

# 稳态 Kalman 滤波及信息融合算法用户界面开发

——《最优估计与系统建模》课程大作业

学院：微电子学院

姓名：胡翰彬

学号：1132109003

邮箱：huhanbinnew@hotmail.com

## 摘要

Kalman 滤波以及多传感器信息融合技术在工程领域有着广泛的应用，但是没有比较理想方便快捷针对性强的工具支持相应的工作。本研究在 Matlab 中成功实现了稳态 Kalman 滤波及信息融合算法的实现，并且完成了相应友善的图形用户交互界面的开发，能够方便快捷地针对不同规模维数的稳态 Kalman 滤波器、预报器以及多传感器融合算法进行仿真模拟。总计编写 16 个文件，共 1142 行 Matlab 代码。研究过程中发现了教材<sup>[1]</sup>中的一处错误，并对不同情况的仿真实例进行了测试，得到了理想的结果。此工具可用于教学、研究、实际使用等多种可能。

## 关键词：

Kalman 滤波，信息融合，Matlab GUI 界面

## 1. 研究目的与意义

稳态 Kalman 滤波以及多传感器信息融合技术在工程领域有着广泛的应用，但是没有比较理想方便快捷针对性强的工具支持相应的工作。上述的两种技术的使用均需要比较复杂的数学理论，这给实际的工程应用带来了学习与使用以及推广上的麻烦。图形用户交互界面（Graphical User Interface, GUI）可以很好地解决这些问题，友善的界面将给使用者带来很多便利，并方便 Kalman 滤波及信息融合技术等的学习。

## 2. 软件架构与开发流程

### 2.1. Matlab 控件介绍

本研究为作者第一次编写 Matlab GUI 界面，故对相应的 Matlab 控件进行了学习，下面介绍主要使用的一些控件，及其主要的使用方法。

#### 2.1.1. Static Text 控件

Static Text 控件为静态文本框，如图 1 左侧所显示的“Time Step:”，即为该控件。

图 1 Static Text 与 Edit 控件示例

可使用以下代码对控件进行配置。

```
01  UIControl(...
02      'Parent', hpanel, ...
03      'Units', 'pixels', ...
04      'Position', [370, 530, 100, 20], ...
05      'BackgroundColor', get(hpanel, 'BackgroundColor'), ...
06      'ForegroundColor', 'w', ...
07      'HorizontalAlignment', 'center', ...
08      'FontSize', 11, ...
09      'FontWeight', 'bold', ...
10      'Style', 'text', ...
11      'String', 'Time Step:');
```

其中“Position”属性用于设置控件大小和位置，“BackgroundColor”和“ForegroundColor”分别用于设置背景色和前景色。“Style”中指定了此控件的类型为 Static Text 框，而“String”中填入了所需要呈现的文字。

### 2.1.2. Edit 控件

Edit 控件用于输入文本信息，如图一中右侧所显示的编辑栏。可通过如下代码对编辑栏进行配置。

```
01  nstep_edit = UIControl(...
02      'Parent', hpanel, ...
03      'Units', 'pixels', ...
04      'Position', [470, 530, 40, 20], ...
05      'Style', 'edit', ...
06      'FontSize', 11, ...
07      'String', '100');
```

其中，nstep\_edit 保存了该编辑框的句柄。若希望获取该编辑框中的具体数值，可使用如下语句：

```
01  nstep = str2double(get(nstep_edit, 'String'));
```

这里 get 命令用于获取相应句柄中的“String”属性值，并通过 str2double 函数将结果转换为数值。

### 2.1.3. UiTable 控件

UiTable 控件用于实现表格输入与展示的功能，如图 2 所示。

	1	2	3
1	1	0	0
2	0	1	0
3	0	0	1

图 2 UiTable 控件示例

本控件可通过以下代码控制。

```
01 fai_table=uitable(...
02     'Parent',hpanel,...
03     'Units','pixels',...
04     'Position',[170,340,200,160],...
05     'ColumnWidth',{35},...
06     'data',eye(NDIM_INIT),...
07     'FontSize',11,...
08     'columneditable',true,...
09     'RearrangeableColumns','off');
```

可以看到其中“ColumnWidth”设置了表格宽度，“data”设置表格中的数据项，“columneditable”属性用于控制表格是否可以输入修改。若希望提取表格中的数据可通过如下语句实现：

```
01 fai=get(fai_table,'data');
```

若希望更改表格中的数据，如将表格变为  $n$  维的单位阵，可以使用如下语句：

```
01 set(fai_table,'data',eye(n));
```

#### 2.1.4. Pushbutton 控件

Pushbutton 即为我们常见的按钮，如下图 3 所示。

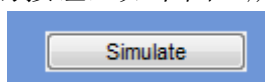


图 3 Pushbutton 控件示例

配置代码如下：

```
01 uicontrol(...
02     'Parent',hpanel,...
03     'Units','pixels',...
04     'Position',[680,10,100,20],...
05     'Style','pushbutton',...
06     'Callback',{@Simulate},...
07     'String','Simulate');
```

其中“Callback”属性调用了 Simulate 函数的句柄以在单击后调用 Simulate 函数执行对应指令。

#### 2.1.5. UitablePanel 控件

这是一个在网上介绍的实现标签页功能的控件。<sup>[3]</sup> 由于总体功能设计可分为三个标签，故如下段代码所示，其中定义了相应的三个标签的名称，并且调用函数 `CreateTab`，并由 `drawtab` 一系列 3 个函数生成各个标签页的 GUI 界面，其中 `NDIM_INIT` 定义了初始矩阵维数，而 `seed` 为随机数种子的值。

```
01 htab = uitabpanel(...
02     'Parent',hfig,...
03     'TitlePosition','centertop',...
04     'Position',[0,0,1,1],...
05     'PanelBorderType','line',...
06     'Title',{'Steady State','Information Fusion','About'},...
07     'CreateFcn',@CreateTab);
08
09 function CreateTab(htab,evdt,hpanel,hstatus)
10     NDIM_INIT = 3;
11     seed=[14 8 4 5 6];
12     drawtab1(hpanel(1),NDIM_INIT,seed);
13     drawtab2(hpanel(2),NDIM_INIT,seed);
14     drawtab3(hpanel(3));
15 end
```

## 2.2. 软件整体架构

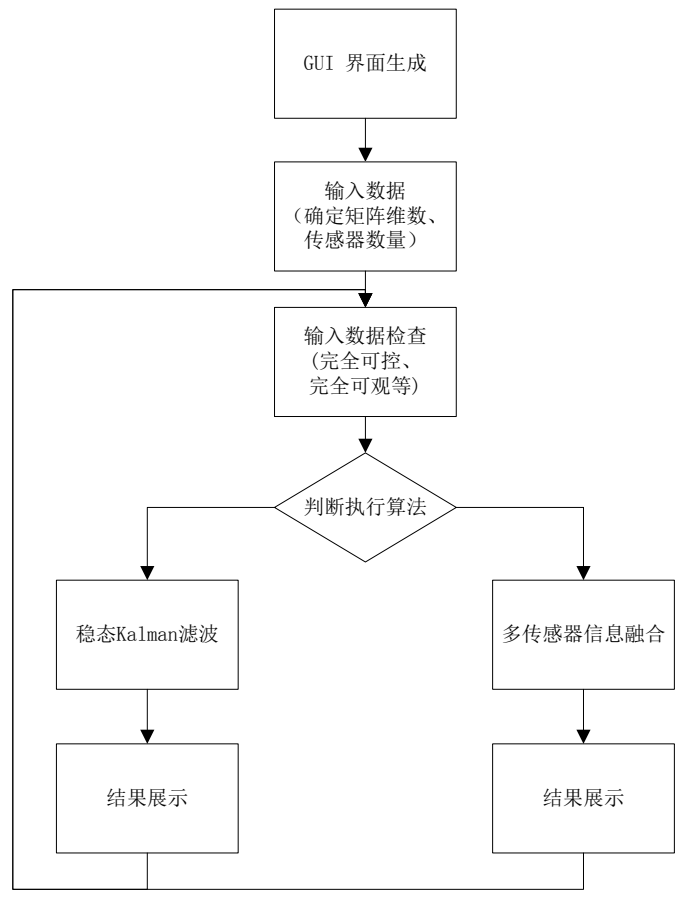


图 4 软件总体流程

上图 4 为软件实现总体流程，首先通过程序预先设置生成各个标签栏的 GUI 界面，然后根据用户所在标签页的输入数据进行错误检查，包括各种可能的输入错误，如输入了字母、时间步数输入了非正整数等。同时还会检查相应的系统状态方程的可控制性及观测方程的可观测性。如发现错误，会根据错误的原因给出相应的提示框，提醒用户改正错误，如图 5 所示。

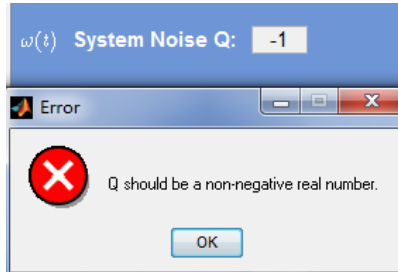


图 5 输入数据检查

在确保所有输入数据正确后，根据目前所在的标签页，运行相应的算法实现，并给出对应的结果输出。

## 2.3. 算法实现介绍

### 2.3.1. 稳态 Kalman 滤波实现

下面是稳态 Kalman 滤波的外层实现代码。

```
01 x = system_model(fai, gama, Q, nstep, seed(1));
02 y = sensor_model(x, H, R, nstep, seed(2));
03 [xigema, kf, kp, pusaif, pusaip, xjian, xxjian] = sskf(fai, gama, H, Q, R, y)
04 xxNjian = N_prediction(fai, xxjian, np);
05 set(xigema_table, 'data', xigema);
06 set(pusaif_table, 'data', pusaif);
07 set(pusaip_table, 'data', pusaip);
08 set(kf_table, 'data', kf);
09 set(kp_table, 'data', kp);
10 plot_all_x_sskf(x, xjian, xxNjian, np, nstep);
```

其中首先通过 `system_model` 函数模拟系统状态方程得到系统状态变化，然后根据所得到的系统状态利用 `sensor_model` 模拟传感器对状态进行观测。然后仿照教材<sup>[1]</sup>中的附录 1，编写 `sskf` 函数模拟 Ricatti 方程实现稳态 Kalman 滤波及预报功能，并使用 `N_prediction` 函数产生 N 步预测的结果。5-10 行均用于数据展示。

### 2.3.2. 多传感器信息融合实现

下面是多传感器信息融合的实现过程。

```
01 x = system_model(fai, gama, Q, nstep, seed(1));
02 for i = 1:ns
03     y(i, :) = sensor_model(x, H(i, :), R(i), nstep, seed(i+1));
04     [xigema, kf(:, i), kp, pusaif(:, :, i), pusaip, xjian(:, :, i), xxjian]...
05         = sskf(fai, gama, H(i, :), Q, R(i), y(i, 1:nstep));
06 end
07 HC = []; RC = eye(n); YC = y;
08 for i = 1:ns
```

```

09     HC = [HC; H(i,:)];
10     RC(i,i) = R(i);
11 end
12 [xigemaC, kfC, kpC, pusaifC, pusaipC, xjianC, xxjianC]...
13     = sskf(fai, gama, HC, Q, RC, YC);
14 [Ps, PsC] = cal_Ps(ns, nstep, fai, gama, Q, H, R,...
15     pusaif, kf, HC, RC, pusaifC, kfC);
16 for i = 1:ns
17     TrP(i) = trace(Ps(:,:,i,i));
18 end
19 TrPC = trace(PsC);
20 [AS, AD, AM, TrS, TrD, TrM] = calWeight(ns, Ps);
21 [xS, xD, xM] = fusion(ns, nstep, xjian, AS, AD, AM);
22 plot_all_x_infF(ns, nstep, x, xjian, xjianC, xS, xD, xM);
23 [EF, ESF, EDF, EMF, ECF] = calError(ns, nstep, x, xjian,...
24     xjianC, xS, xD, xM);
25 msgError(ns, nstep, TrP, TrPC, TrS, TrD, TrM, EF, ECF,...
26     ESF, EDF, EMF);

```

首先 1-6 行实现了系统状态的生成以及各个传感器的观测信号模拟，并对其进行了 Kalman 滤波。7-13 行使用集中式方法对信号进行了估计。14-15 行根据 Lyapunov 计算了各项稳态误差方差阵。16-19 行通过对误差方差阵求迹算得理论精确度。20 行利用教材<sup>[1]</sup>中附录 2 的方法对三种加权方法的权重进行计算。21 行使用得到的权重采用 3 种不同的加权方法(矩阵加权, 对角阵加权, 标量加权)对信息进行融合。22 行绘制相应曲线。23-26 计算了实验仿真得到的当前步数情况的精确度，并输出与理论精确度比较。

### 3. 软件使用方法与测试

#### 3.1. 软件使用方法

本软件在 Matlab 2012a 下编写，未经过其他 Matlab 版本测试，请最好在 Matlab 2012a 或更高 Matlab 版本下运行。

本软件共包含 17 个文件，其中除去 uitabpanel.m<sup>[3]</sup> 之外的 16 个文件均为自行编写得到的结果，共 1142 行代码。

程序运行方法如下所示：

- a. 打开 Matlab，切换文件夹至软件所在目录。
- b. 在 Matlab 命令行中输入 gui\_kalman，弹出如图 6 窗口。

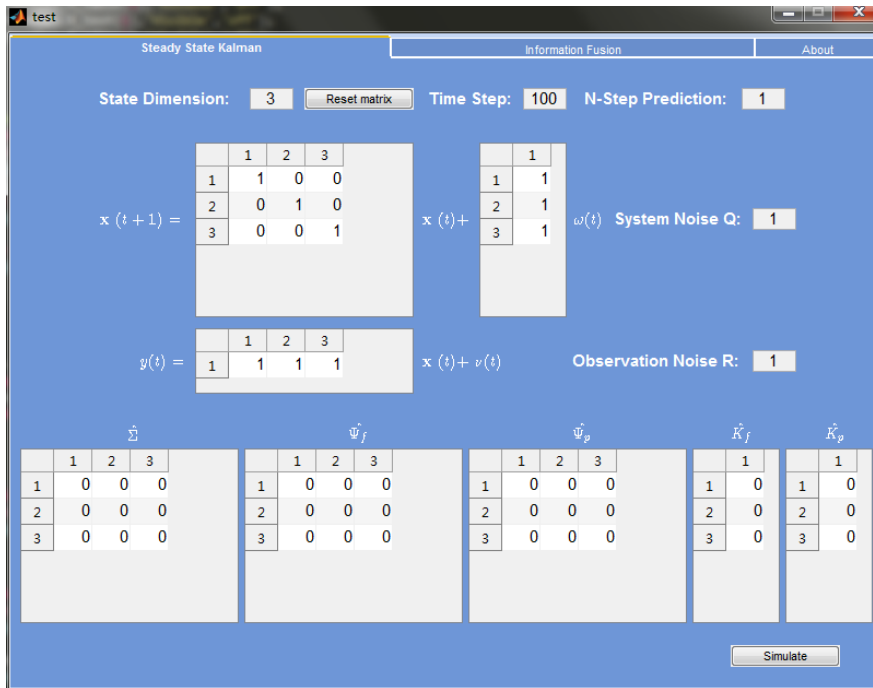


图 6 软件界面展示

- c. 共有三个标签页可选，“Steady State Kalman”为稳态 Kalman 滤波模块，“Information Fusion”为信息融合模块，“About”为作者相关信息。选择相应功能标签。下面以“Information Fusion”（信息融合）为例。
- d. 在“State Dimension”栏中填入系统状态维数，按下“Reset Matrix”按钮，更新各矩阵维数。如输入 2，得到下图 7 结果。注意，此栏仅允许输入正整数，如输入错误，程序会自动报警提示。

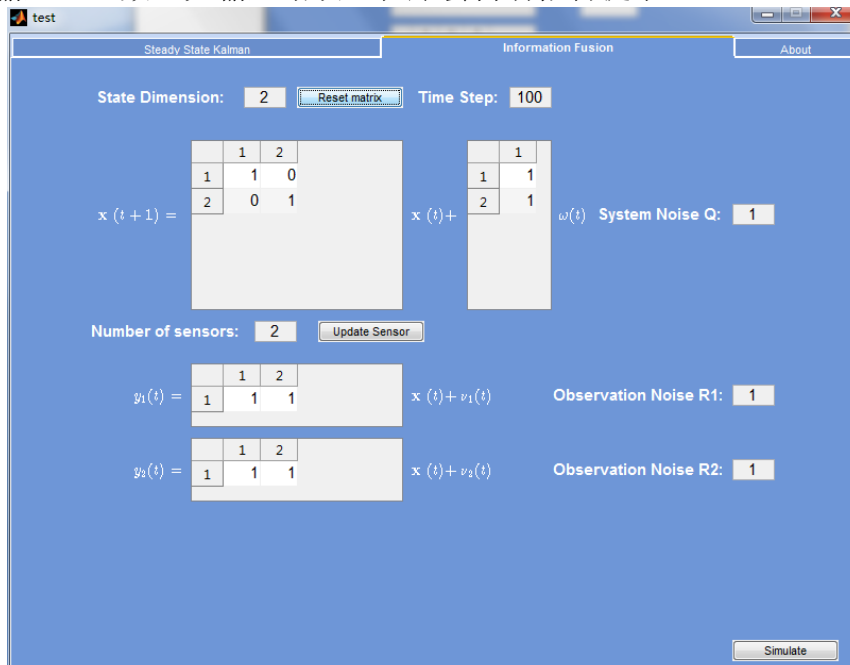


图 7 矩阵维数改变展示

- e. 在“Number of sensors”栏中输入传感器数量，按下“Update Sensor”按钮，会更新传感器输入栏个数。如输入 3，得到下图 8 结果。注意，此栏仅允许输入 2、3、4（由于软件页面大小限制），如输入错误，程序会

自动报警提示。

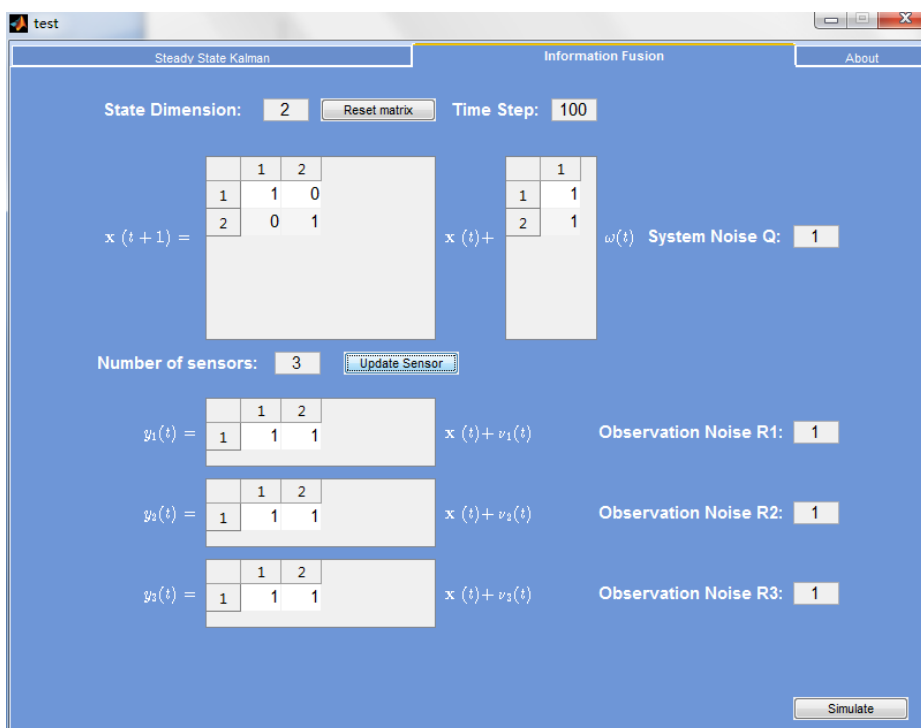


图 8 传感器个数改变展示

- f. 根据用户需求填入各个矩阵的相应数据，时间步数，白噪声方差大小等。
- g. 按下“Simulate”按钮，进行仿真，得到结果，具体结果截图由下一小节给出。所有填写栏均有输入保护功能，会自动检查是否是有效输入，否则会弹出一提示框，提示错误原因，如下图所示。

### 3.2. 软件测试结果

#### 3.2.1. 稳态 Kalman 滤波测试结果

本测试首先选用教材<sup>[1]</sup>上例 3.7.3 的数据，教材中声称此系统完全可控、可观。但在输入程序验证后发现，本系统不可控，如下图 9 所示。

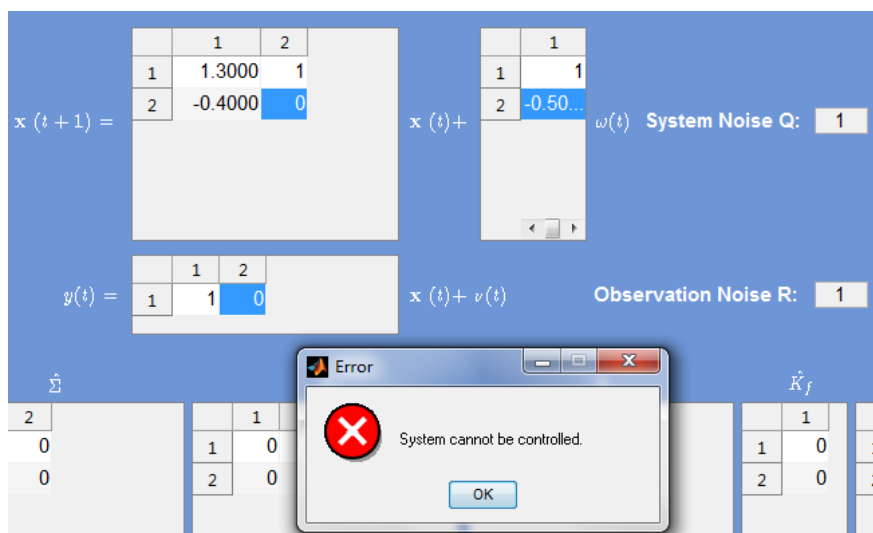


图 9 完全可控、完全可观检测功能展示

在如图设置情况下，系统不可控。故对本系统的方程进行计算发现：



$$\text{rank}[\Gamma, \Phi\Gamma] = \text{rank}\left[\begin{pmatrix} 1 \\ -0.5 \end{pmatrix}, \begin{pmatrix} 1.3 & 1 \\ -0.4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -0.5 \end{pmatrix}\right] = \text{rank}\begin{bmatrix} 1 & 0.8 \\ -0.5 & -0.4 \end{bmatrix} = 2$$

可以看到控制矩阵的秩不等于系统状态维数 2，故本系统的确不完全可控，为教材<sup>[1]</sup>中的一处错误。

下面针对教材<sup>[1]</sup>上习题 3.3 所给数据进行测试，可以得到如下配置图 10：

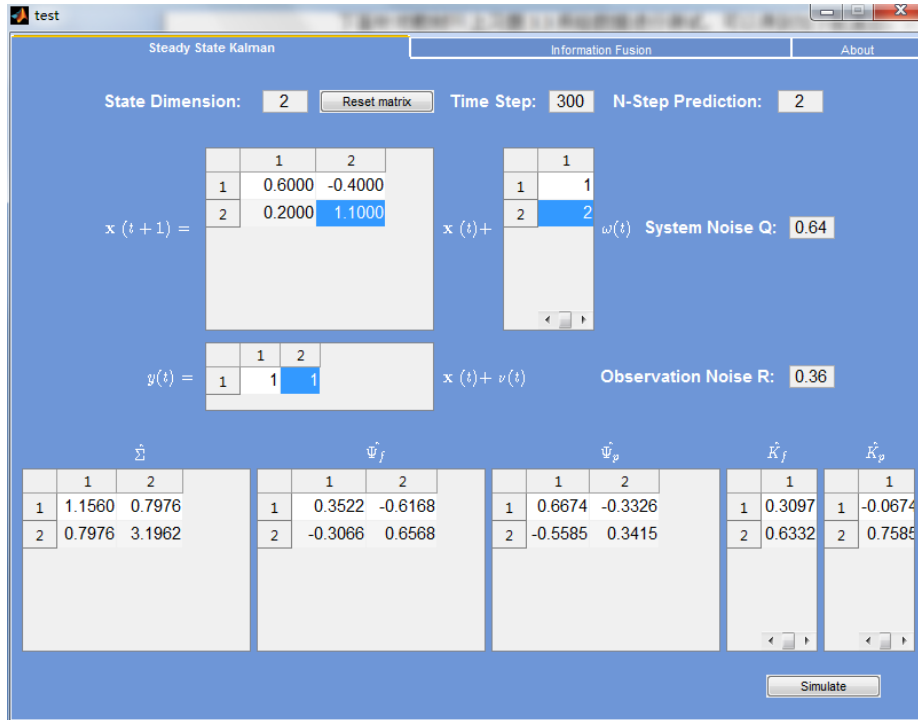


图 10 稳态 Kalman 滤波结果

可以看到在上方输入系统相关数据后，并点击“Simulate”按钮后，会在下方的矩阵中显示对应矩阵的稳态估计结果，同时也可以得到相应的滤波和预报的各个状态变量曲线，如下图 11 所示。

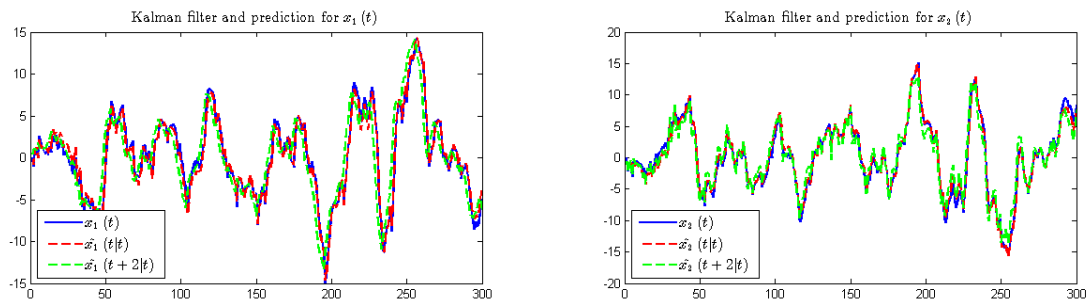


图 11 稳态 Kalman 滤波与预报曲线

仿真结果与平时作业习题结果一致。

### 3.2.2. 信息融合测试结果

本测试使用教材<sup>[1]</sup> 上例 4.4.1 的数据，得到了如下图 12 精确度对比结果：

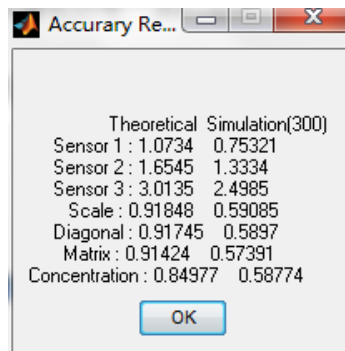


图 12 300 步步数下精确度对比

可以看到理论精确度结果与教材<sup>[1]</sup> 中表格 4.4.1 一致。可以看到 300 步数的情况下实际实验结果与理论有一定差距。若调整为 10000 步，则可得到如下结果：

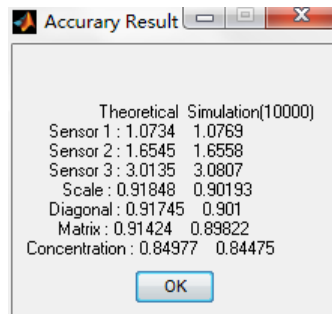


图 13 10000 步步数下精确度对比

发现图 13 中在增大步数的情况下实验仿真精确度向理论值收敛，两者基本一致。而且这也验证了集中式方法精确度优于矩阵加权，矩阵加权优于对角阵加权，对角阵加权优于标量加权，标量加权优于各个传感器得到的数据的结论。<sup>[1,2]</sup> 同时可以自动得到如下各个传感器局部、集中式方法及 3 种信息融合方法关于两个状态变量的滤波曲线，如下图 14、15 所示。

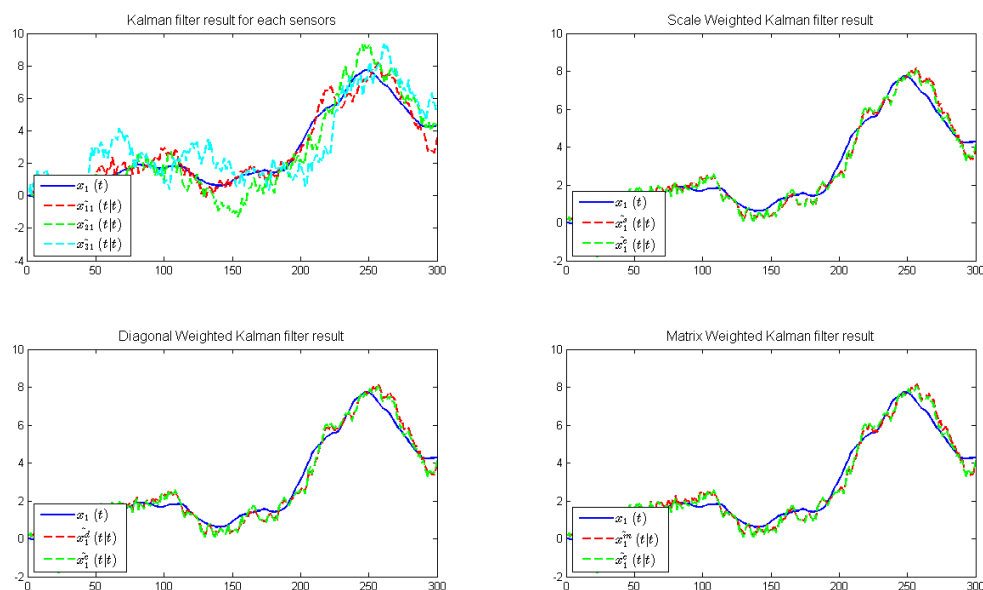


图 14 多传感器信息融合  $x_1(t)$  各种滤波方法对比曲线

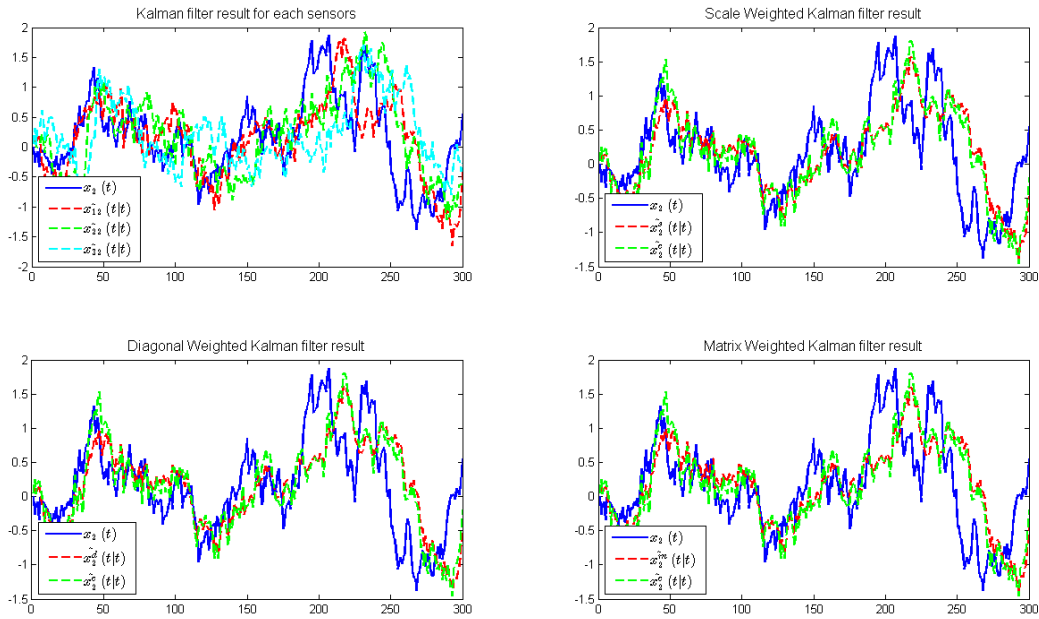


图 15 多传感器信息融合  $x_2(t)$  各种滤波方法对比曲线  
上述曲线与教材<sup>[1]</sup> 中所显示的曲线均基本一致。

#### 4. 结论

本研究在 Matlab 中成功实现了稳态 Kalman 滤波及信息融合算法的实现，并且完成了相应图形用户交互界面的开发，能够方便快捷地针对不同规模维数的稳态 Kalman 滤波器、预报器以及多传感器融合算法进行仿真模拟。总计编写 16 个文件，共 1142 行 Matlab 代码。研究过程中发现了教材<sup>[1]</sup> 中的一处错误，并对不同情况的仿真实例进行了测试，得到了理想的结果。此工具可用于教学、研究、实际使用等多种可能。

#### 致谢

感谢王晓梅教授一个学期的以来在课程教授过程中的辛勤指导与付出，让我学习到了许多新知识与技能。感谢助教对作业的批改与帮助。

感谢刘帅同学这个学期与我一同进修这门课程，与我共同进步。

感谢 Zhiying Zhao 博士提供的优秀的 uitabpanel 控件。

#### 参考文献

- [1] 邓自立，王欣，高媛，建模与估计，科学出版社，2007
- [2] 王晓敏，《最优估计与系统建模》课程讲义，2014
- [3] Zhiying Zhao, uitabpanel.m, 2006, Online Available: <http://www.mathworks.com/matlabcentral/fileexchange/11546-uitabpanel#comments>