Introduction to MATLAB

MATLAB Toolboxes

Computer Applications in CEE

Drs. Trani and Rakha

Civil and Environmental Engineering

Virginia Polytechnic Institute and State University

Spring 2000

Sample Toolboxes

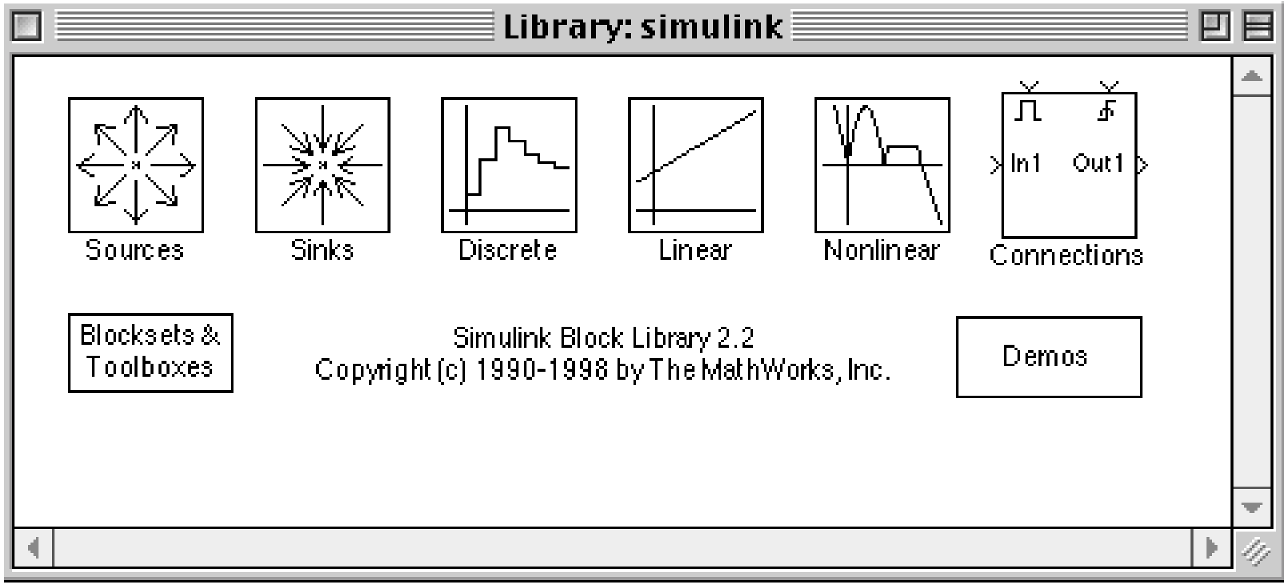
- MATLAB has been extended over the years to respond to the needs of various users
- Several toolboxes exist to add to the power of the original language. For example:
  - Simulink
  - Fuzzy Logic
  - Neural Networks
  - Optimization
  - Controls
  - C/C++ compiler library
  - Real-time workshop

Simulink

- Simulink is a powerful toolbox to solve systems of differential equations
- Simulink has applications in Systems Theory, Control, Economics, Transportation, etc.
- The Simulink approach is to represent systems of ODE using block diagram nomenclature
- Simulink provides seamless integration with MATLAB. In fact, Simulink can call any MATLAB function
- Simulink interfaces with other MATLAB toolboxes such as Neural Network, Fuzy Logic, and Optimization routines

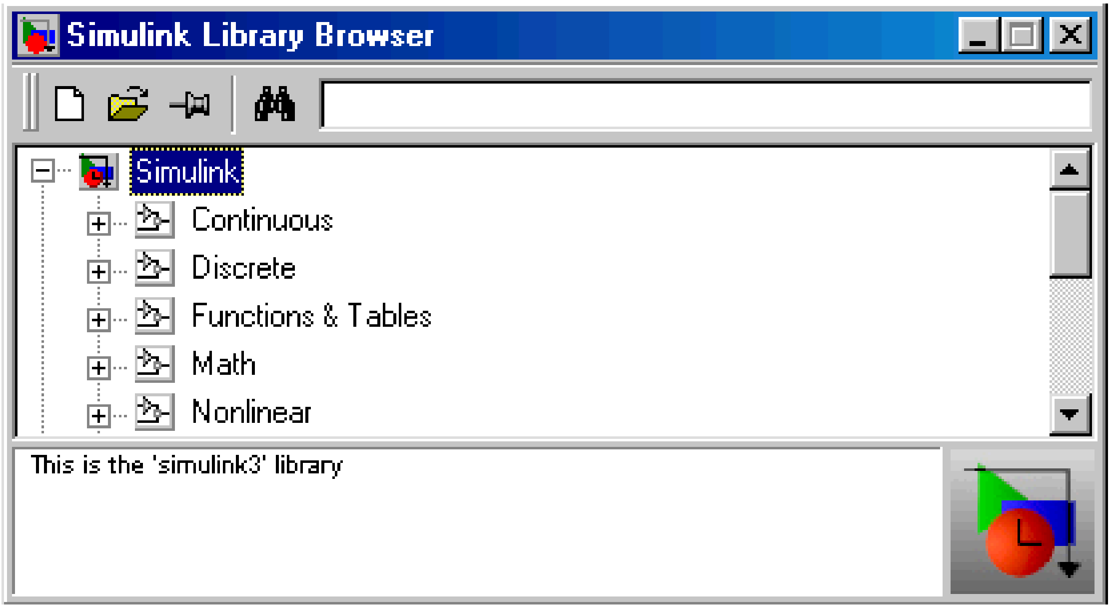
Simulink Building Blocks

- Simulink has a series of libraries to construct models
- Libraries have object blocks that encapsulate code and behaviors
- Connectors between blocks establish causality and flow of information in the model



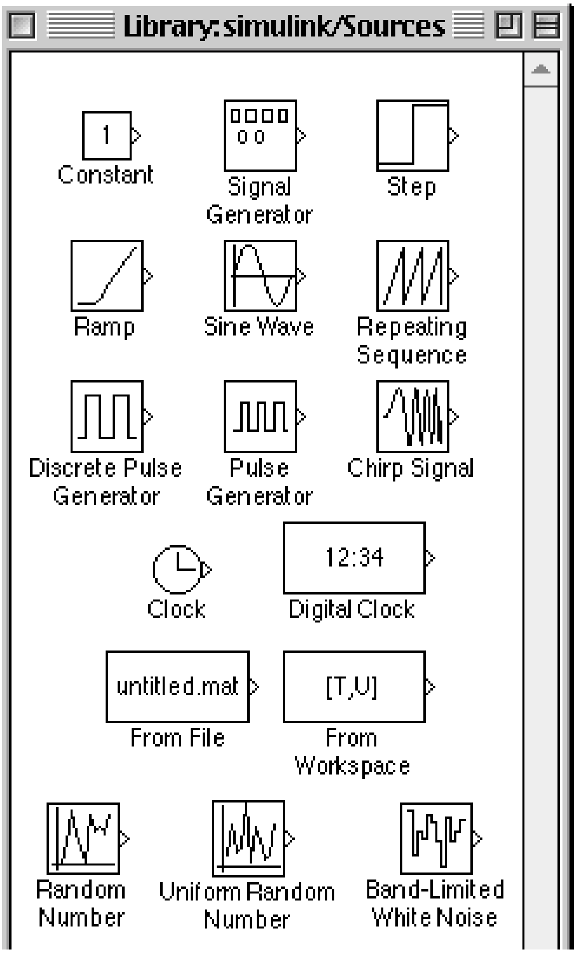
Simulink Interface

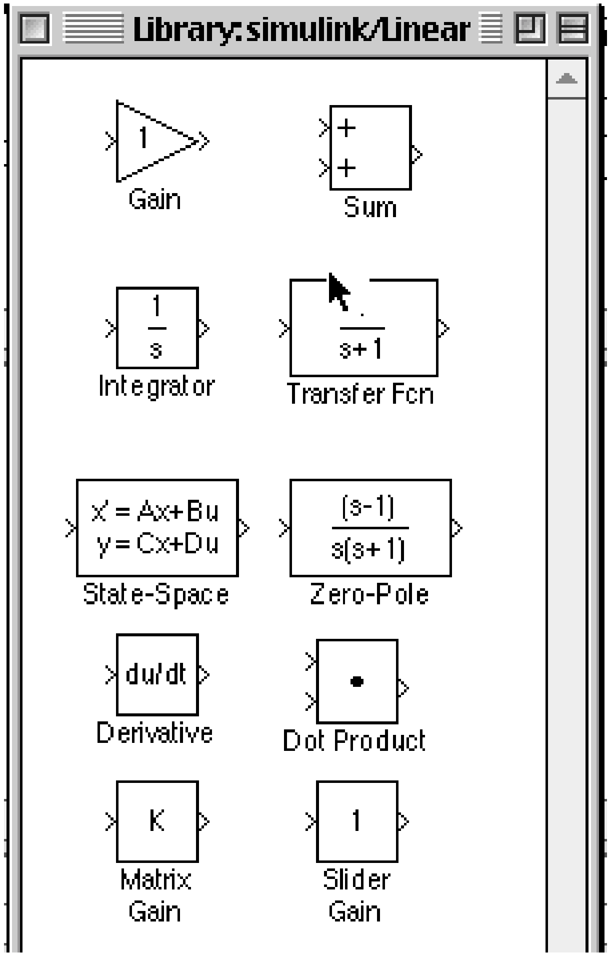
- The main application of Simulink is to model continuous systems
- Perhaps systems that can be described using ordinary differential equations



Typical Simulink Libraries

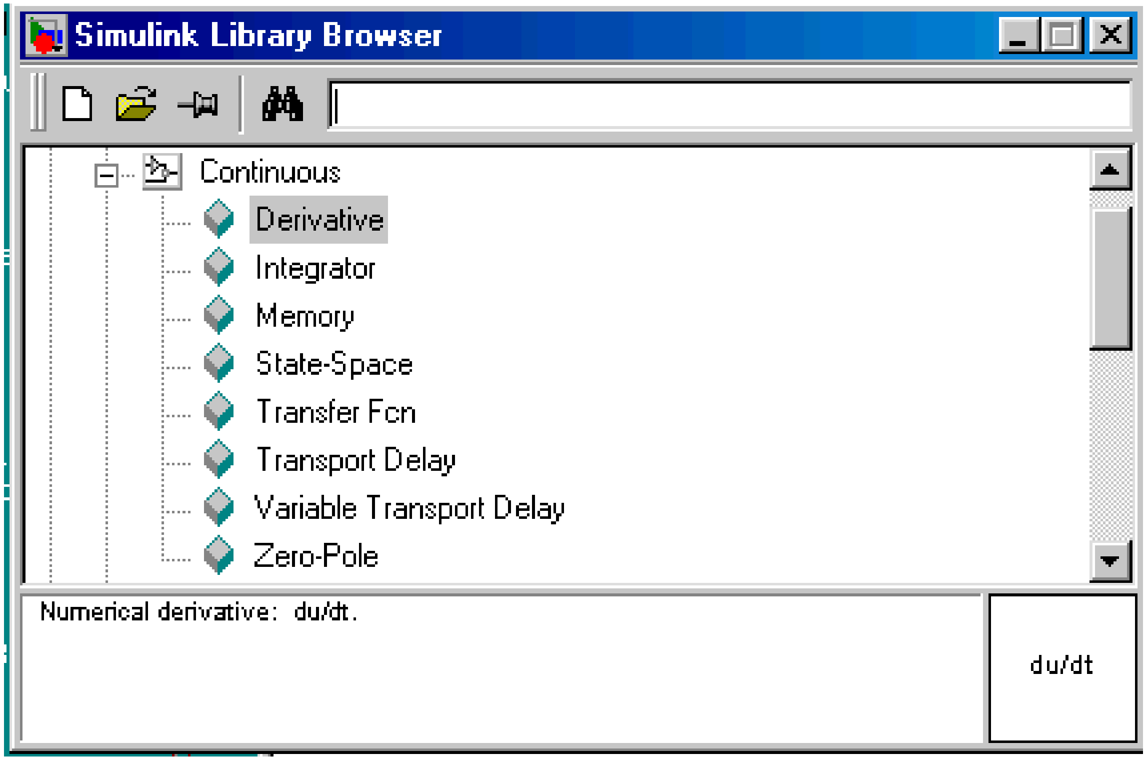
Shown are some typical Simulink libraries (Macintosh)





Sample Simulink Library (Windows and UNIX)

The new Simulink interface in Windows/UNIX uses standard OOP interfaces (Visual C++, Visual Basic)



Simulink Example

The following example illustrates the use of Simulink to solve a two vehicle car-following problem. This problem has been studied for the past 40 years in traffic flow theory

$$\ddot{x}(t+\tau)_{fc} = k(\dot{x}(t)_{lc} - \dot{x}(t)_{fc})$$

where: k is a gain constant of the response process

$\ddot{x}(t+\tau)_{fc}$ is the acceleration of the following vehicle

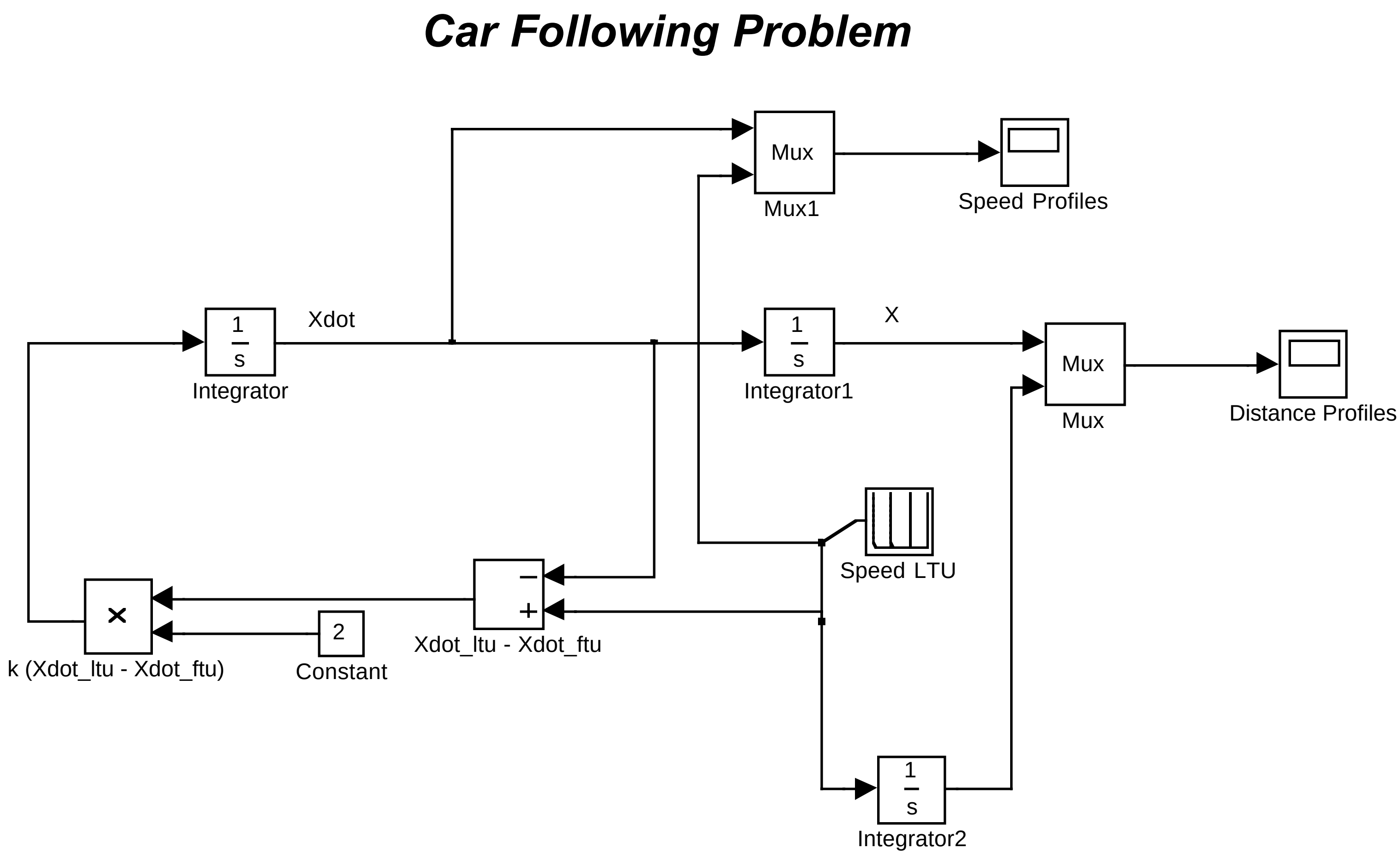
$\dot{x}(t)_{lc}$ is the speed of the leading vehicle

$\dot{x}(t)_{fc}$ is the speed of the following vehicle

Set-Up of the Problem

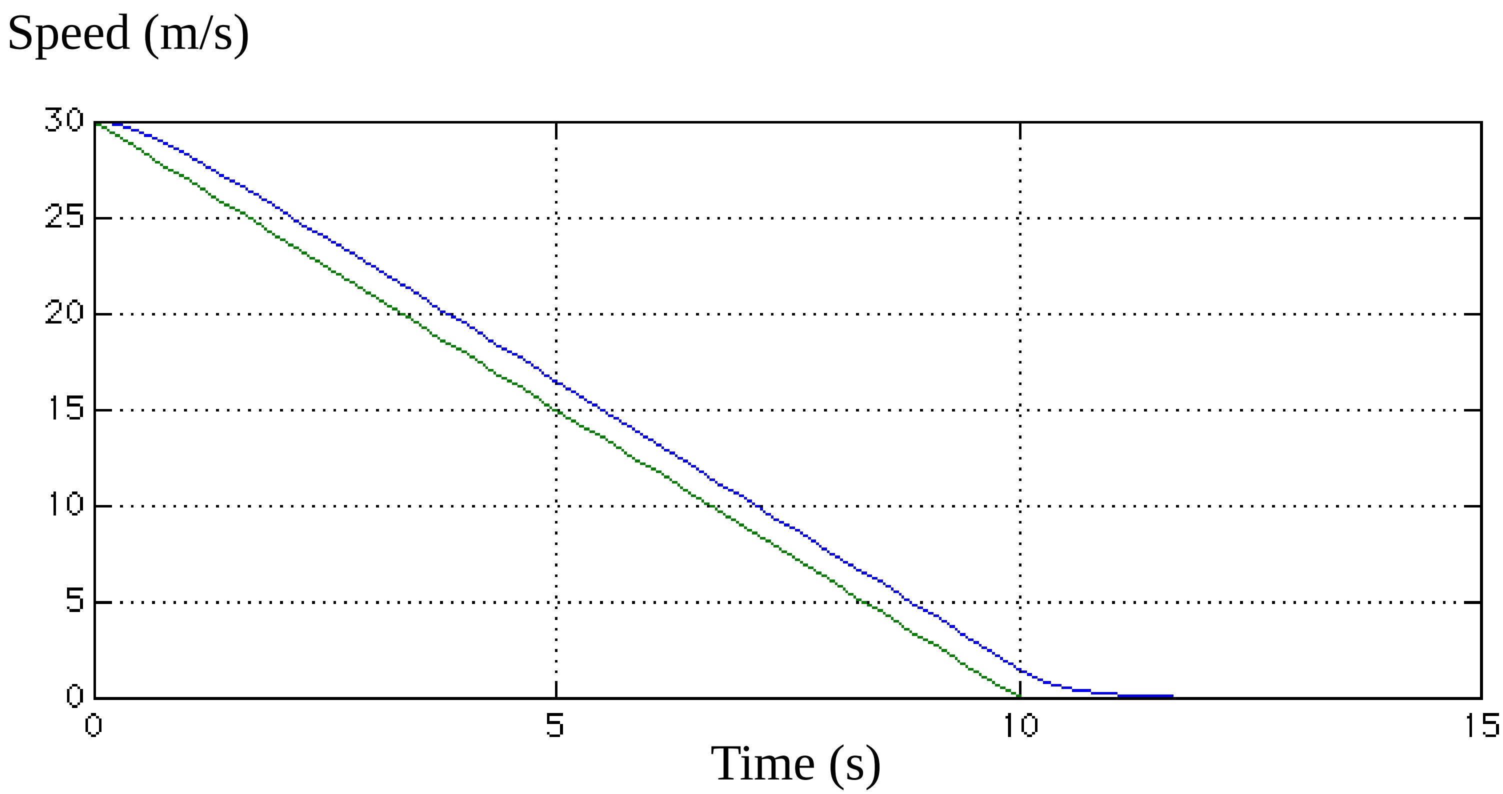
- Assume the velocity profile of the leading vehicle is known (we "drive" this car to test the response of the following vehicle)
- Initially, assume no time lags in the acceleration response of the following vehicle ($\tau = 0$)
- Test an emergency braking maneuver executed by the first vehicle at 3 m/s^2
- Test a new scenario with a deterministic time lag response time of 0.75 seconds
- Verify that both cars do not collide

Simulink Representation of the Car-Following Problem



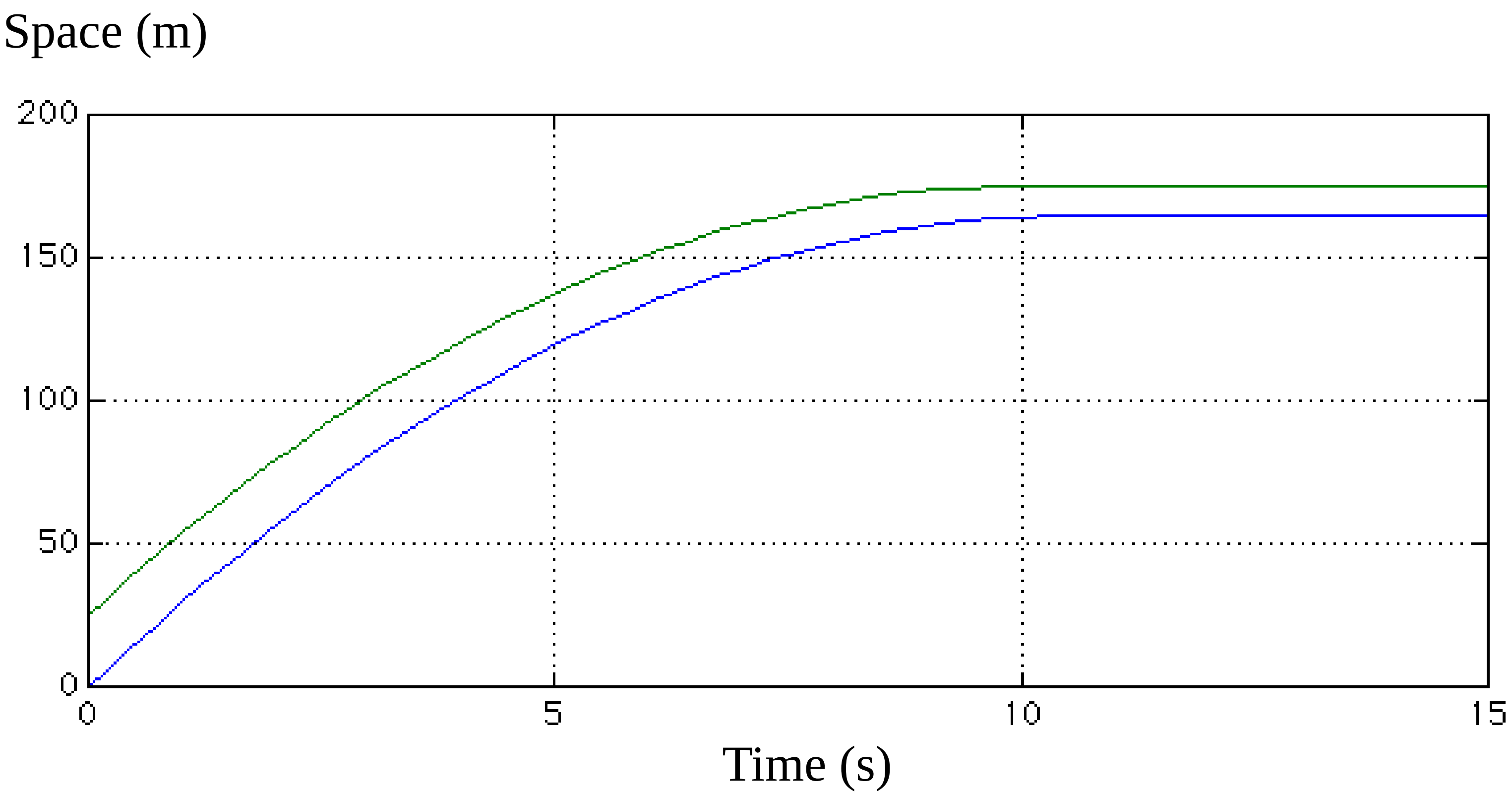
Car-Following Model Output

Velocity profiles for leading and trailing vehicles



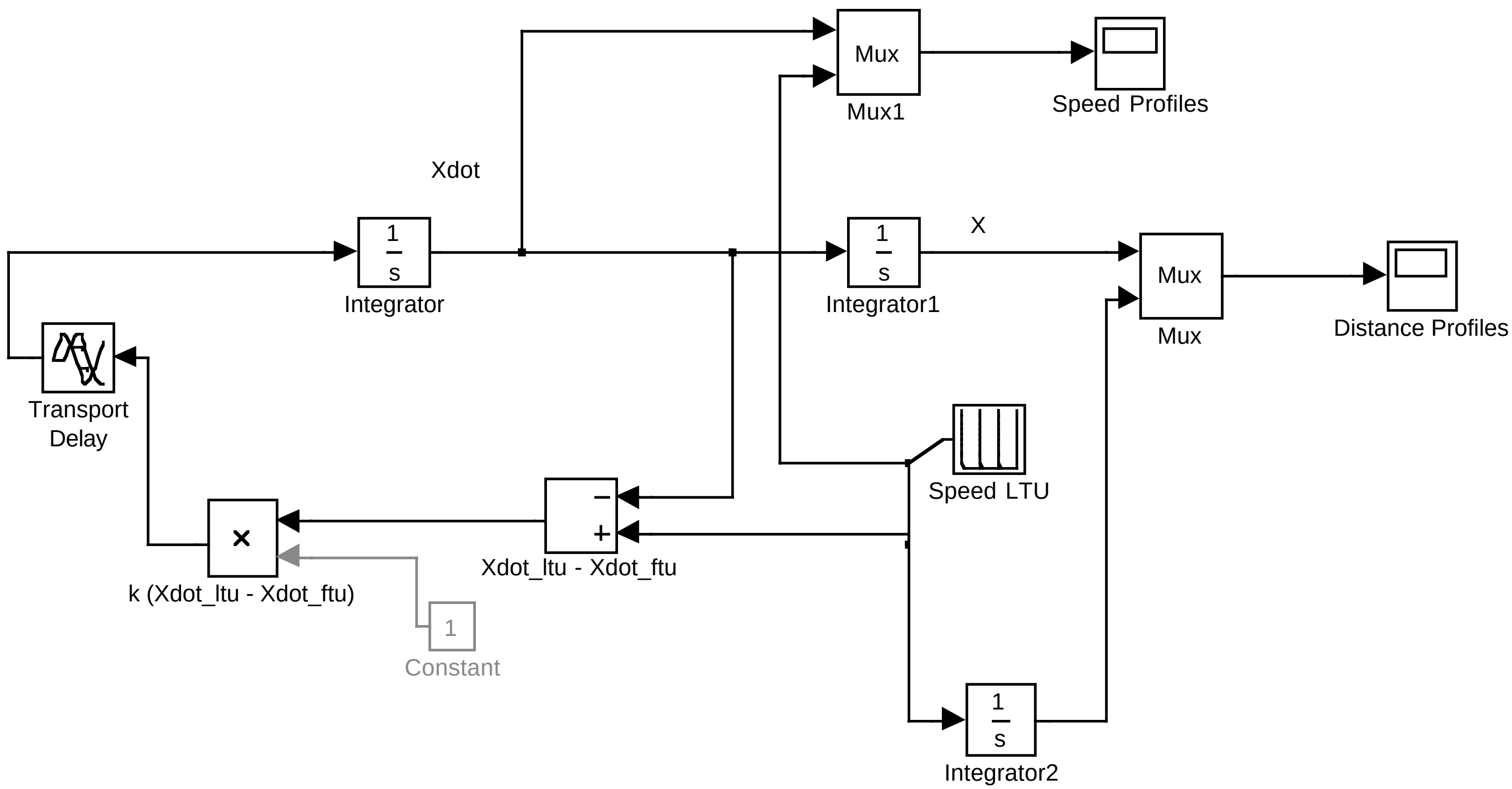
Car-Following Model Output

The time space diagram below illustrates an emergency braking maneuver for the leading vehicle



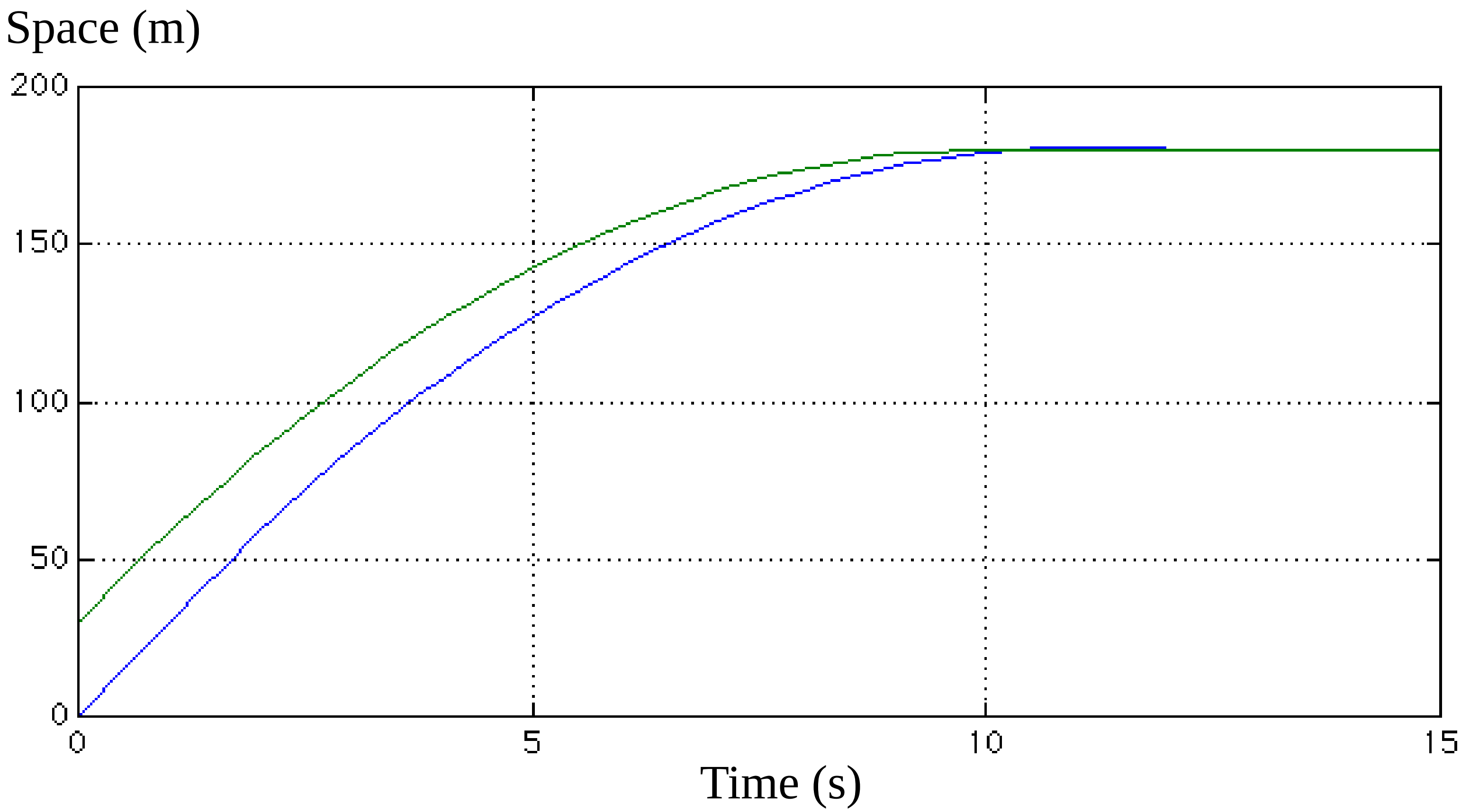
Car-Following Model with Delay

A pure transport delay block is added to the original model simulating the transport lag dynamics of a man-machine system



Output of the Car-Following Model with Delay

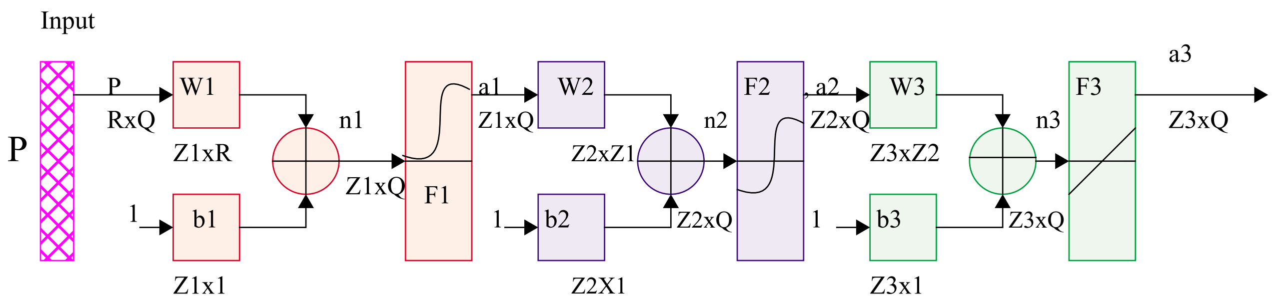
- The output suggests that a collision occurs with τ =0.75s.



Brief on C/C++ Compiler

- Converts MATLAB M-Files to C and C++ code
- Using this toolbox all graphics and math code can be converted to develop standalone applications
- Typically requires MATLAB compiler and the MATLAB C/C++ library
- New features in version 5.3 also convert cell and structure arrays
- My limited experience with this toolbox suggest substantial gains in speed if no vector operations have been implemented in the code
- Vector operations show very little improvement when the code is compiled

Neural Network Toolbox



- Provides 100+ functions to simplify the training, generalization and implementation of artificial neural networks
- The functional implementation of the ANN toolbox is just through a series of MATLAB functions
- All ANN functions are execued from the command line

Relevant Functions Provided

- Layer initialization functions (Nguyen-Widrow)
- Learning functions (gradients, perceptron, Widrow-Hoff)
- Analysis functions (max. learning rate, error surface)
- Line search functions (Golden section, backtraking)
- Network functions (competitive, feed-forward wth backpropagation)
- Performance functions (mean absolute error, SEE)
- Training functions (BFGS, Bayesion regularization, Fletchel-Powell, Lavenberg-Marquardt, etc.)
- Transfer functions (log signmoid, tangent sigmoid, etc.)

Example of ANN Using MATLAB

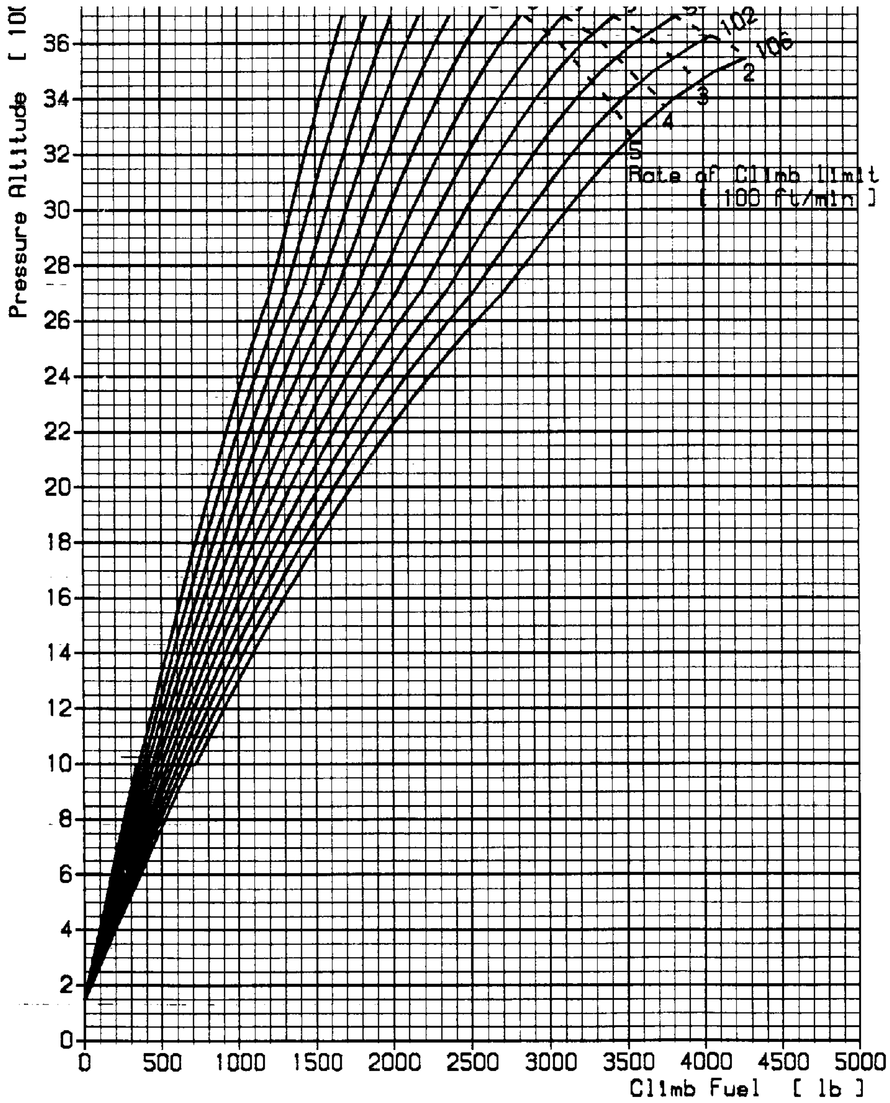
Problem:

- To estimate aircraft fuel consumption based on aircraft performance parameters
- Implementation on fast-time simulation models (SIMMOD, RAMS, etc.)

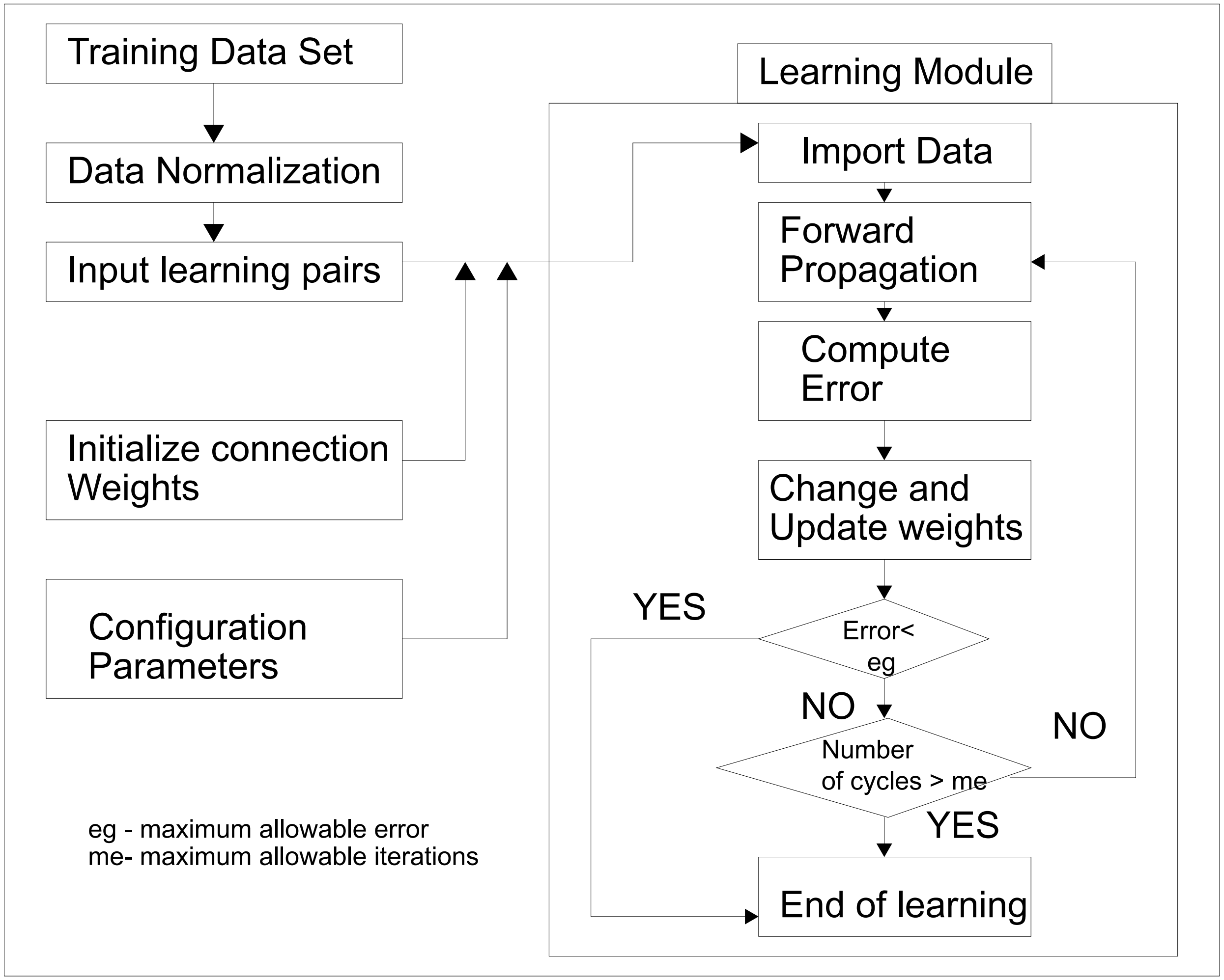
Solution

- Prototype model using MATLAB's neural Network Toolbox
- Verify the accuracy of the model with the current state-of-the art

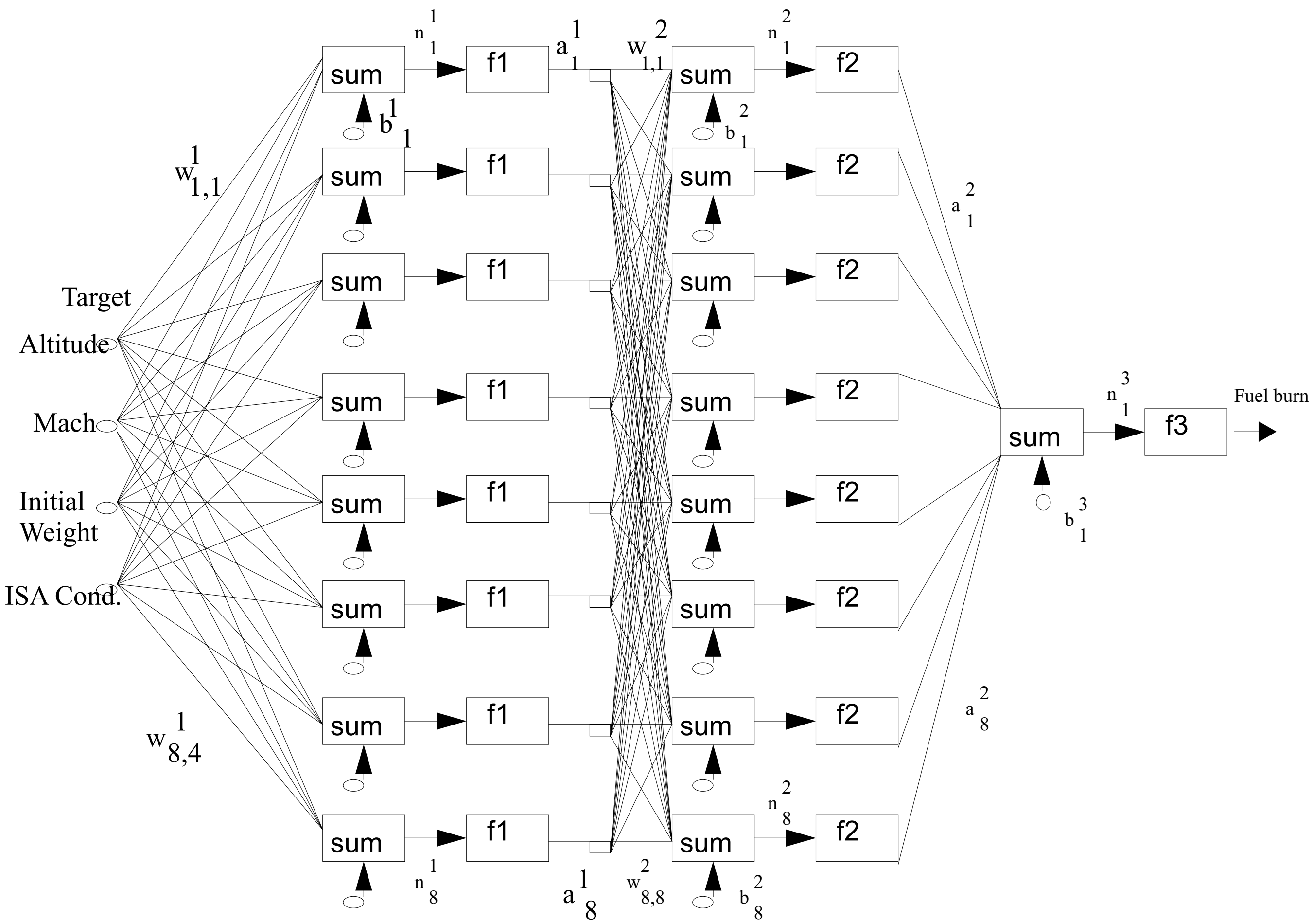
Sample Data Presented in Flight Manual



Modeling Approach



Selected ANN Topology



Where f1 - log-sigmoid transfer function
f2 - tansigmoid transfer function
f3 - purelin transfer function

ANN Data

The data sets used to develop and train the ANN are shown below. Generalization of the ANN was conducted using another (random) data set

Flight Phase	Number of Testing Points
Takeoff and Climbout	Nor applicable (linear regression used instead)
Climb to Cruise Altitude	850 (Fuel) 850 (Distance)
Cruise	805
Descent	1210 (Fuel) 140 (Distance)

Summary of Results (Fokker F100)

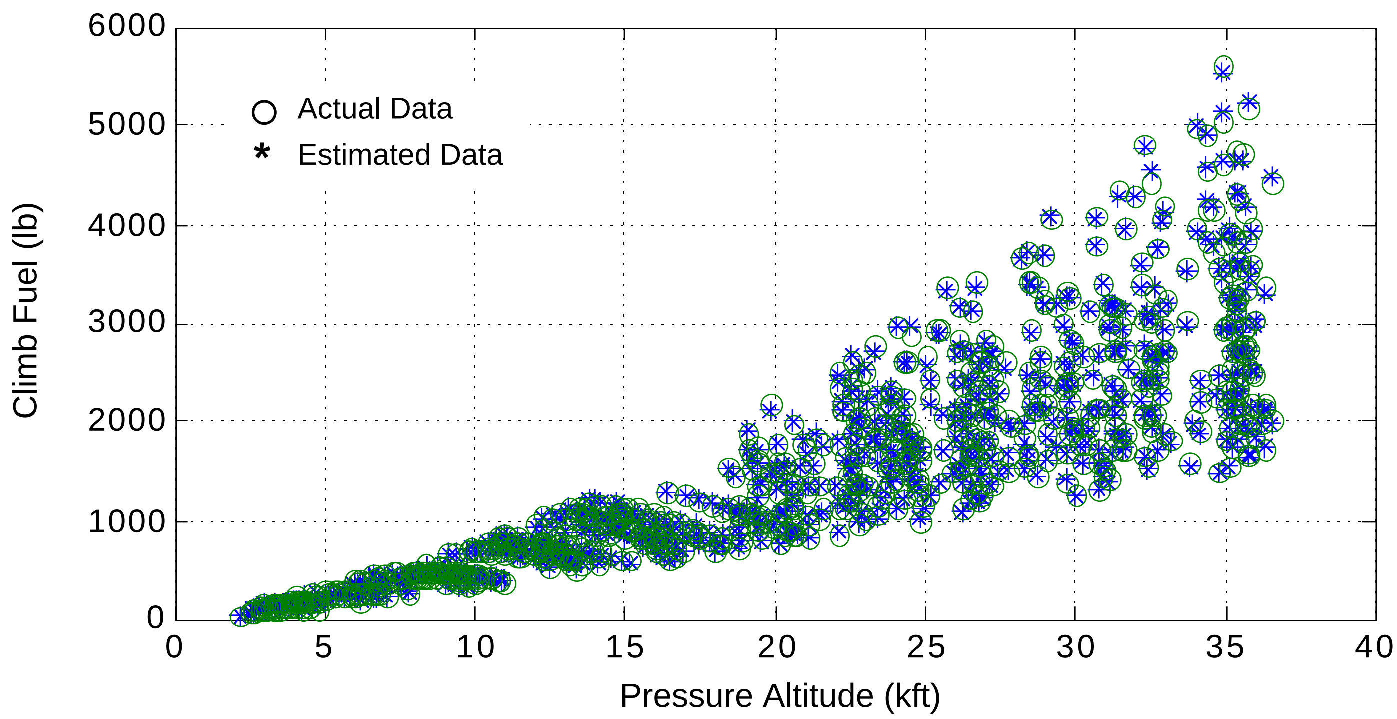
The results shown in the table illustrate the accuracy of the model

Flight Phase	Mean Error (%)	Standard Deviation (%)	Null Hypothesis (t-test at $\alpha = 0.01$)
Climb • Distance • Fuel	0.377 1.026	0.305 0.190	Accept Accept
Cruise Specific Air Range	-0.034	0.334	Accept

Flight Phase	Mean Error (%)	Standard Deviation (%)	Null Hypothesis (t-test at $\alpha = 0.01$)
Descent • Distance • Fuel	1.760 1.423	1.860 1.177	Accept Accept

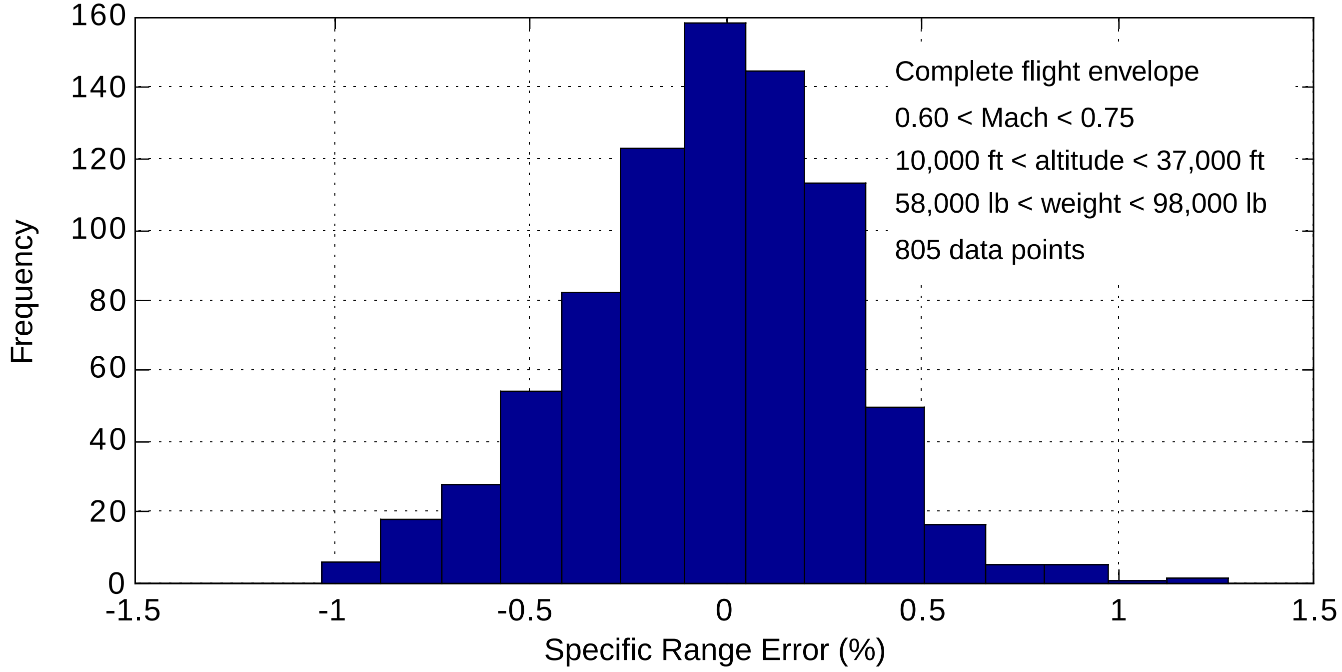
Sample Results (Climb Fuel)

- The results of training an ANN with 3 layers are shown below



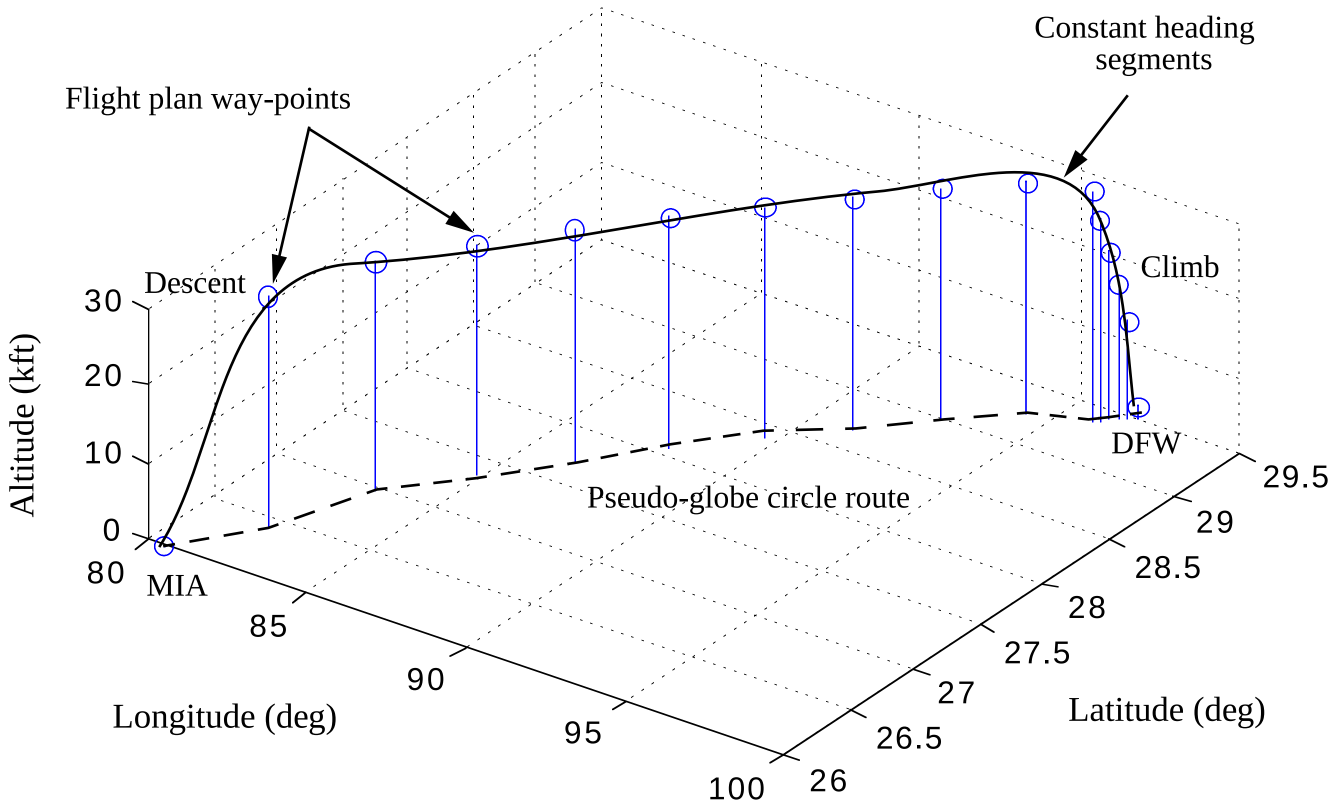
Absolute Error of ANN

The errors of the ANN are very small as depicted in this graph for SFC



Application of the Model to Complete Flight Plans

The ANN model developed has been applied to a generic flight trajectory generator with very accurate results



Complete Flight Plan Correlation

Flight	Cruise Flight Level (FL)	Distance (nm) / Time (hr)	Flight Manual Fuel Burn (lb)	Neural Net Fuel Burn (lb)	Percent Difference (%)
ROA[a]-MDW[b]	280	448 / 1:08	6,457	6,546	1.37
	310	448 / 1:10	6,360	6,330	0.46
MIA[c]-DFW[d]	310	972 / 2:24	11,851	11,865	0.12
	350	972 / 2:13	11,510	11,544	0.29
ROA-LGA[e]	290	352 / 0:57	5,298	5,260	0.71
	330	352 / 0:58	5,343	5,429	1.61
ATL[f]-MIA	290	518 / 1:20	6,990	7,047	0.80
	330	518 / 1:21	7,009	7,082	1.04

a. ROA - Roanoke Regional Airport (Virginia)
b. MDW - Midway Airport (Illinois)
c. MIA - Miami International (Florida)
d. DFW - Dallas-Forth Worth International (Texas)
e. LGA - Laguardia Airport (New York)
f. ATL - Atlanta Hartsfield International Airport (Georgia)

Sample Source Code

```
% Initialize Weights and Biasis

nns = 8; % Number of Neurons in each layer
nns2 = 8;
 %***********************************
%       For Climb Distance    **
%***********************************
[W31_cb_d,b31_cb_d,W32_cb_d,b32_cb_d,W33_cb_d,
  b33_cb_d ]=initff(P1_cbd,nns,'logsig',nns2 ...
,'tansig',T1_cbd,'purelin');

% Taining of the neural networks using Lavenberg-
  Marquardt Alogrithm
```

```
[W31_cb_d,b31_cb_d,W32_cb_d,b32_cb_d,W33_cb_
  d,b33_cb_d ]= trainlm(W31_cb_d,b31_cb_d,'logsig' ...
,W32_cb_d,b32_cb_d,'tansig',W33_cb_d,b33_cb_d,'pure
  lin',P_cbd,Ta_cbd,tp);

%*********************************
%       For Climb Fuel       **
%*********************************
[W31_cb_f,b31_cb_f,W32_cb_f,b32_cb_f,W33_cb_f,b3
  3_cb_f ]=initff(P1_cbf,nns,'logsig',nns2,'tansig' ...
,T1_cbf,'purelin');

% Taining of the neural networks using Lavenberg-
  Marquardt Alogrithm

[W31_cb_f,b31_cb_f,W32_cb_f,b32_cb_f,W33_cb_f,b3
  3_cb_f ]=
  trainlm(W31_cb_f,b31_cb_f,'logsig',W32_cb_f ...
```

```
,b32_cb_f,'tansig',W33_cb_f,b33_cb_f,'purelin',P_cbf,
  Ta_cbf,tp);
```