COMP 145 UNC-Chapel Hill

Design Specification

KALMAN FILTER ON-LINE LEARNING TOOL

Submitted to
Prof. Greg Welch

February 20, 2001

_________________________  Chris Riley, Director
_________________________  Tom Bodenheimer, Producer
_________________________  Erin Parker, Admin. Leader
_________________________  John Carpenter, Librarian
PREFACE
This document is a formal description of the design that the Kalman Filter Team will use in order to meet its contract obligations. It is intended for the benefit of the team and the boss Greg Welch, as distinguished from the client, Greg Welch.

1 Introduction
This design is based on several weeks’ collaboration amongst the team and with the boss, Greg Welch. No prototypes have been completed, nor is this a revision of a previous design.

2 High-Level Design Specification
2.1 Structural Model
The Kalman Filter Learning Tool consists of the modules shown in Figure 1: a user interface, a Kalman filter engine, a data repository, and a plotting tool. The model is an example of the Façade pattern described in “Design Patterns: Elements of Reusable Object-Oriented Software” by Gamma, Helm, Johnson, and Vlissides. The user interface separates the user from the inner workings of the learning tool, but more importantly, the Façade pattern separates the subsystems from each other. The user interface knows about (i.e., has references to) the other modules, but the remaining subsystems do not know about each other or the user interface.

2.1-A User Interface
The user interface sets the parameters for the Kalman filter simulation and returns Kalman filter state and error estimates to the user in the form of plots and text. The user interface delegates these tasks to the other subsystems (Kalman filter engine, data repository, and plotting tool).

2.1-B Kalman Filter Engine
The Kalman filter engine implements the Kalman filter equations and is responsible for computing both single and multiple iterations of the filter.

2.1-C Data Repository
The data repository stores a record of the filter’s iterations and the values computed.

2.1-D Plotting Tool
The plotting tool generates plots of the state and error estimates produced by the filter. The user interface is responsible for displaying these plots.
3 Detailed Design Specification
3.1 Object Models

**User Interface**

- `select_Truth()` – defaults to constant, radio button selectable options of filling and/or sloshing
- `select_Model()` – defaults to constant, radio button selectable options of filling and/or sloshing
- `select_Measurement_Model()` – radio button selection of linear OR non-linear
- `set_Process_Noise()` – radio button selection of one of three values (.01, 1, or 100)
- `simulate()` – top level method to run and display result with a set of selected parameters
- `do_calculations(KFengine)` – pass the selected parameters to the KFengine
- `get_Plots(Plotter)` – request the plot frames from the plotter
- `display_Plots()` – pops up window to display plots
- `close_Plots_Window()` – close pop-up plot window
- `stepper()` – top level method to step through values generated by KFengine
- `get_KF_Values(DataRepository)` – get the values of the filter at time t from the DataRepository
- `display_Stepper()` – displays step window
- `close_Stepper_Window()` – close pop-up stepper
- `step()` – button allows user to step to value at time t
- `new_Simulation()` – button resets parameters
- `show_Hyperlink()` – all user selectable options are hyperlinked to explanatory pages, so show page

Figure 2: *User Interface* Object Model Diagram.
3.2 Data Flow Diagram
The data flow diagram from Contract II is basically sufficient for our needs, internally. However, the diagram has been updated to include the action of a user selecting hyperlinks in the user interface (these hyperlinks direct the user to an HTML “help” page explaining various controls, settings, formulas, etc.)
4 Requirements Traceability
Detailed Design modules are listed down the rows on the left, and the System-Level Requirements from Contract II (with hyperlink and data-dump requirements added, see Contract III). An ‘x’ in a cell indicates a connection between the two.
5 Schedule

The following is an updated chart of the schedule (formerly our PERT Chart). It has been modified to include several milestones and emphasize work on the sub-systems that make up our project.

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Meet with client</td>
<td>1 day</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Read “An Introduction to the Kalman Filter”</td>
<td>2 days</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Write ‘Preliminary Report’</td>
<td>2 days</td>
<td>1,2</td>
</tr>
<tr>
<td>4</td>
<td>Choose/procure Java IDE</td>
<td>3 days</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Learn the Java IDE</td>
<td>5 days</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Write ‘Contract I/Schedule I’</td>
<td>6 days</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>‘Contract I’ delivered to boss and client</td>
<td>milestone</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>Choose/procure a plotting tool **</td>
<td>3 days</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Learn the plotting tool **</td>
<td>5 days</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>Develop ‘Prototype’</td>
<td>10 days</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>Client approves/disapproves of ‘Prototype’</td>
<td>milestone</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>Write ‘Contract II/Schedule II’</td>
<td>4 days</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>Client signs ‘Contract II’</td>
<td>milestone</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>Write ‘Design Specification’</td>
<td>5 days</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>‘Design Specification’ delivered to boss</td>
<td>milestone</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>Write ‘User Manual’</td>
<td>7 days</td>
<td>15</td>
</tr>
<tr>
<td>17</td>
<td>‘User Manual’ delivered to boss</td>
<td>milestone</td>
<td>16</td>
</tr>
<tr>
<td>18</td>
<td>Prepare for ‘Group Status Report’</td>
<td>5 days</td>
<td>11</td>
</tr>
<tr>
<td>19</td>
<td>Write ‘Implementation Manual’</td>
<td>7 days</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td>‘Implementation Manual’ delivered to boss</td>
<td>milestone</td>
<td>19</td>
</tr>
<tr>
<td>21</td>
<td>Complete the KF engine (John) **</td>
<td>14 days</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td>Team members certify KF engine **</td>
<td>milestone</td>
<td>21</td>
</tr>
<tr>
<td>23</td>
<td>Integrate engine and plotter **</td>
<td>7 days</td>
<td>9,22</td>
</tr>
<tr>
<td>24</td>
<td>Complete the user interface (Chris) **</td>
<td>14 days</td>
<td>5,14</td>
</tr>
<tr>
<td>25</td>
<td>Team members certify UI **</td>
<td>milestone</td>
<td>24</td>
</tr>
<tr>
<td>26</td>
<td>Integrate UI and engine **</td>
<td>7 days</td>
<td>21,25</td>
</tr>
<tr>
<td>27</td>
<td>Integrate UI and plotter **</td>
<td>7 days</td>
<td>25</td>
</tr>
<tr>
<td>28</td>
<td>Complete data repository (Erin) **</td>
<td>7 days</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Team members certify data repository **</td>
<td>milestone</td>
<td>28</td>
</tr>
<tr>
<td>30</td>
<td>Integrate data repository to UI/engine/plotter **</td>
<td>7 days</td>
<td>26,27,29</td>
</tr>
<tr>
<td>31</td>
<td>Team members certify integration of sub-systems **</td>
<td>milestone</td>
<td>30</td>
</tr>
<tr>
<td>32</td>
<td>Test and validate KF On-line Tutorial</td>
<td>10 days</td>
<td>31</td>
</tr>
</tbody>
</table>
5.1 Development Milestones
The milestones of the schedule are the following. (Milestones pertaining to the certification of sub-systems are denoted by **.)

- ‘Contract I’ delivered to boss and client
- Client approves/disapproves of ‘Prototype’
- Client signs ‘Contract II’
- ‘Design Specification’ delivered to boss
- ‘User Manual’ delivered to boss
- ‘Implementation Manual’ delivered to boss
- Team members certify KF engine **
  Team members agree that the Kalman filter engine module performs the functionality required of it (for all possible combinations of truth and model options, for both float and angular cases).
- Team members certify UI **
  Team members agree that the user interface module contains all of the required buttons and panels, and that every action from the user produces the correct reaction by the module. (Also the last chance to make certain that the UI is appropriate for the KF On-line Learning Tool and user-friendly.)
- Team members certify data repository **
  Team members agree that the data repository manages the correct data (especially making certain that the variables are of the correct dimensionality for each case), and responds to inquiries from the main module to deposit/recover data.
- Team members certify integration of sub-systems **
  Team members agree that each module has been integrated appropriately and that the interaction among sub-systems is correct (e.g., does the UI correctly call plotter with data produced from KF engine?).
- KF Tool passes test cases (matches Matlab)

5.2 Gantt Chart
The Gantt Chart that corresponds to the chart above, is an attachment.

5.3 Risk Analysis and Management
Most risks in the schedule are analyzed in ‘Contract II/Schedule II’. The following emphasizes the risks associated with the sub-systems.

5.3.1 Identified Schedule Risks
Each of the sub-systems (user interface, KF engine, data repository, plotting tool) is crucial to the Kalman Filter On-line Learning Tool. The plotting tool is pre-existing software the team will procure, while the other sub-systems will be original software. The user interface is particularly important, as it controls the other modules. Any slip in the completion of a sub-system endangers the completion of the entire system.

5.3.2 Plans
The following are plans for each sub-system that address possible slips in the schedule.

- **User Interface**: Although the completion of the UI (id 24) is on the schedule’s critical path, we anticipate working out design details while working on the prototype (id 10). The prototype development gives team members and client a chance to make certain that all of the required buttons, labels, etc. are included and their functionality is well defined. Then, completion of the UI requires only the implementation these features.

- **Kalman Filter Engine**: Completion of the KF engine (id 21) actually has a little bit of slack in the schedule before affecting completion of the overall project. In addition, implementation of the engine benefits from having the client’s Matlab version as a model.

- **Data Repository**: Completion of the data repository also has quite a bit of slack in the schedule before affecting the completion of the overall project.

- **Plotting Tool**: Procurement and familiarization of the plotting tool is scheduled early, in case an appropriate tool can not be procured. There is enough time in the schedule for the team to implement their own plotting tool, if necessary.