

LANGLEY
GRANT
IN-64-CR
287679
36P.

Report for NASA Research Grant NAG-1926

**COMPUTATIONAL TOOLS FOR
MULTI-LINKED FLEXIBLE STRUCTURES**

by

Gordon K. F. Lee (Principal Investigator)
Mars Mission Research Center
North Carolina State University
Raleigh, NC 27695-7910

and

Thomas A. Brubaker, James R. Shults (Investigators)
Dept. of Electrical Engineering
Colorado State University
Ft. Collins, Colorado 80523

(NASA-CR-186139) COMPUTATIONAL TOOLS FOR
MULTI-LINKED FLEXIBLE STRUCTURES (North
Carolina State Univ.) 36 p CSCL 12A

N90-24058

Unclass
63/64 0287679

July 2, 1990

ABSTRACT

This report discusses a software module which designs and tests controllers and filters in Kalman Estimator form, based on a polynomial state-space model. The user-friendly program employs an interactive graphics approach to simplify the design process. A variety of input methods are provided to test the effectiveness of the estimator. Utilities are provided which address important issues in filter design such as graphical analysis, statistical analysis, and calculation time. The program also provides the user with the ability to save filter parameters, inputs, and outputs for future use.

Table of Contents

Abstract	ii
Table of Contents	iii
I. Introduction	1
II. Kalman Filter Theory	2
III. Kalman Filter Parameters	5
IV. Kalman Filter Design Program	9
V. Kalman Filter Design	16
VI. Results	21
VII. Future Work	22
VIII. Conclusions	23
IX. Bibliography	24
Appendix A. Digital Filter Theory	25
Appendix B. Listing of Input File	28
Appendix C. State - Space Transformation Method	31
Appendix D. State - Space Equation File	33

I. Introduction

The Kalman Filter is one of the most powerful state estimation tools available. It is one of the few state estimators which can be used on systems with dynamic parameters. One such system is the flexible link. The parameters of a flexible link vary with load and other environmental restrictions.

Previously, the Kalman Filter has been used extensively to analyze data in an off-line environment. This is due to the amount of time required to calculate the state estimations. With advances in computing speed, the filter is becoming a useful tool for real time applications such as the flexible link.

Another attractive feature of the Kalman Filter is the ease of design. Since it is a time-varying structure, it adapts to any system with very few parameters which must be specified. These parameters are well suited to design in a semi-automated manner by a computer program. Such a program has been developed and will be explained in detail in the following chapters. It is well-known that Linear Optimal Controllers are dual in structure to the Kalman Filter. Hence the user may also employ this package to investigate Linear Quadratic Regulators using a polynomial state-space model.

Chapter 2 presents a brief theory of the Kalman Filter while in Chapter 3, the significant filter parameters which must be defined are explained in detail. The use of the design program is covered in Chapter 4. In Chapter 5, a step by step design of a Kalman Filter is done to clarify the design process. Finally, Chapters 6, 7, and 8 present results, future work, and conclusions, respectively.

II. Kalman Filter Theory

The Kalman Filter is a time-varying recursive digital filter based on a polynomial state-space model. It uses all of the past input data to produce an estimate of the current state vector.

The input signal $z(t)$ is represented by a polynomial of order m . At time $t=nT$, where T is the sampling period of the system, the state vector for the signal is given by

$$z(nT) = [z \ dz/dt \ \dots \ d^m z/dt^m]_{t=nT}^T \quad (2.1)$$

Redefining the state vector as

$$x(nT) = [z \ Tdz/dt \ \dots \ T^m/m!d^m z/dt^m]_{t=nT}^T \quad (2.2)$$

allows the use of a Taylor Series representation for each element of $x(nT)$. The state of the system at $t=(n+h)T$ is then given by the relationship

$$x[(n+h)T] = O[h] \times x[nT] \quad (2.3)$$

where $O[h]$ is the following $(m+1) \times (m+1)$ state transition matrix:

$$O[h] = \begin{bmatrix} 1 & h & \dots & h^m \\ 0 & 1 & \dots & mh^{m-1} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Table 2.1 . Polynomial State Transition Matrix.

The filter is implemented by first producing forecasts of the estimate

$$x_1(nT) = O(1) \times [(n-1)T] \quad (2.4)$$

and covariance matrix

$$S_1(nT) = O(1)S[(n-1)T]O^T(1) + Q \quad (2.5)$$

at $t=nT$ using the previous estimate, $x[(n-1)T]$, and covariance matrix, $S[(n-1)T]$. These forecasts are obtained by using the state transition relationship given in equation 2.3. The matrix Q is the covariance of the driving noise. It allows the designer to "fade" the effects of past inputs. The covariance forecast is then used to calculate the Kalman Gain Matrix,

$$K(nT) = S_1(nT)M^T[o^2 + MS_1(nT)M^T]^{-1} \quad (2.6)$$

The term o^2 is the variance of the measurement noise. The matrix M is a row matrix which relates the measurable state variables to the actual measurements. For this filter, the measurements are assumed to be components of the state-vector $z(nT)$ given in equation 2.1. Therefore, M relates the estimate vector $x(nT)$ given in equation 2.2 to the measured components of $z(nT)$. The Kalman Gain matrix is then used to obtain the covariance estimate,

$$S(nT) = [I - K(nT)M]S_1(nT) \quad (2.7)$$

and the state-vector estimate,

$$x(nT) = x_1(nT) + K(nT)[y(nT) - Mx_1(nT)] \quad (2.8)$$

In equation 2.8, the term $y(nT)$ is the data measurement vector at $t=nT$ which consists of the measured components of $z(nT)$. Equations 2.4 through 2.8 are the basis for the Kalman Filter. A complete derivation for these equations can be found in [1].

The parameters which must be designed for the Kalman Filter are o^2 and Q . The term o^2 is the variance of the measurement noise and Q is the covariance matrix of the driving noise. The variance of the measurement noise can be determined from the sensors used to obtain the

samples. The matrix Q is the key design parameter. For this paper, the driving noise is assumed to be uncorrelated white noise. This simplifies the matrix Q to a diagonal matrix. After testing different forms for the Q matrix, little difference was found. Therefore, the matrix Q was taken to be the identity matrix times a constant, f .

The constant f is known as the fading factor. This term determines how the filter will handle past data. When $f = 0$, the filter is simply an expanding memory filter. This means that all past data is used evenly to calculate the present estimate. This will cause the variance of the estimate and its covariance matrix to decrease to zero as time increases, but deterministic errors will become large. If $f > 0$, more emphasis is placed on the present sample than on past samples. Thus, the past samples are faded from the filter's memory. Larger values of f cause past samples to fade more quickly. The fading of past samples causes the Kalman Filter to have a much smaller deterministic error, but the variance of the estimate and its covariance matrix approach the values at the input.

One way to compromise with the fading factor is to make it small and periodically reinitialize the filter to correct the deterministic error. A nonrecursive digital filter is used to initialize and reinitialize the Kalman Filter. With this digital filter, a window of the past samples is used to estimate the state vector and its covariance. This digital filter is described in detail in Appendix A. The only problem with this approach is that on reinitialization, large jumps are possible in the output value. These jumps are due to the deterministic error in previous output values. Another problem with reinitialization is that the covariance of the estimate is reset to an initial value. This causes an increased amount of noise to be passed through the Kalman Filter for a small number of sampling periods. Both problems can be minimized by varying the fading factor to a point where an acceptable balance is found between the deterministic error and the amount of noise passed through the filter.

III. Kalman Filter Parameters

In the Kalman Filter design program there are seven parameters which must be specified to completely design a filter. Each of these seven parameters will be explained in detail in the following paragraphs.

Order of the Polynomial Model (m)

The order of the polynomial model specifies the number of derivatives which the filter will estimate. The order is chosen by considering two factors. The first is the angular velocity of the input signal oscillations. This determines a maximum sampling period and the order of the polynomial model. The order must be larger for structures with higher angular velocities. When testing the Kalman Filter, a maximum angular velocity of 10 Hz was used. A third order polynomial model was necessary for the filter to track the input well.

Table 3.1. Multiply-Adds for Polynomial Orders 2 and 3.

Operation	Order 2	Order 3
Multiply	123	280
Add	89	221
Divide	1	1
Subtract	3	4
Total	216	978

The second parameter which must be considered is the calculation time necessary for the filter. This time is determined by the order of the filter and the speed of the microprocessor

The number of multiply-adds for orders two and three are shown in table 3.1.

Data Measurement Vector

The Data Measurement Vector is the information which is received from the sensors at each sampling instance. The Data Measurement Vector sets the size of the vector $y(nT)$ in equation 2.8 from chapter II. Examples of this vector include position only or position and velocity. The smallest possible value of the Data Measurement Vector is one corresponding to position only. The largest possible value equals the order plus one. This means that all components of the state vector are being measured. In general a filter with a large data measurement vector will be more effective than one with a smaller vector.

Number of Samples for Initialization

This parameter tells the filter how many samples to use to initialize and reinitialize. The number of samples must be an integer greater than the order of the filter divided by the size of the Data Measurement Vector. This minimum reinitializes the filter estimate vector to the exact value of the most recent data measurement vector and approximates the remaining terms. The maximum number of samples for initialization is an integer less than eight divided by the Data Measurement Vector. This is a restraint created by the filter program, not the filter itself. If the number of samples for initialization is larger than the minimum, the filter approximates the estimate vector as a weighted average of the samples.

Variance of Sensor Noise (σ^2)

The variance of the sensor noise is used to calculate the covariance matrix for the estimate. It is also used in the calculation of the Kalman Gain Matrix in equation 2.6. The variance of the noise is a parameter which is determined by the type of sensor used to obtain

variance of the noise is a parameter which is determined by the type of sensor used to obtain samples.

Fading Factor (f)

The fading factor is a scale factor for the covariance of the driving noise, Q , in equation 2.5. It regulates the deterministic error and noise reduction of the Kalman Filter.

The selection of this parameter should be done in a recursive manner. At first, the filter should be tested with a fading factor of zero. If this test does not meet deterministic error criteria, the fading factor should be increased just to the point where the deterministic error is satisfactory. The minimum fading factor which meets deterministic error criteria is desired for maximum noise reduction. The recursive selection process can be done very effectively with an interactive graphics approach. This way the designer can verify the operation of the filter visually.

Number of Samples to Filter

This parameter is useful only in the filter design program. Due to limited memory, the computer can only process a set number of points. All of these points are stored in memory for future analysis. The program filters data in sets of 100 points. Therefore, the number of samples must be between 100 and 900 in multiples of 100. The limit of 900 points is the maximum number of points which may be stored in the computer's memory.

Sampling Period (T)

The sampling period is the amount of time between samples in seconds. This time can be any positive real number greater than zero. The sampling period should be small enough to satisfy the Nyquist rate:

$$f_s = 1/T \geq 2f_m \quad (3.1)$$

where f_m is the maximum frequency of the bandlimited input signal and f_s is the sampling frequency. The only restraint is that when the filter is implemented, the sampling period must be large enough to complete all calculations necessary for the filter and any control. This restraint need not be adhered to in the design program, but it must be followed when applied to on-line applications.

IV. Kalman Filter Design Program

The Kalman Filter Design Program is a collection of utilities which enable and simplify the design of a Kalman Filter. The program has the capability to design filters, filter data sets, and analyze the resulting output. All of these functions can be performed from the main menu. The following paragraphs describe the capabilities of the design program in detail.

Parameters

This is where the Kalman Filter is designed. All of the parameters which are mentioned in chapter III can be changed here. These parameters should be set before using any other design utilities.

When parameters is selected from the main menu, another menu is brought up which lists all of the parameters that can be changed. A box in the bottom left corner of the screen shows the current value of all Kalman Filter parameters. A parameter is changed by selecting it on the parameter menu, then entering the new value in the prompting box in the middle of the screen. Once a valid value has been entered, the prompt box is cleared and the program returns to the parameter menu. At this point, another parameter can be changed or the user can return to the main menu.

Input

This selection allows the designer to specify the source of the data which is used as input for the Kalman Filter. The input must be setup before filtering data or running the timing tests (two utilities which will be explained later). The input data can be acquired from a text file or produced by state-space system simulation. To start setting up the input, select setup input from the main menu. The designer will then be prompted to choose input from a file or state-space simulation.

To use data from a text file, type **1** to choose **read input from a file**. The program will then prompt for the name of the text file. This name is the DOS filename of the file including the path. Once a valid filename is entered, the program will prompt for the scale factor of all the inputs. If the Data Measurement Vector is 1, the program will ask for the scale factor of position. If it is 2, the program will ask for position and velocity scaling. This allows the designer a simple way to scale input data. If the data is properly scaled in the file, the scale factors should be unity. After all scale factors have been entered, the program will read the inputs from the file, scale them, and store them in memory.

The text file which holds the data should have the data in columns. The first column should be position, the second velocity, and so on. The file should have the number of columns specified by the Data Measurement Vector. For example, if the Data Measurement Vector is 1, the text file should have only one column representing position. Each row should represent one sample. Therefore, there should be at least as many rows in the text file as the number of samples to be filtered. A sample input data file is shown in Appendix B. This file consists of 100 samples of position and velocity measurements. This file, with 900 samples, is included on the program disk under the name `real2.dat`.

The state-space equations of a system can also be used to produce input for the Kalman Filter. To use **state-space system simulation**, type **2** to choose **simulate a system with feedback**. A simulation setup menu will give the user three choices relevant to setting up a system simulation. The first choice is **state equations**. This selection will facilitate the entry of state-space equations. The state-space equations should have the following format

$$\frac{dx}{dt} = Ax + Bu \quad (\text{Analog}) \quad (4.1)$$

$$y(t) = Cx(t) \quad (4.2)$$

$$x[n+1] = Ax[n] + Bu[n] \quad (\text{Digital}) \quad (4.3)$$

$$y[n] = Cx[n] \quad (4.4)$$

After selecting state equations, the first question the user is prompted for is whether the state-space equations are in analog or digital form. If they are analog, they will be transformed to digital with all states preserved. The transformation method is described in Appendix C. Next, the program asks if the state equations should be read from a file or entered from the keyboard. If file entry is selected, the program prompts for the name of the file containing the equations. This file should be a text file of the form shown in Appendix D. If keyboard entry is selected, the program will first prompt for the order of the system. Then it will prompt for the A, B, and C matrices. The A matrix should have the same number of rows and columns as the order of the system. The B matrix should have one column and system order rows. The C matrix should have system order columns and the same number of rows as the Data Measurement Vector. Row 1 should produce position, Row 2 velocity, and so on. After equation entry is complete, the program will return to the system simulation menu.

The second item on the system simulation menu is **reference input**. This input is used along with feedback to produce $u[n]$, the input to the state-space system. This is done in the following manner

$$u[n] = \text{Gain} * (r[n] - F * Y[n]) \quad (4.5)$$

In this equation, the vector F, Y[n], and Gain are part of the feedback and will be discussed later. The term $r[n]$ is the reference input. This is the desired response of the system. For example, if the step response is desired, $r[n] = 1$. Four choices are available for the reference input. The first is **step**. The program will ask for the magnitude of the step and then it will set $r[n]$ to

$$r[n] = \text{magnitude} \quad (4.6)$$

The second option is **ramp**. The program will ask for the slope of the ramp. Then the equation for $r[n]$ is

$$r[n] = \text{slope} * n * T \quad (4.7)$$

where T is the sampling period of the system. The third available reference input is a **sine** wave. The program will ask for the magnitude and angular velocity in radians per second of the signal. The reference input then becomes

$$r[n] = \text{magnitude} * \sin(\text{angular velocity} * n * T) \quad (4.8)$$

where T is still the sampling period. The final way to set $r[n]$ is to **read the points from a text file**. This file should have one column of data with each row representing one sampling period. There should be at least as many points in this file as the number of samples to be filtered.

The final system simulation option is **feedback**. This option allows the designer to do state feedback with the outputs of the Kalman Filter. Here the vector $Y[n]$ is the output of the filter and has the following format

$$Y[n] = [\text{position} \ \text{velocity} \ \text{acceleration} \ \dots]^T \quad (4.9)$$

The vector F is the feedback vector. This vector has the form

$$F = [F[0] \ F[1] \ F[2] \ \dots] \quad (4.10)$$

where $F[0]$ represents the gain of the position state, $F[1]$ is the gain of velocity, and so on. The default value of these gains is 0 implying no feedback. The term **Gain** is the proportional system gain for the system. This term has a default value of 1 or unity gain.

Filter Data

After the designer has set the Kalman Filter Parameters and setup the input for the filter, the designed Kalman Filter can be run. The valid input for the filter is either input from a file or system simulation. The Filter Data option runs the filter automatically when it is selected. The only action the designer must make is to wait for the program to return to the main menu after completion of filtering. At any time during the filtering of data, the user can stop the filter and return to the main menu by pressing ESC.

Timing Tests

This utility allows the user to run the Kalman Filter and test for the filtering rate. In this option, optimized filter modules are run to find the average calculation time for each sample. This is done using a time stamp before the filter is started and one after it is completed. The difference between these two times is the total filtering time. The average calculation time per sample is simply the total filtering time divided by the number of samples filtered. This method of timing does not provide a maximum calculation time because the resolution of the timing is only 1/100th of a second. A timing method with a much smaller resolution must be found if the maximum calculation time is to be found. When running the timing tests, only input data from a file can be used as input to the Kalman Filter. System simulation is not allowed because the time to do the simulation would provide inaccurate calculation times.

Graph Output

A graphing utility has been designed which allows the designer to analyze the output of the Kalman Filter. This utility plots filter input and output on the same graph to allow visual verification of filter performance. The Graph Output utility begins by giving the user a menu

of available plots. This menu gives options such as position, velocity, etc. After the user selects a plot, the program will ask for the units of the y-axis. This is the units of the signal which is to be filtered such as radians, meters, or radians/sec. The graph will then be plotted on the screen. The user can return to the available plots menu by pressing any key after the graph is completed.

Analyze Data

This option will analyze filter inputs and outputs for statistical data. This information can then be used to design filters more efficiently. The user is presented with a menu with choices of available data to be analyzed such as position data, velocity data, etc. The following statistical information is calculated for the selected signal:

Variance of input and output

Average of input and output

Summed Absolute Error between input and output

Average Error per sample

The statistical information is displayed on the screen after calculation. The user can return to the analyze data menu by pressing any key.

Save Data

The design program allows the user to store filter inputs, outputs, and parameters in a text file for future reference. The program prompts for the name of a file to save the data in. If the file exists, it will be erased and replaced with the present data. If the file does not exist, it will be created.

Read Data

This utility is used to load data into the program from files created by the **Save Data** utility. It prompts the user for the name of a text file containing data. When a valid filename is entered, it will read the Kalman Filter inputs, outputs, and parameters from the file. The user should take caution when using this utility because all of the data and parameters which the program had before reading data from the file will be lost. The input which is read from the file is treated as input from a file and can be refiltered at the users discretion.

V. Kalman Filter Design

Now that the Kalman Filter and the program utilities have been explained, a filter can be designed. The Kalman Filter Package consists of two disks. The first disk is the program disk. The files which are included on this disk are shown below

KALMAN.EXE - Kalman Filter Design Program

KALMAN.HLP - Program Help File

MSHERC.COM - Hercules Graphics Driver

READ.ME - Instructions for Kalman Filter Program

The file **KALMAN.HLP** contains help information which is used by **KALMAN.EXE** to provide **on-line help** to the designer. This help file should not be altered or erased. Such action could disable the on-line help. This help can be called from any menu by pressing the F1 key. A help screen on the current selection will be displayed. The file **MSHERC.COM** is used only when the program is run on a system with a **Hercules compatible monochrome monitor**. When such a monitor is to be used, the designer should type **MSHERC** at the **DOS** prompt before starting the Kalman Filter Design Program. The file **READ.ME** is an ASCII file which contains general instruction on running the Kalman Filter Program. This file can be printed with the DOS print command or viewed with the DOS type command.

The second disk is the data disk

REAL1.DAT - Input file for position only

REAL2.DAT - Input file for position and velocity

REAL3.DAT - Input file for position, velocity, and acceleration

REAL4.DAT - Input file for position, velocity, acceleration, and
d/dt(acceleration)

ASTATE.3 _ 3rd order analog state-space equations

ASTATE.4 - 4th order analog state-space equations

ASTATE.8 - 8th order analog state-space equations

DSTATE.3 - 3rd order digital state-space equations

The input files **REAL1.DAT** and **REAL2.DAT** contain data which was acquired from the flexible link test bed at Colorado State University. **The input files REAL3.DAT and REAL4.DAT contain files for testing Data Measurement Vectors of sizes 3 and 4. These files were created by using position and velocity data from the flexible link and estimating the acceleration and d/dt(acceleration) data with the Kalman Filter Program.** **ASTATE.8** is the 8th order state-space model for the hub of the flexible link. **ASTATE.3** and **DSTATE.3** are 3rd order analog and digital state-space models of a system with a unit step response of

$$y(t) = 1 - e^{-3.1416t} * \cos(62.832 * t) \quad (5.1)$$

The file **ASTATE.4** is the same state-space model as in **ASTATE.3** with the addition of a rigid mode.

To begin the design process type **KALMAN** at the DOS prompt to start the design program. A main menu will be displayed which allows access to the utilities described in chapter IV. These utilities can then be used to design, run, and analyze a Kalman Filter. The following chapter will present 2 examples of Kalman Filter design. The first example will be a filter with the following parameters:

Order = 3 Data Measurement Vector = 2

Number of Samples for Initialization = 2

Number of Samples = 100

Sampling Period = 0.01

Variance of Sensor Noise = 1×10^{-4}

Fading Factor = 1×10^{-9}

Input from file REAL2.DAT

The second example will be a filter with the following parameters:

Order = 2 Data Measurement Vector = 1

Number of Samples for Initialization = 2

Number of Samples = 100

Sampling Period = 0.01

Variance of Sensor Noise = 1×10^{-4}

Fading Factor = 5×10^{-12}

Input is system simulation from file ASTATE.8

The design of both of these filters is discussed in the paragraphs that follow with example one presented first.

Example 1

From the main menu, the first selection should be Parameters. The parameters menu will be displayed on the screen. At this time all filter parameters should be set as described above except for the input method. After setting the parameters, select **Return to Main Menu**. From the main menu, select **Input**. When prompted for input method, type 1 to select **Read Inputs from a File**. Next, the name of the file should be entered as **REAL1.DAT**. The program will ask for the scale factor for position and velocity. Type 1 for both because the input is already scaled properly. The program will then read the inputs from the file and return to the main menu. At this point, the filter design is completed. The designer can then run the filter using the **Filter Data** or **Timing Test** utilities. After the data has been filtered, the program will return to the main menu. The data can then be analyzed or saved using the **Graph Output**, **Analyze Data**, or **Save Data** utilities.

Example 2

As in example 1, the first selection should be **Parameters**. The parameters should be set as specified above. After the parameters have been set, select **Return to Main Menu**. From the main menu, select **Input**. When prompted for input method, type **2** to select **state-space simulation**. The program will then display the system simulation menu. Select **State Equations** to enter the state-space equations. The first prompt is if the equations are analog or digital. Type **1** to select analog equations. Then the program will ask if the equations should be read from a file or entered from a keyboard. Type **1** to select file entry. When the program prompts for the file name, type **ASTATE.8**. The program will read the equations, convert them to digital form, and return to the system simulation menu.

Next select **Reference Input**. Another menu will be displayed with the choices of input. Select **Step** to get the step response of the system. For the magnitude, enter **1** for the unit step response.

The final selection from the system simulation menu should be **Feedback**. The program will ask for **F[0]**, **F[1]**, **F[2]**, and **Gain**. They should be set to the following values

$$F[0] = 1$$

$$F[1] = 0$$

$$F[2] = 0$$

$$\text{Gain} = 2.5$$

After setting the feedback, the program will return to the system simulation menu. At this time the designer should select **Return to Main Menu**.

The Kalman Filter can now be run using the **Filter Data** utility (Timing Test is not allowed for system simulation). After completion of filtering, the data can be analyzed or saved using the **Analyze Data**, **Graph Output**, or **Save Data** utilities.

The examples presented here are presented only to aid in learning to use the Design Program. The designer should try altering the filter parameters to gain more insight into their effect on the filter. The sample data files can be used for this purpose or new files can be created to characterize any system desired by the designer.

VI. Results

The Kalman Filter has been studied as a state estimator for on-line applications. This filter was found to give very good state estimations when based upon a polynomial model. The polynomial model also provided for very simple design methods. Since the design of the Kalman Filter is very straight forward, a program has been designed to aid in the design process. This program allows a designer to design a filter and test the output to confirm the effectiveness of the filter. An interactive graphics approach has been used along with statistical data to provide insight into the effectiveness of the designed filter. After an acceptable design has been completed, a timing test utility can be used to check the speed of the filter. These utilities provide a very uncomplicated way to design an effective state estimator based on the Kalman Filter.

VII. Future Work

Future work on the Kalman Filter consists of implementation issues. A good design method has been set forward, but many implementation issues have not been addressed. The major implementation problem comes with the selection of hardware to run the Kalman Filter. This hardware must consist of a microprocessor which is fast enough to meet the sampling rate, but inexpensive enough to make its use feasible. Another major problem comes in the sensors which will provide state information for the Kalman Filter. These sensors must be selected for each application such that they provide accurate information with a low noise level. A final problem which must be addressed is a method to download the Kalman Filter program to the hardware. This downloading could be done by the design program with the appropriate interfaces. After these problems have been resolved, the Kalman Filter will provide a very effective state estimator in many applications.

The hardware implementation and downloading components will be completed upon final discussions with NASA personnel on the computer systems available in the SPACECRAFT Dynamics Branch.

VIII . Conclusions

An investigation has been started into the usefulness of the Kalman Filter as a state estimator in an on-line environment. Previously, the filter has been used strictly in an off-line capacity to do data analysis. With the advances in computing speed, the filter is now becoming feasible as a real-time state estimator.

A Kalman Filter based on a polynomial state-space model has been tested on flexible structure data. This filter has proven to give excellent state-estimation and noise reduction in such systems. The polynomial model provides for a very easy design of the filter. Due to the ease of design, a program has been written to provide assistance in the design process.

The Kalman Filter Design Program provides a very straight forward design methodology with an interactive graphics approach. This approach allows the designer to see how well the filter works and the effects of changes in the design parameters. When the program is combined with the appropriate hardware, a very effective state estimation tool will become available for use in the real time environment.

IX. Bibliography

- T Brubaker, F. Cornett, and C. Pomernacki, Linear Digital Filtering for Laboratory Automation, Proceedings of the IEEE, Vol. 63, No. 10, October 1975.
- C. Phillips and H. Nagle Jr., Digital Control System Analysis and Design, Englewood Cliffs, N.J.: Prentice Hall, Inc., 1984.
- B. Kuo, Automatic Control Systems, Englewood Cliffs, N.J.: Prentice Hall, Inc., 1987.

Appendix A. Digital Filter Theory

The filter which is used to initialize and reinitialize the Kalman Filter is a nonrecursive digital filter. This filter uses a window of the past outputs to estimate the present state vector.

The nonrecursive digital filter is based on the polynomial model given in chapter 2 on Kalman Filter theory. The derivation begins by assuming that a signal $z(t)$ is approximated by a polynomial over a finite data window. Defining the state of the signal by equation (2.2) in chapter 2, the state of the signal at time $t=(n+h)T$ is then given in terms of the signal at time $t=nT$ by equation (2.3). Given that the signal starts at $t = 0$, the first l observations are each defined as

$$m[jT] = Mx[jT] + v[jT], \quad j = 0, 1, 2, \dots, l-1 \quad (\text{A-1})$$

where M is a row matrix that relates the measurable state variables to the actual measurements. The elements of each noise vector $v[jT]$ are the measurement noises. In general, these are taken as random variables with zero mean, and the time-dependent autocovariance matrix

$$R[jT] = E[v(jT)v^l(jT)] \quad (\text{A-2})$$

The noise covariance matrix is usually not known, and for scalar measurements, the noise variance is assumed constant and the noise uncorrelated. For the remainder of this appendix, only the scalar measurements and uncorrelated measurement noise with time-invariant statistics will be assumed. The mean value of the noise will be taken as zero and the variance as σ^2 . The results are easily extended to vector measurements and to measurement noise with time-varying statistics.

For l observations, the total observation vector at $t = nT$ is defined as

$$m_l[nT] = [m[nT] \ m[(n-1)T] \ \dots \ m[(n-l+1)T]]^T \quad (\text{A-3})$$

This vector now forms a data window of l data points. For an estimate of the data at $t=nT$, the use of the expression

$$x[(n-j)T] = O(-j)x[nT] \quad (\text{A-4})$$

is combined with (A-3) to yield

$$m_l[nT] = [M \ M O(-1) \ \dots \ M O(-l+1)]^t x[nT] + v_l[nT] \quad (\text{A-5})$$

The matrix of constants $H[nT]$ is now defined as

$$H = [M \ M O(-1) \ \dots \ M O(-l+1)]^t \quad (\text{A-6})$$

so that (A-5) can be written as

$$m_l[nT] = Hx[nT] + v_l[nT] \quad (\text{A-7})$$

The optimal estimate of the data at $t=nT$ is now given in terms of minimum variance. If the covariance matrix for the total observation vector is

$$R_l[nT] = E[v_l(nT)v_l^t(nT)] \quad (\text{A-8})$$

the optimal minimum variance estimate is

$$y[nT] = Wm_l[nT] \quad (\text{A-9})$$

where W is a series of constant weights given by

$$W = [H^t H]^{-1} H^t \quad (\text{A-10})$$

The covariance matrix for the estimate given by (A-8) is given by

$$S = [H^t H]^{-1} \sigma^2 \quad (\text{A-11})$$

for the uncorrelated noise. The equations (A-9) (A-10) and (A11) are used in the Kalman Filter program for initialization and reinitialization of the Kalman Filter.

Appendix B. Listing of Input File

This appendix contains a listing of the input text file **REAL2.DAT**. The data in this file was acquired from the flexible link at Colorado State University. This file consists of 100 samples of position and velocity measurements. It is to be used as input for a Kalman Filter with a Data Measurement Vector of 2. The first column is position and the second column is velocity. Each row represents one sample with the first row being sample number 1 and the last row being sample number 100. Comment in parenthesis have been included in the following listing, but they should not be included in the actual file.

(position)	(velocity)	
-0.0066472501	-0.0156782214	(sample 1)
-0.0047297743	0.0049885251	(sample 2)
-0.0008948221	0.0562990718	
0.0040906155	0.1489431113	
0.0102265384	0.2743688822	
0.0167459566	0.3933808506	
0.0240323655	0.5138180852	
0.0317022689	0.6164391637	
0.0409061536	0.7190603018	
0.0489595532	0.7874743342	
0.0566294566	0.8216813803	
0.0639158636	0.8558884263	
0.0708187819	0.8672907352	
0.0773381963	0.8444860578	
0.0830906257	0.8095663786	
0.0884595588	0.7668076158	
0.0919110179	0.7204855680	
0.0953624696	0.6499335766	
0.0984304324	0.5758183599	
0.1011148989	0.5024157763	
0.1041828617	0.4368522763	
0.1068673283	0.3898176253	
0.1095517948	0.3506220579	
0.1126197577	0.3328059018	
0.1156877205	0.3299553096	
0.1191391721	0.3427829444	
0.1222071350	0.3541852832	
0.1256585866	0.3449208736	
0.1279595643	0.3484841287	
0.1302605271	0.3342311978	

0.1321780086	0.3157023787
0.1333284974	0.2850585878
0.1337119937	0.2408744991
0.1340954900	0.1938398331
0.1333284974	0.1297016591
0.1314110160	0.0612875968
0.1287265569	-0.0199541003
0.1252750903	-0.1133107841
0.1210566461	-0.1931271851
0.1160712093	-0.2957482934
0.1107022762	-0.3855417371
0.1041828617	-0.4774731398
0.0980469361	-0.5587148070
0.0926780030	-0.6064621210
0.0880760625	-0.6235656738
0.0846246034	-0.6043242216
0.0819401368	-0.5608527660
0.0811731517	-0.4924387038
0.0815566480	-0.3926681876
0.0834741220	-0.2822079957
0.0865420848	-0.1418166459
0.0907605290	0.0007126465
0.0961294621	0.1489431113
0.1022653878	0.2829206586
0.1095517948	0.4161855280
0.1172216982	0.5380480886
0.1252750903	0.6456577182
0.1333284974	0.7432902455
0.1421488822	0.8231066465
0.1509692818	0.8858195543
0.1590226740	0.9200266004
0.1678430587	0.9357048273
0.1747459769	0.9471071362
0.1885517985	0.8772677779
0.1946877241	0.8444860578
0.2012071460	0.8031525612
0.2077265680	0.7760720253
0.2146294713	0.7675202489
0.2215323895	0.7760720253
0.2280518115	0.7639570236
0.2353382111	0.7760720253
0.2422411293	0.7896122932
0.2495275438	0.7988767028
0.2568139434	0.8216813803
0.2644838393	0.8337963820
0.2725372612	0.8444860578
0.2802071571	0.8444860578
0.2867265642	0.8359342813
0.2936294973	0.8238193393
0.2989984155	0.8017272949

0.3039838672	0.7646696568
0.3074353039	0.7005314827
0.3097362816	0.6278415322
0.3108867705	0.5252204537
0.3108867705	0.4111970067
0.3101197779	0.2971735895
0.3082022965	0.1674719155
0.3062848151	0.0498852506
0.3039838672	-0.0448967256
0.3016828895	-0.1261384189
0.2993819118	-0.1817248464
0.2970809340	-0.2202077508
0.2959304452	-0.2394492179
0.2947799861	-0.2273342162
0.2943964899	-0.2166445255
0.2951634824	-0.1710351557
0.2970809340	-0.1054716781
0.3001489043	-0.0185288079
0.3036003709	0.0805290490

(sample 100)

Appendix C. State - Space Transformation Method

A technique is developed in this section for determining the discrete state equations of a digital system directly from the continuous state equations. In fact, the states of the continuous model become the states of the discrete model. Thus, the natural states of the system are preserved.

To develop this technique, consider the continuous state equations

$$dv(t)/dt = A_c v(t) + b_c u(t) \quad (C-1)$$

$$y(t) = C_c v(t) + d_c u(t)$$

The solution to these equations is

$$v(t) = O_c(t - t_0)v(t_0) + \int_{t_0}^t O_c(t - \tau)b_c u(\tau)d\tau \quad (C-2)$$

where the initial time is t_0 and where

$$O_c(t - t_0) = \sum_{k=0}^{\infty} \frac{A_c^k (t - t_0)^k}{k!} \quad (C-3)$$

To obtain the discrete model we evaluate (C-2) at $t = kT + T$ with $t_0 = kT$, that is

$$v(kT + T) = O_c(T)v(kT) + m(kT) \int_{kT}^{kT+T} O_c(kT + T - \tau)b_c d\tau \quad (C-4)$$

Compare this result with the discrete state equations

$$x(k+1) = Ax(k) + bm(k) \quad (C-5)$$

$$y(k) = Cx(k) + dm(k)$$

If we let

$$x(kT) = v(kT)$$

$$A = O_c(T) \quad (C-6)$$

$$b = \int_{kT}^{kT+T} O_c(kT + T - \tau) b_c d\tau$$

we obtain the discrete state equations for the digital system. When the output equation of (C-1) evaluated at $t=kT$, we get the output equation of (C-5). Thus the discrete c and d values are equal to the continuous C_c and d_c values respectively.

The discrete system matrices A and b may be evaluated by using the series (C-3). Then expanded this series becomes

$$O_c(T) = I + A_c T + A_c^2 T^2 / 2! + A_c^3 T^3 / 3! + \quad (C-7)$$

Since this is a convergent series, the series can be truncated with adequate resulting accuracy.

The integral necessary for the computation of b can also be easily evaluated using a series expansion. The resulting equation for b is

$$b = IT + A_c T^2 / 2! + A_c^2 T^3 / 3! + \quad (C-8)$$

A computer program has been written into the Kalman Design Program which implements equation (C-7) and (C-8) to derive A and b .

Appendix D. State - Space Equation File

The files which are used to store the state-space equations for system simulations are text files with very straight forward format. The files for continuous and discrete equations have the same format. The example file shown below is a continuous state-space system of order 3 for Data Measurement Vectors of 1, 2, or 3. The comments in the parenthesis are provide to simplify the example, but they should not be included in the file.

Example

3	0	1	
			0(Order of System = 3)
0	0	1	(3x3 A matrix)
0	-3947.8	-3.1416	
0			
0			(3x1 b matrix)
3947.8			
1	0	0	(position)
0	1	0	(velocity) (DMVx3 C matrix)
0	0	1	(acceleration)