \sigma^2 = \begin{bmatrix} \frac{1}{2} \Delta t^2 \\ \frac{1}{2} \Delta t^2 \end{bmatrix} \sigma^2 \\ \begin{bmatrix} \frac{1}{2} \Delta t^2 \\ \frac{1}{2} \Delta t^2 \end{bmatrix} \sigma^2 \tag{12}
\]

where \( \sigma^2 \) the process model.
2.1.2. **For third-order kinematic model.** The position, velocity, and acceleration equations of the moving object ‘in equation (1), (2) and (3)’ are used with the first, second, and third derivatives being velocity, acceleration, and jerk. So,

\[
\begin{bmatrix}
\Delta x_k \\
\Delta v_k \\
\Delta a_k \\
\end{bmatrix} = \begin{bmatrix}
x_0 + \dot{x}_0 \Delta t + \frac{1}{2} \ddot{x}_0 \Delta t^2 + \frac{1}{6} \dddot{x}_0 \Delta t^3 \\
\dot{x}_0 + \ddot{x}_0 \Delta t + \frac{1}{2} \dddot{x}_0 \Delta t^2 \\
\dddot{x}_0 + \dddot{x}_0 \Delta t \\
\end{bmatrix}
\]

Comparing with ‘equation (13)’ and Newton’s laws of motion concluding that \( X_k = AX_0 + Bu \);

\[
X_k = \begin{bmatrix}
x_k \\
\dot{x}_k \\
\ddot{x}_k \\
\end{bmatrix}, A = \begin{bmatrix}
1 & \Delta t & \frac{\Delta t^2}{2} \\
0 & 1 & \Delta t \\
0 & 0 & 1 \\
\end{bmatrix} \text{ and } B = \begin{bmatrix}
\frac{\Delta t^3}{6} \\
\frac{\Delta t^2}{2} \\
\Delta t \\
\end{bmatrix}
\]

\[
Q_k = \sigma^2 \int_0^T G_k G_k^T \frac{\Delta t^2}{2} \left[ \frac{\Delta t^2}{2} \Delta t \right] d t = \sigma^2 \int_0^T \begin{bmatrix}
\Delta t^4 / 4 \\
\Delta t^3 / 2 \\
\Delta t^2 \\
\Delta t \\
\end{bmatrix} \left[ \frac{\Delta t^5}{20} \frac{\Delta t^4}{8} \frac{\Delta t^3}{6} \\
\Delta t^2 / 2 \\
\Delta t \\
\end{bmatrix} d t = \sigma^2 \begin{bmatrix}
\frac{\Delta t^5}{4} \\
\frac{\Delta t^4}{8} \\
\frac{\Delta t^3}{16} \\
\Delta t \\
\end{bmatrix}
\]

2.1.3. **For fourth-order kinematic model.** The position, velocity, acceleration, and jerk equations of the moving object ‘in equation (1), (2), (3) and (4)’ are used with the first, second, third and fourth derivatives being velocity, acceleration, jerk and snap. So,

\[
\begin{bmatrix}
\Delta x_k \\
\Delta v_k \\
\Delta a_k \\
\Delta j_k \\
\end{bmatrix} = \begin{bmatrix}
x_0 + \dot{x}_0 \Delta t + \frac{1}{2} \ddot{x}_0 \Delta t^2 + \frac{1}{6} \dddot{x}_0 \Delta t^3 + \frac{1}{24} \dddot{x}_0 \Delta t^4 \\
\dot{x}_0 + \ddot{x}_0 \Delta t + \frac{1}{2} \dddot{x}_0 \Delta t^2 + \frac{1}{6} \dddot{x}_0 \Delta t^3 + \frac{1}{24} \dddot{x}_0 \Delta t^4 \\
\dddot{x}_0 + \dddot{x}_0 \Delta t + \frac{1}{2} \dddot{x}_0 \Delta t^2 + \frac{1}{6} \dddot{x}_0 \Delta t^3 + \frac{1}{24} \dddot{x}_0 \Delta t^4 \\
\dddot{x}_0 + \dddot{x}_0 \Delta t + \frac{1}{2} \dddot{x}_0 \Delta t^2 + \frac{1}{6} \dddot{x}_0 \Delta t^3 + \frac{1}{24} \dddot{x}_0 \Delta t^4 \\
\end{bmatrix}
\]

Comparing with ‘equation (15)’ and Newton’s laws of motion concluding that \( X_k = AX_0 + Bu \);
\[ X_k = \begin{bmatrix} x_k \\ x_k \\ x_k \\ x_k \\ x_k \\ x_k \end{bmatrix}, A = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 & \Delta t^3/6 \\ 0 & 1 & \Delta t & \Delta t^2/2 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} \Delta t^4/24 \\ \Delta t^3/6 \\ \Delta t^2/6 \\ \Delta t \end{bmatrix} \]

\[ Q_k = \sigma^2 \int_0^T G_k G_k^T = \sigma^2 \int_0^T \begin{bmatrix} \Delta t^3/6 \\ \Delta t^2/6 \end{bmatrix} \begin{bmatrix} \Delta t^3/6 \\ \Delta t^2/6 \end{bmatrix} \text{ and } R = \begin{bmatrix} \Delta t^7/252 & \Delta t^6/72 & \Delta t^5/30 & \Delta t^4/24 \\ \Delta t^6/72 & \Delta t^5/20 & \Delta t^4/8 & \Delta t^3/6 \\ \Delta t^5/30 & \Delta t^4/8 & \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^4/24 & \Delta t^3/6 & \Delta t^2/2 & \Delta t \end{bmatrix} \]

2.2. Kalman Gain

Kalman gain is basically used to decide update value for moving object’s state and is just a simple percentage formula. Gain value is to indicate how much we trust innovation and compare uncertainty in prediction and measurement methods by calculating percentage of uncertainty in system model value to that of total uncertainty.

\[
\text{Kalman Gain ( } K_k \text{) } = \frac{\text{Uncertainty in model's state}}{\text{Uncertainty in model's state + Uncertainty in observation}} = \frac{(P_k + H^T H)}{(H + P_k + H^T H + R)}
\]

Where uncertainty in predicting state is matrix \( P_k \) (2x2 matrix) for second-order model and that measurement noise Covariance matrix or uncertainty matrix in measured value is matrix \( R(1x1 \text{ matrix}) \). Both of them are of different size. Hence, we use \( H(1x2) \) and transpose of \( H(2x1) \) matrices to convert \( P \) matrix to correct size or dimensions. If system model value and its uncertainty \((X, P)\), measure value and its uncertainty \((z, R)\) and Kalman gain values have in hand, it is enough to correct the estimated state \( \hat{X} \) and its corresponding covariance matrix \( \hat{P} \). The hat over the \( \hat{X} \) means that it is the state estimation \([9]\). Kalman filter estimates a state iteratively from one time step to the next step using the above equations into two groups: prediction and correction \([10]\). An implementation flowchart of the KF algorithm is summarized in ‘figure 3’.

Kalman filter can adjust its gains according to innovation requirements. However, it has large computation complexity due to its multiple state variables and has difficulty in calculating the matrices. Kalman gain \( K_k \) is computed by the given model parameters and noise covariances of the process and the measurement respectively. To determine \( K_k \), the noise covariance matrices \( Q_k \) and \( R \) are priorly needed to estimate from the observation data. The process noise covariance matrix \( Q_k \) might change each time step according to \( Q_k \) equation. So the measurement noise variance \( R \) is estimated priorly during filter operation.

3. Implementation of Kalman Filter to Motion control

In order to survey the Kalman filter’s performance on real world signal and data, the filter is evaluated experimentally using a digital position encoder fixed on the 12V brushed DC motor with a 100:1 gear ratio, controlled by the Arduino IDE from a PC computer with a sampling period of 1\( \mu \)s. For this
The experimental setup shows in ‘figure 1’, a sinusoidal signal with amplitude of 50deg and \( r(t) = \text{Amplitude} \ast \sin(2 \ast \pi \ast f \ast \text{time}) \), and an encoder which can provide 64 counts per revolution of the motor shaft are integrated into the experimental process. The position readings from the encoder are acquired at fixed time intervals each \( \Delta t = 1 \mu s \), it counts the number of pulses from the encoder sensor during a fixed interval of time. If the fixed time interval is too long, the encoder will not sense any new data until the next consecutive time.

![Figure 1](image1.png)

**Figure 1.** Block diagram of the velocity control system of the motor shaft with (a) Kalman and (b) FDM algorithms.

The position generated from the incremental encoder is highly affected by the quantization noises and Gaussian noises. In order to see the noise behavior and resolution of the encoder sensor, the output signal \( z_k \) obtained from the encoder sensor and its uncertainty of the measurement is evaluated with a sampling period of time. Experimental results express the change of the velocity estimations by using Kalman filtering. For this investigation, finite difference method (FDM) is integrated into the experimental system for comparison with Kalman filtering method. ‘Figure 1’ shows the block diagram sketches of the velocity control system of the motor shaft with (a) Kalman and (b) FDM algorithms.

![Figure 2](image2.png)

**Figure 2.** Experimental setup for Noise Filtering.

The control algorithm needs to estimate the velocity of the motor shaft. Note that the velocity is not measured. In order to solve this problem, one can use the Kalman filter with the encoder sensor for position measurement (see in ‘figure 1(a)’). The Kalman model can be used to describe the relationship between two measurements: a measurement (collected in the vector \( X \)) and a second measurement (collected in the vector \( z_k \)). Then, the goal of the Kalman filter is to perform the "best" measurement fusion between the primary and secondary measurements. Otherwise, if we need to estimate the rate of change of position with respect to time in this case, we can use the finite difference method to calculate the approximate value of the speed. Three types of the finite difference method are basically considered: forward, backward, and central finite differences. The backward finite difference is applied to compare with KF. The formula for the backward finite difference is:

\[
\hat{\dot{\theta}} = \frac{\theta_t - \theta_{t-1}}{T^2}
\]

where \( \hat{\dot{\theta}} \) is the output velocity estimation of backward FD method, \( \theta_t \) represents the angle of the motor shaft at current time \( t \), \( \theta_{t-1} \) is the angle of the motor shaft at previous time \( t-1 \) and \( T^2 \) is the sample time period.

In off-line sample measurements, determining the measurement noise covariance \( R \) is usually
performed with the help of VAR( ) function in excel and the computed variance will be used as the constant value of $R$ in the Kalman process. However, the measurement error does not remain constant in on-line measurements. So, the measurement noise is needed to adjust adaptively through experimentation to obtain desired filter performance during filtering operation. The position readings from the encoder are stored in a data file for analysis using MATLAB. Users need to set the initial values $\hat{X}_0$ and $P_0$ for KF in the initialization step. Different from the most publications which keep $Q_k$ and $R$ constant. In this paper, the $Q_k$ of the KF is iteratively estimated and updated during each correction step and $R$ is adaptively adjusted through the experiments.

**Figure 3. Structure of Kalman filter.**

4. **Experimental Results**

In this paper, to investigate the Kalman filter performance with second-order model, the noisy position data from the digital encoder sensor is filtered with the Kalman filtering (KF) method and Finite Difference (FD) method used for comparison with KF. The velocity estimation results of the KF and FD methods are demonstrated using the experimental output data of a DC motor’s shaft angle. ‘Figure 4(a) and 4(b)’ show the output velocity estimation of FD filtering method using 0.125Hz and 1Hz and then ‘figure 4(c)’ gives the velocity estimation of KF method using 0.125Hz. Firstly, both filtering methods produce the noisy velocity estimation as shown in ‘figure 4’ in which red lines represent the desired velocity signal, and the blue lines represent FD and KF velocity estimations recorded from the experiments. As above-explained, Kalman filter can adjust its gains according to innovation requirements so we can adjust its measurement noise covariance, but we cannot adjust FD filtering as needed.

Figures from 5(a) to (c)’ give the output velocity estimation with 0.125Hz under the different measurement noise covariance $R$ value and we can see the performance of Kalman filter from the experimental results. According to the experimental evaluation, ‘figure 5 (a)’ show noisy velocity estimation using small magnitude of $R=0.01$ and ‘figure 5 (c)’ has large overshooting and a significant
signal time delay at a large magnitude of $R=10$. The action of this filter eliminates the noise in the choice of appropriate measurement noise variance $R=1.5$ (see blue ‘figure 5 (b)’). And then these experiments repeat at frequency changing by adjusting the value of $R$ shown in ‘figure (6)’. As you can see, ‘figure 6 (a)’ has noisy output signal and ‘figure 6 (c)’ produces a little overshoot and delay output signal. In ‘figure 6 (b), more smooth velocity output is produced by using the value of $R=1$.

![Figure 4.](image1)

**Figure 4.** (a) Velocity estimation result of FD with 0.125Hz, (b) FD result with 1Hz and (c) Velocity estimation result of KF with $f=0.5$Hz and $R=0.00001$ by using second-order model.

![Figure 5.](image2)

**Figure 5.** Velocity estimation of KF, filter performance evaluated on sinusoidal motions with 0.125Hz and changing magnitude of R (a) $R=0.01$, (b) $R=1.5$ and (c) $R=10$.

According to the above-mentioned experimental analysis and test, it may be seen that the FDM method cannot eliminate the noises of input signals and it produces a noisy output estimation at any frequency. The Kalman filtering algorithm produces a smooth filter result, but it requires the appropriate value of measurement noise covariance and, may get a more accurate performance if the measurement noise covariance is adaptively adjusted in real time. When the measurement of noise value is inappropriate, the output result will be far from the desired value. The produced estimation should be smooth without a large delay time to avoid instabilities. As it can be experimented, we should increase the magnitude of $R$ if the estimation can’t keep smooth performance without overshoot and reduce the magnitude of $R$ if the user seems to be overshooting. The value of $R$ should be increased as much as possible to obtain a properly smooth signal without a great filter delay. In order to keep the filter responsive to the changes of frequency changes, the value of $R$ should be reduced if the filter cannot follow the measured position signal.

![Figure 6.](image3)

**Figure 6.** Velocity estimation of KF, filter performance evaluated on sinusoidal motions with 1Hz and changing magnitude of R (a) $R=0.001$, (b) $R=1$ and (c) $R=10$. 


When the system model and measurement relationship are perfect, the Kalman filter can provide the best (minimum variance, unbiased) state estimation for noisy measurements at any frequencies. When they are not perfect, the filter can incorrectly estimate the state, and the estimation will diverge. It can be seen that the filtering with KF has more precision and less noise than that by differentiation method. The velocity estimations provided by FDM (see blue line in ‘figure 4(a) and 4(b)’) is not as good as velocity estimations obtained from the KF method (see blue line in ‘figure 5(b) and 6(b)’).

5. Conclusion

In this paper, the performance of the Kalman filtering algorithm has been investigated based on real-time experimental measurements in a motion control system. Regarding this, the velocity estimation of the KF method has been analyzed in various frequency ranges using the incremental encoder. According to this investigation, the accuracy of the KF method is limited by the numerical precision of the calculations and by the adaptive choosing of the measurement noise covariance. In order to obtain the smooth and optimal state estimation results from Kalman filter without delay time, its measurement noise covariance R is needed to adjust adaptively through the experimentation during filter operation. To maintain the filter responds to the changes of frequencies, we should adaptively adjust the measurement covariance. We should increase the magnitude of R if the estimation can’t keep a smooth estimation without a large delay time, and reduce the magnitude of R if the user seems to be overshoot. The measurement noise covariance R is an significant influence and important parameter for the Kalman filter.

References

[1] Kalman R E 1960 A new approach to linear filtering and prediction problems J. Basic Eng. (ASME) vol 82D pp 35–45
[2] Bar-Shalom Y, Li X R, and Kirubarajan T 2001 Estimation with Application to Tracking and Navigation: Theory Algorithm and Software (New York: Wiley)
[3] Tripathi R P, Ghosh S and Chandle J O 2016 Tracking of object using optimal adaptive Kalman filter 2nd IEEE International Conference on Engineering and Technology
[4] Kleinbaur R 2004 Kalman filtering with implementation of MATLAB pp 5-12
[5] Almagbile A, Wang J and Ding W 2010 Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration Journal of Global Positioning Systems vol 9 no 1 pp 33-40
[6] Mohamed A and Schwarz K 1999 Adaptive Kalman filtering for INS/GPS Journal of Geodesy vol 73 pp 193-203
[7] Rodriguez-Maldonado J 2019 Estimation of angular velocity and acceleration with Kalman filter, based on position measurement only Measurement pp 130-136
[8] Thombre D V, Nirmal J H and Lekha D 2009 Human detection and tracking using image segmentation and Kalman filter International Conference on Intelligent Agent & Multi-Agent Systems Chennai pp 1-5
[9] Welch G and Bishop G 2001 An introduction to the Kalman filter SIGGRAPH vol 7 no 1 Citeseer
[10] Kirchhoff J and von Stryk O 2018 Velocity estimation for ultralightweight tendon-driven series elastic robots IEEE Robotics and Automation Letters vol 3 no 2 pp 664-671