A Java Tool for Exploring State Estimation using the Kalman Filter

Declan Delaney[1] and Tomas Ward[2]

[1]*Department of Computer Science,*
National University of Ireland, Maynooth
IRELAND
E-mail: decland@cs.may.ie

[2] *Department of Electronic Engineering,*
National University of Ireland, Maynooth
IRELAND
E-mail: tomas.ward@eeng.may.ie

_____

Abstract – **This paper describes a novel software application that assists in understanding the process of estimating the state of a linear dynamic system based on noisy output signal measurements. An interactive Java tool, based on the Kalman Filter, is described. This consists of two main parts, a simple one-dimensional filter and a multi-dimensional filter tool. The user can set all input parameters through a single interface or by following a series of guided steps. Raw measurement data can be generated automatically or inserted manually by the user and the results of estimating the true system state are then presented as a series of graphs in real-time. The application is described through the use of two simple examples. Such an application could be used to teach signal and systems engineering.**

Keywords – **Kalman Filter, Java application, Signal and systems engineering, Computer assisted learning**

_____

I INTRODUCTION

Formal engineering education appeared for the first time around the middle of the 18th century [1]. As the discipline evolved, various inventions set the standards for 'modern' engineering. These inventions called into existence new industries and new standards for education and training [2]. Prior to World War II the typical electrical/electronic engineer studied very little fundamental science and mathematics, with the emphasis resting on the acquisition of experience [3, 4]. By the end of World War II all this had changed as it was realised that fundamental knowledge of science led directly to new applications in the military domain, with consequential effects on engineering curricula [5] [1]. During the past decade the World Wide Web and computer-assisted learning has again posed interesting challenges and possibilities for approaching the education of electrical and electronic engineers [6 -9].

Fundamental understanding of key concepts in signals and systems is often difficult to teach successfully. In dynamic linear systems, the concept of estimating the state variables from a series of noisy measurements is one such example. The most widely accepted algorithm for performing an estimation of this kind is the Kalman Filter. This algorithm is deceptively simple and its thorough understanding encapsulates a number of engineering topics such as statistical modelling, engineering computation, linear system dynamics, measurement science and digital signal processing [10]. Studying the operation of the Kalman filter therefore leads to an appreciation of the inter-disciplinary nature of electronic engineering. However, a deeper understanding of the theory, and awareness of practical implementation issues, can only be experienced by employing the filter in practical situations. A computer-based tool that allows experiments based on 'what if' scenarios would challenge users, motivate them to ask questions and facilitate the consolidation of their engineering knowledge.

This paper describes a novel Java implementation of the Kalman Filter, which aims to assist users in understanding the effects of noise on the output signals generated by a linear dynamic system and on estimating the state of such a system. Two distinct implementations are described - a two-dimensional and a multi-dimensional Kalman Filter – and two illustrative examples are provided.

In section II we briefly examine the origins of the Kalman Filter and describe the theory underlying its operation. Sections III and IV describe the two-dimensional and multi-dimensional Java implementations respectively together with examples that illustrate their use. Inspiration for the

application derived from one of the authors' industrial experience in seismic exploration and from experience in teaching electronic engineering students.

II   THE KALMAN FILTER

The first method for forming an optimal estimate from noisy data is the method of least squares, a discovery attributed to Carl Friedrich Gauss in 1795 [10]. During World War II Norbert Wiener worked on the problem of automatically controlling the direction of anti-aircraft fire using noisy radar information and derived the solution for the least-mean-squared prediction error in terms of the autocorrelation functions of the signal and noise. By combining probability theory and the idea of state variables Kalman subsequently derived the Weiner Filter, which became known as the Kalman Filter

a) What is the Kalman Filter?

The Kalman Filter is named after Rudolf Kalman who first introduced the algorithm in 1960. It has been employed in myriad applications including process control systems, vehicle tracking, marine navigation, geology, demographic estimation and stock price prediction. It estimates the instantaneous state of a linear dynamic system perturbed by Gaussian white noise by using measurements that are linearly related to the system state but that are corrupted by Gaussian white noise. The filter recursively minimizes the mean square estimation error without directly observing the system state or knowing the nature of the modelled system. We only observe some measurements using an array of noisy sensors – see Figure 1.
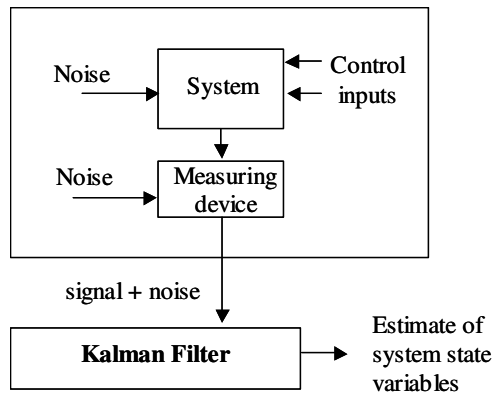

Figure 1: The Kalman filter estimates the system state variables based on a multidimensional signal + noise as input

The system state itself evolves with time under the effect of random perturbations or control inputs. The Kalman Filter then provides an optimal estimate of the unobserved system states and their uncertainties based on noisy measurements of the process. The Kalman filter operates online so that the best estimate of the system state and its uncertainty can be computed by updating the previous estimates with new measurements.

b) Kalman Filter Theory

An outline description of the Kalman Filter algorithm is presented here in equations (1) to (7). For a detailed theoretical explanation of the Kalman Filter refer to [10] and [11].

The Kalman Filter attempts to estimate the n-dimensional state vector x of a first order, discrete-time controlled process with inputs governed by a linear stochastic difference equation, and an m-dimensional measurement vector:

$$x_\kappa = \Phi_{\kappa-1} x_{\kappa-1} + B u_{\kappa-1} + w_\kappa \qquad (1)$$

$$z_\kappa = H_\kappa x_\kappa + v_\kappa \qquad (2)$$

where k is the time index, w_k is the process noise, v_k is the measurement noise, Φ is the state transition matrix, B is the control matrix, u is the control input vector, H is the sensitivity matrix and z is the measurement vector.

Equation (1) defines the *system dynamic model* and equation (2) defines the *measurement model*.

The Kalman Filter iteratively applies two stages of computations using feedback control:
1. time update computations (the prediction) – equations (3) and (4);
2. measurement update computations (the adjusted prediction) – equations (5) to (7)

$$\hat{x}_\kappa^- = \Phi_{\kappa-1} \hat{x}_{\kappa-1} + B u_{\kappa-1} \qquad (3)$$

$$P_\kappa^- = \Phi_{\kappa-1} P_{\kappa-1} \Phi_{\kappa-1}^T + Q_{\kappa-1} \qquad (4)$$

$$K_\kappa = P_\kappa^- H_\kappa^T \left(H_\kappa P_\kappa^- H_\kappa^T + R_\kappa \right)^{-1} \qquad (5)$$

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H_k \hat{x}_k^- \right) \qquad (6)$$

$$P_\kappa = \left(I - K_\kappa H_\kappa \right) P_\kappa^- \qquad (7)$$

where $\hat{x}_k^-$ is the a priori state estimate at step k, $\hat{x}_\kappa$ is the a posteriori state estimate at step k, P_k^- is the a priori estimate error covariance matrix, P_{k-1} is the a posteriori estimate error covariance matrix , K_n is the Kalman Gain matrix.

Equation (3) is the *State estimation extrapolation*, (4) is the *error covariance extrapolation*, (5) is the *Kalman Gain*, (6) is the *state estimate observational update* and (7) is the *error covariance update*. These equations are implemented in the Kalman Filter application which we will now describe.

III   THE 2D TEACHING TOOL

The Kalman Filter teaching tool is a Java application that can be used to verify the students' knowledge and allow them explore their own understanding of the effects of noise on measuring a system response.

The initial application window offers the user three paths to follow (see Figure 3):
1. A two-dimensional version with scrolling graphics and with parameters that can be varied in real-time;
2. A guided step-by-step version for inputting data that can extend to six inputs and six measurements;
3. A multi-dimensional version so that the user can input data in any order. There is an upper limit of six inputs and six measurements.

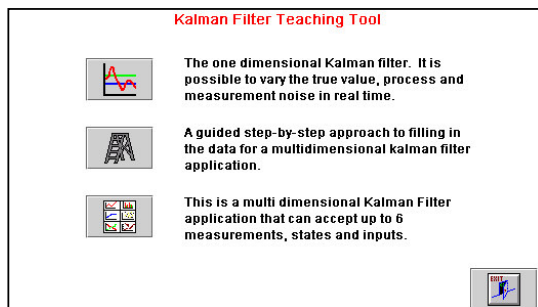We will now explore each of these versions in more detail through the use of two simple examples.

Figure 3: The initial screen view for the Kalman Filter teaching tool showing the three options : 2D, step-by-step or multiD versions.

The 2D version is limited to a single measurement of a single variable as a function of time. For example, we might want to optimally estimate the temperature of a room over time using a sensing device. The user sets the relevant parameters through an intuitive graphical user interface (see Figure 4). When all data is entered the user starts the Kalman Filter

process and data is plotted to screen. The application can be stopped and started to experiment with the tuning and transient effects of the filter settings. All parameters can be varied in real-time and the effects on the output observed simultaneously. When the output window fills with data it scrolls to accommodate new data. Help buttons are associated with each function button to provide a short explanation of each feature.

As an example we will imagine we have a bucket of nominally 10-ohm resistors, accurate to 1% RMS – see du Plessis [12]. Imagine that we select each resistor in turn from the bucket and using an ohmmeter with an accuracy of 0.3-ohms RMS random error, we randomly choose a resistor from the bucket, measure its value using the ohmmeter and attempt to estimate its true value. This example involves no dynamics and no matrix notation. The process transition assumes that all the resistors have the same value and that any variation is accounted for by random Gaussian noise. The measurement model says that we are measuring the actual state with the addition of noise. The values given are inserted into the application as follows:

Process Transition	1.0
Meas./State relation	1.0
Measurement noise variance	0.3
Process noise variance	0.1
True value	10

The Kalman Filter is then started and the plot obtained is shown in Figure 4. In this plot, we can see two transient periods. The process noise is the same in both cases. The data between the first and second transient are calculated based on the parameters given above. For the second transient, the measurement noise has been increased so that the Kalman Filter will place less trust in the measurement model and more trust in the process model.

IV   THE MULTI-DIMENSIONAL TEACHING TOOL

This version operates up to a maximum of six inputs, six states and six measurements. The actual measurements can either be simulated by the program or inserted manually by the user. There are two ways to input the Kalman Filter set-up data, by following a guided step-by-step procedure or viewing the entire Kalman Filter set-up screen at once. The choice is made from the initial screen view (see Figure 3). We will briefly describe each version before presenting a simple illustrative example.

The step-by-step version guides the user through a logical process of entering the data. Each step provides an explanation of the data being requested
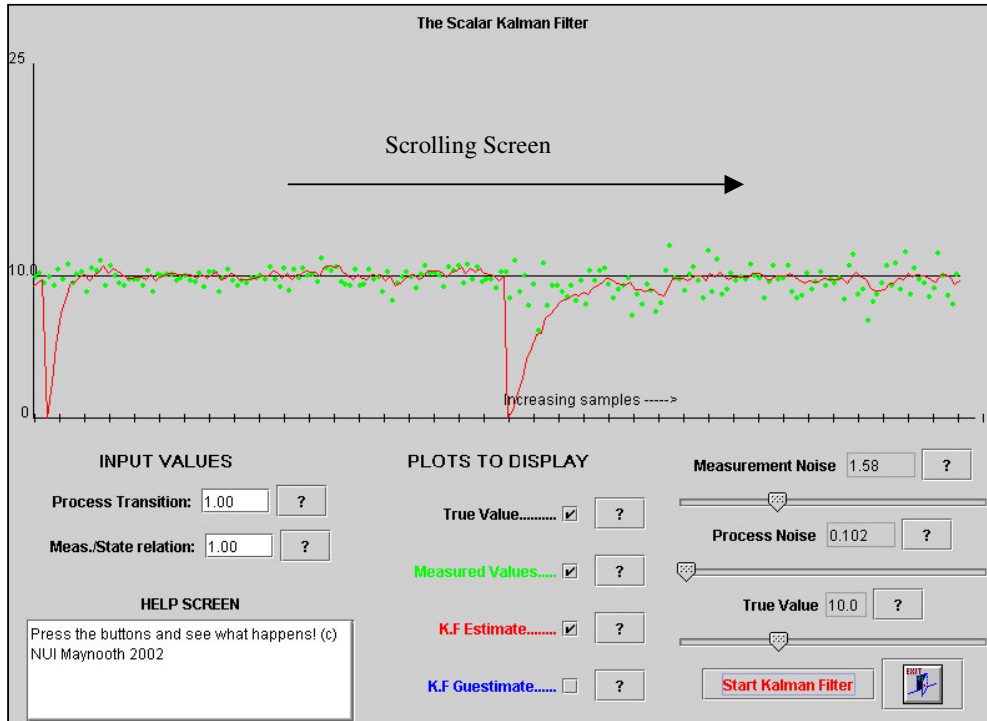
Figure 4: The 2D Kalman Filter Teaching Tool Screen Shot. All input parameters can be varied in real time to view the effects of noise on the system and the measurement. In this example the application has been stopped and restarted with new parameters. The parameters can also be varied in real time.

and outlines the theoretical prerequisites for being able to fully understand the step at hand. An example of one of these steps is shown in Figure 5. When the last step is reached the full multi-dimensional Kalman Filter tool is launched with all the values initialised to those the user inserted.
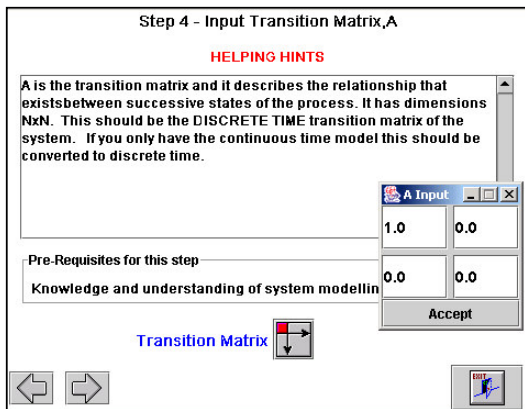

Figure 5: Step four of the step-by-step data insertion tool.

The full multiD version can also be launched by choosing the third option in the initial screen. Now the user can click any button on the screen in any order to insert the filter set-up data. The screen layout is designed to be intuitive and to reflect the actual Kalman equations. Data is entered starting from the left and working towards the right, as the dimensions of the matrices depends on the number of states, inputs and measurements. When the data is inserted the user presses the 'Start' button, the calculations are performed and the results are graphically presented to the user. Figures 6a and 6b show two examples of the main screen for entering data. In each case there is a schematic diagram to assist the understanding of the parameter currently being set. There is a hypertext help file, which also explains the Kalman Filter theory, but it is indispensable that a novice be guided in the use of the multiD version of the application.

Now we will examine a simple illustrative example based on an object moving in the X-Y plane at a constant velocity subject to random variations in its trajectory. The motion is described by a fixed motion model. The new position (X,Y) is the old position plus the velocity (ΔX, ΔY) plus noise w. At any time t we therefore have:

$$
\begin{bmatrix} X(t) \\ Y(t) \\ \Delta X(t) \\ \Delta Y(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X(t-1) \\ Y(t-1) \\ \Delta X(t-1) \\ \Delta Y(t-1) \end{bmatrix} + \begin{bmatrix} w_X \\ w_Y \\ w_{\Delta X} \\ w_{\Delta Y} \end{bmatrix} \quad (8)
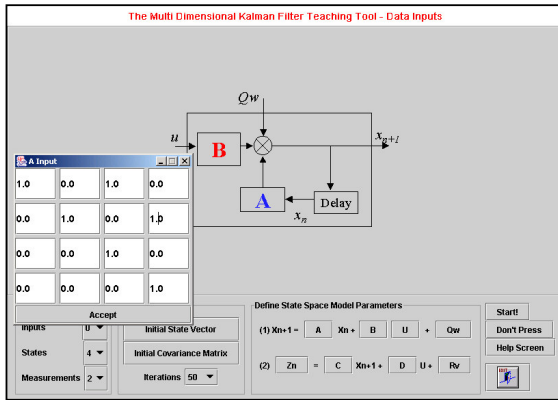$$

Figure 6a : A view of the main application data input screen. In this case the user has selected to input the transition matrix. They have already indicated that there are four states. The schematic assists in understanding the transition matrix as part of the system process.

If we assume that we can only observe the position of the object (not its velocity), then we get the following measurement model equation:

$$\begin{bmatrix} Z_X(t) \\ Z_Y(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X(t) \\ Y(t) \\ \Delta X(t) \\ \Delta Y(t) \end{bmatrix} + \begin{bmatrix} v_X \\ v_Y \end{bmatrix} \qquad (9)$$

Suppose we start out at position (0,0) moving to the right with velocity (4,3) and we sample a random trajectory of length 50. The initial covariance matrix is assumed to be a diagonal matrix with values of 1000, indicating that we do not trust our initial estimate. The measured data we use comes from a simple model of the system simulated using another application.

Figure 7a : Each state and it's covariance matrix are plotted as a function of time.

When all the measurement data is entered and the Kalman Filter parameters are set up, the process should be started by selecting the START button.
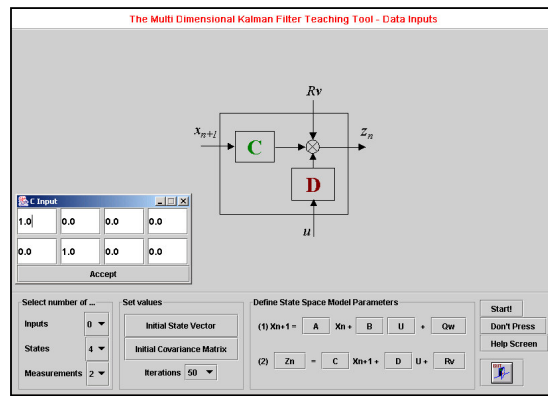
Figure 6b : Another view of the main application data input screen. In this case the user has selected to input the measurement matrix. They have already indicated that there are two measurements and four states. The schematic assists in understanding the measurement model.

This produces a number of graphs as output. The output graphs for the input data used are shown in Figures 7a and 7b.

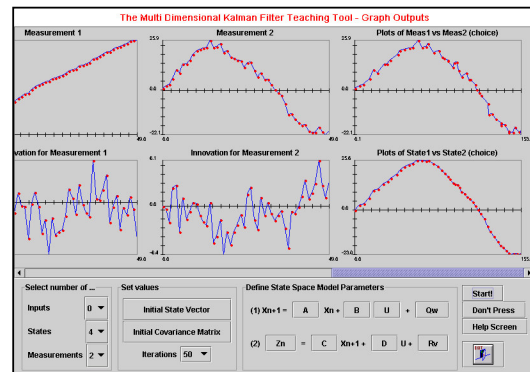Figure 7b : Each measurement is plotted together with its innovation.
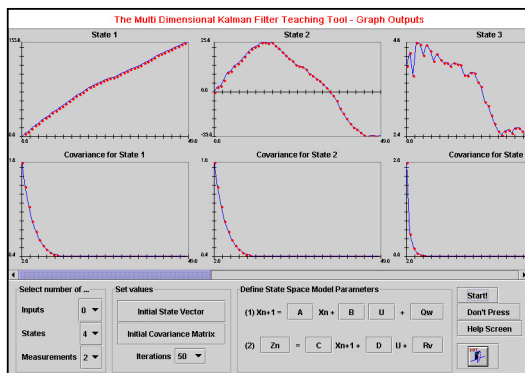
These plots indicate the action of the Kalman Filter. We are estimating the object's position and velocity as a function of time, given its position only. Our input model describes the noise associated with the measurements and with our model of the actual system. As we collect more data we combine these uncertainties to obtain a maximum likelihood estimate of the object position and velocity. The covariances associated with the state estimates then decreases exponentially as can be seen in Figure 7a. In Figure 7b we can see a plot of the raw measurements (input X vs input Y) and the estimated state (State 1 vs State 2). The estimated state plot is a smoothed estimate of the raw measurements plot.

V   CONCLUDING REMARKS

This paper described the theory and operation of the Kalman Filter. It then detailed an innovative Java-based Kalman Filter tool that operates in two dimensional or multi-dimensional mode. Such computer-assisted learning applications are essential to allow students experiment with 'what if' scenarios in the domain of signals and systems and thus deepen their awareness of fundamental engineering concepts.

Our future work will extend the current application to include signal filtering and prediction and will then look at the extended Kalman filter for non-linear systems.

REFERENCES

[1]   J. Michel, Another Look at Engineering Education: From Information to Knowledge, Australasian J. of Engng. Educ., Vol. 6, No. 2, 1995

[2]   M. C. Duffy, The Changing Nature of Engineering, Engineering Science and Education Journal, Vol. 5 , Issue 5, pp. 231-239, October 1996.

[3]   R. Kline, World War II: A watershed in Electrical Engineering Education, IEEE Technology and Society Magazine, Vol. 13, Issue 2, pp17-23, Summer 1994.

[4]   F. E. Terman, A Brief History of Electrical Engineering Education, Proceedings of IEEE, Vol. 64, No. 9, pp. 1399-1406, 1976 (also: Proceedings of IEEE, Vol. 86, No. 8, August 1998).

[5]   K. D. Stephan, All this and Engineering Too: History of Accreditation Requirements for Non-Technical Curriculum Content in U.S. Engineering Education 1933-2000, Proceedings of International Symposium on Technology and Society, pp. 155-161 Stamford, CT, USA, July 6th - 7th 2001.

[6]   E Ibrahim and R. Cockrum, Engineering Education in the Era of Global Markets, Proceedings of the Twenty-Third Annual Frontiers of Education Conference, pp. 291-296, Washington DC, 1993.

[7]   S. Rothberg, F. Lamb and A. Wallace, Computer Assisted Learning in Engineering Degree Programmes: A Survey at the End of the 20th Century, Int. J. Engng Ed. Vol. 17, No. 6, pp502-511, 2001.

[8]   E. A. Lee and D. G. Messerschmitt, Engineering an Education for the Future, IEEE Computer, January 1998.

[9]   D. McGrath, The Bologna Declaration and Engineering Education in Europe, Universities and the Bologna Declaration - The Strategy of Changes, Proceedings of the Conference held in Brno, Czech Republic, 2-3 November 2000, Edited by J.Fukac, J. Kazelle, A.Mizerova, Brno University of Technology - VUTIUM Press, Brno 2001.

[10]   M. S. Grewal and A. P. Andrews, Kalman Filtering Theory and Practice, ISBN 013211335X, Prentice Hall, 1993.

[11]   G. Bishop and G. Welch: "An Introduction to the Kalman Filter", SIGGRAPH 2001 short course.

[12]   R. M. du Plessis, Poor Man's Explanation of Kalman Filtering or How I stopped Worrying and Learned to Love Matrix Inversion, Autonetics Division of North American Rockwell Corporation, June 1967.