The Kalman Filter Learning Tool

Greg Welch, client

January 25, 2001

Chris Riley, Director
Tom Bodenheimer, Producer

Erin Parker, Admin. Leader
John Carpenter, Librarian
1 Overall Goal

The Kalman filter is a set of mathematical equations that provides an efficient computational estimate of the state of a process (e.g., the position and orientation of an airplane) given a time-varying sequence of noisy measurements (e.g., air speed, pressure, temperature, engine thrust). The filter is a popular mathematical estimator due to its efficiency and robustness.

Our goal is to develop a web-based tool to help develop the intuition and insight of novice users regarding the behavior of the Kalman filter. Users would have the ability to change various input parameters and then see how the Kalman filter responds for a given set of noisy measurements. Because the software is to be primarily used as a teaching aid, we have several design goals. First, the interface should be simple, elegant, and easy to use. Input choices will be limited as necessary to avoid overwhelming the user. Parameters will be restricted to realistic but interesting values. A single, concrete physical example (e.g., sensing water level in a tank) will be used as well for all simulations. Performing multiple simulations using different physical models can lead to confusion for a novice user. Second, the output from a simulation should be organized and displayed in a manner that is easy to understand. Users should be able to see a (graphical) overview of the filter’s estimation of the process state and how it compares to the actual state. Users should also be able to step through the measurement data and the equations that comprise the Kalman filter and trace the intermediate steps of the algorithm. This helps in understanding the inner workings of the Kalman filter. Finally, the resulting software should be easy to extend, modify, and maintain. The design of the software should not preclude for example the use of different physical examples or the addition of input choices and parameters.

2 Product Features (Rank)

- One concrete, 1-D example that is extendable for all model choices/options; sensing water level in a tank is a possibility (10).

- Input
  - *Truth* options: constant, periodic, and an ‘in-between’ (10).
  - *Dynamic model* options that match truth model options (10).
  - More model options; i.e., expand ‘in-between’ to slow and sloshy (5).
  - *Linear measurement model*; wires on a stick if water tank example (10).
  - *Non-linear measurement model*; camera observing stick if water tank example (5).
  - *Process noise* knob/slider limited to realistic values (2).
  - *Measurement noise* knob/slider limited to realistic values (10).
  - *Measurement rate* fixed to an appropriate constant (10).

- Output
  - Show *truth* for each simulation (10).
  - Show *Kalman Filter estimate* for each simulation (10).
  - Show *error signal* for each simulation (5).
  - Show *measurements* for each simulation (5).
  - Print *overall stats* for each simulation; RMS error, peak error, etc. (9).
Stack/overlay results of a limited number of simulations (7).

- **Processing**
  - Normal mode of simulation; run filter, show results (10).
  - Step mode of simulation; step through data and all equations (4).

- **Miscellaneous**
  - Show pictures of example’s physical setup (4).
  - Show diagram with equations, if in step mode (4).
  - Show graphical highlight of steps through equations (4).

### 3 Interface Thoughts

One of the challenges of this product is to manage the complexity of the Kalman filter. There are a tremendous number of free variables that must be defined to describe an example use of a Kalman filter. Our goal is fix just enough of these variables so that a student will not be overwhelmed, but still be able to perform a meaningful experiment.

Since we will only be modeling a single one-dimensional example, possible user choices for the true motion of the process will be constant, slow (linear), and sinusoidal. The user will then be able to choose whether to model this process with a constant, slow, or sinusoidal model. The user will also be able to enter the measurement and process noise using some continuous indicator such as a slider bar. The measurement rate will be fixed to some appropriate constant.

The format of the output has many options, some of which will depend on the final choices for input parameters. In the batch mode, at a very minimum there will need to be some graphs showing the true state of the system and the Kalman filter’s estimate of that state over time. It might also be useful to show the measured value of the state over time. More difficult questions will be how to show the noise in the true state, the measurement, and that assumed by the Kalman filter to make its predictions. One method we considered for displaying the measurement noise was by varying the thickness of the line that described the measured value. One concern about this approach was how difficult it would be given the graphics capabilities of Java.

A major use of this tutorial will be to allow the user to run several, similar experiments and observe how the filter behaves differently on each run. A major issue then is how to simultaneously display this data for comparison. We will assume that a user’s screen is no bigger than 15 inches. As a result, the windows may be small enough that overlaying the results of more than one set of parameters look too cluttered on screen. An alternative option would be to have a separate window and graph for each run. The best solution will likely be clearer once we have determined more specifics about the example and the corresponding differences in output for the various input parameters we will allow.

### 4 Package Options

#### 4.1 Version 1.0

We have divided the project into four modules and assigned one to each of the team members. Each team member is certainly not responsible for doing all of the work required for each module, but is in charge of leading the team in each effort. Below is the list of modules, with team member assignments.
User Interface. Main control window, including parameter fields/sliders and overall layout of the application (Chris).

Kalman Filter Engine. Implementation of the Kalman filter equations (John).

Plotter. Graphical plotting tool (Tom).

Equation Display and Stepper. Interface for displaying diagrams and equations, highlighting and stepping as necessary (Erin).

Quick development of any or all modules may be accomplished by exploiting existing tools that fit into our design.

In version 1.0, we include all features ranked 9-10: limited truth and dynamic model options, a linear measurement model, a measurement noise slider, a measurement rate constant, truth and Kalman filter estimate outputs, overall simulation stats, and a normal mode of simulation. As one can see, version 1.0 is a minimal implementation of the project.

Show stoppers. Failure to find/obtain existing tools for interfacing, plotting, etc. will considerably slow our progress on the project, but will not block development. Team members are prepared to code such tools from scratch, but certainly it will preclude the development of additional features.

Some apprehension exists over the feasibility that all features of the application can be executed within the constraints of a Java applet. However, all team members will proceed with this in mind.

Things to learn. All team members and the project client need to solidify the design and layout of the user interface. Also, the team and client must agree on an appropriate concrete example.

Things to procure. Chris and Erin (user interface and equation display/stepper modules) are working to find and obtain a GUI development tool. Tom (plotter module) is searching for a generic graphing tool for Java. John (Kalman filter module) is looking for existing Kalman filter implementations.

Things to understand. All team members could benefit from a better understanding of the Kalman filter. In fact, Danette Allen (a local expert on the subject) has offered to give the team a Matlab demonstration/tutorial of the Kalman filter.

4.2 Version 2.0

If development of version 1.0 is smooth and quick, version 2.0 will be supplemented with the features of rank 5-8: more options/settings for the models, a non-linear measurement model, output of error signal and measurements for each simulation, and an overlay of the results of several simulations.

4.3 Version 3.0

Finally, features of rank 1-4 will be added in version 3.0, if there is enough time: a process noise slider, a picture of the example’s physical setup, a step mode for simulation, and equation diagrams with highlighting.

5 Schedules

We intend to have client meetings on Mondays from 10:45AM to 11:30AM, as needed. Our weekly team meeting is on Wednesdays at 11:00AM, supplemented by additional team meetings as necessary.