Simulation in Simulink

Bruce A. Desmarais

Odum Institute for Research in Social Science

February 14, 2009

Welcome!

Who am I?

Welcome!

Who am I?

- Bruce Desmarais; Red Sox Fan and PhD student in the UNC Poli Sci Dept.

Welcome!

Who am I?

- Bruce Desmarais; Red Sox Fan and PhD student in the UNC Poli Sci Dept.
- Extensive experience with matrix algebra, simulation and statistical computing in Matlab and other software/languages

Welcome!

Who am I?

- Bruce Desmarais; Red Sox Fan and PhD student in the UNC Poli Sci Dept.
- Extensive experience with matrix algebra, simulation and statistical computing in Matlab and other software/languages
- No formal training in the instruction of Matlab (comments/suggestions will be much appreciated)

What will we cover?

Welcome!

Who am I?

- Bruce Desmarais; Red Sox Fan and PhD student in the UNC Poli Sci Dept.
- Extensive experience with matrix algebra, simulation and statistical computing in Matlab and other software/languages
- No formal training in the instruction of Matlab (comments/suggestions will be much appreciated)

What will we cover?

- Basic System design and Simulation in Simulink

Welcome!

Who am I?

- Bruce Desmarais; Red Sox Fan and PhD student in the UNC Poli Sci Dept.
- Extensive experience with matrix algebra, simulation and statistical computing in Matlab and other software/languages
- No formal training in the instruction of Matlab (comments/suggestions will be much appreciated)

What will we cover?

- Basic System design and Simulation in Simulink
- Manipulation of Simulink Model files

Welcome!

Who am I?

- Bruce Desmarais; Red Sox Fan and PhD student in the UNC Poli Sci Dept.
- Extensive experience with matrix algebra, simulation and statistical computing in Matlab and other software/languages
- No formal training in the instruction of Matlab (comments/suggestions will be much appreciated)

What will we cover?

- Basic System design and Simulation in Simulink
- Manipulation of Simulink Model files
- Simulink Interface with Matlab

Welcome!

Who am I?

- Bruce Desmarais; Red Sox Fan and PhD student in the UNC Poli Sci Dept.
- Extensive experience with matrix algebra, simulation and statistical computing in Matlab and other software/languages
- No formal training in the instruction of Matlab (comments/suggestions will be much appreciated)

What will we cover?

- Basic System design and Simulation in Simulink
- Manipulation of Simulink Model files
- Simulink Interface with Matlab
- Examples

What is Simulink?

What is Simulink?

- Graphical extension to MATLAB for modeling and simulation of systems; Similar to Winbugs

What is Simulink?

- Graphical extension to MATLAB for modeling and simulation of systems; Similar to Winbugs
- Systems drawn on screen as block diagrams

What is Simulink?

- Graphical extension to MATLAB for modeling and simulation of systems; Similar to Winbugs
- Systems drawn on screen as block diagrams
- Integrated with MATLAB

What is Simulink?

- Graphical extension to MATLAB for modeling and simulation of systems; Similar to Winbugs
- Systems drawn on screen as block diagrams
- Integrated with MATLAB
- Combines intuitive graphical-user interface with excellent speed and storage capabilities

What is Simulink?

- Graphical extension to MATLAB for modeling and simulation of systems; Similar to Winbugs
- Systems drawn on screen as block diagrams
- Integrated with MATLAB
- Combines intuitive graphical-user interface with excellent speed and storage capabilities
- Is free to you!! (or included in tuition; however you want to look at it)

New Model...First Steps

- Open Matlab

New Model...First Steps

- Open Matlab
- Type `simulink`

New Model...First Steps

- Open Matlab
- Type `simulink`
- Simulink library browser pops up

New Model...First Steps

- Open Matlab
- Type `simulink`
- Simulink library browser pops up
- Start a new model file `Crtl + N`

New Model...First Steps

- Open Matlab
- Type `simulink`
- Simulink library browser pops up
- Start a new model file `Crtl + N`
- The blank file is where model will be built..save it

New Model...First Steps

- Open Matlab
- Type `simulink`
- Simulink library browser pops up
- Start a new model file `Crtl + N`
- The blank file is where model will be built..save it
- Libraries contain building blocks

Blocks

Model is a collection of blocks connected by lines. There are a number of different classes of blocks:

Blocks

Model is a collection of blocks connected by lines. There are a number of different classes of blocks:

- Sources: Used to generate signals

Blocks

Model is a collection of blocks connected by lines. There are a number of different classes of blocks:

- Sources: Used to generate signals
- Sinks: Used to output

Blocks

Model is a collection of blocks connected by lines. There are a number of different classes of blocks:

- Sources: Used to generate signals
- Sinks: Used to output
- Discrete: Linear, discrete-time system elements (transfer-functions)

Blocks

Model is a collection of blocks connected by lines. There are a number of different classes of blocks:

- Sources: Used to generate signals
- Sinks: Used to output
- Discrete: Linear, discrete-time system elements (transfer-functions)
- Nonlinear: Arbitrary nonlinear operators

Blocks

Model is a collection of blocks connected by lines. There are a number of different classes of blocks:

- Sources: Used to generate signals
- Sinks: Used to output
- Discrete: Linear, discrete-time system elements (transfer-functions)
- Nonlinear: Arbitrary nonlinear operators
- Connections: Multiplex, Demultiplex

Blocks

Model is a collection of blocks connected by lines. There are a number of different classes of blocks:

- Sources: Used to generate signals
- Sinks: Used to output
- Discrete: Linear, discrete-time system elements (transfer-functions)
- Nonlinear: Arbitrary nonlinear operators
- Connections: Multiplex, Demultiplex

Blocks

Model is a collection of blocks connected by lines. There are a number of different classes of blocks:

- Sources: Used to generate signals
- Sinks: Used to output
- Discrete: Linear, discrete-time system elements (transfer-functions)
- Nonlinear: Arbitrary nonlinear operators
- Connections: Multiplex, Demultiplex

Blocks have input and output terminals indicated by wedges pointing towards and away from the block respectively. Lines transmit signals from input to output.

Lets build our first model

- Click on `Source`

Lets build our first model

- Click on `Source`
- Drag the `radnom` box into the left of the model

Lets build our first model

- Click on `Source`
- Drag the `radnom` box into the left of the model
- Produces a gaussian signal with set parameters.

Lets build our first model

- Click on `Source`
- Drag the `radnom` box into the left of the model
- Produces a gaussian signal with set parameters.
- Return to library main, click on `User Defined`

Lets build our first model

- Click on `Source`
- Drag the `radnom` box into the left of the model
- Produces a gaussian signal with set parameters.
- Return to library main, click on `User Defined`
- Drag the `Matlab Fcn` into the model

Lets build our first model

- Click on `Source`
- Drag the `radnom` box into the left of the model
- Produces a gaussian signal with set parameters.
- Return to library main, click on `User Defined`
- Drag the `Matlab Fcn` into the model
- Return to library main, click on `Sinks`

Lets build our first model

- Click on `Source`
- Drag the `radnom` box into the left of the model
- Produces a gaussian signal with set parameters.
- Return to library main, click on `User Defined`
- Drag the `Matlab Fcn` into the model
- Return to library main, click on `Sinks`
- Drag `Scope` into the model, connect all

Block Manipulation

- Blocks have many sorts of parameters

Block Manipulation

- Blocks have many sorts of parameters
- To set a block's parameters, right-click on a block and select the parameters option

Block Manipulation

- Blocks have many sorts of parameters
- To set a block's parameters, right-click on a block and select the parameters option
- Our gaussian signal has mean and variance and seed parameters. Right click and set them to 0.5 and 1.5 respectively, be sure to note or set seed in rng's

Block Manipulation

- Blocks have many sorts of parameters
- To set a block's parameters, right-click on a block and select the parameters option
- Our gaussian signal has mean and variance and seed parameters. Right click and set them to 0.5 and 1.5 respectively, be sure to note or set seed in rng's
- The Matlab Function takes a function as a parameter with vector-valued argument u. Lets set it to $sin(u) + 1$.

Block Manipulation

- Blocks have many sorts of parameters
- To set a block's parameters, right-click on a block and select the parameters option
- Our gaussian signal has mean and variance and seed parameters. Right click and set them to 0.5 and 1.5 respectively, be sure to note or set seed in rng's
- The Matlab Function takes a function as a parameter with vector-valued argument u. Lets set it to $sin(u) + 1$.
- Time arguments tell blocks 'when' to act during a simulation

Block Manipulation

- Blocks have many sorts of parameters
- To set a block's parameters, right-click on a block and select the parameters option
- Our gaussian signal has mean and variance and seed parameters. Right click and set them to 0.5 and 1.5 respectively, be sure to note or set seed in rng's
- The Matlab Function takes a function as a parameter with vector-valued argument u. Lets set it to $sin(u) + 1$.
- Time arguments tell blocks 'when' to act during a simulation
- The scope box

A First Run

A First Run

1. Open a useful monitor of the simulation by double clicking on the scope box

A First Run

1. Open a useful monitor of the simulation by double clicking on the scope box
2. Now lets run the simulation...

A First Run

1. Open a useful monitor of the simulation by double clicking on the `scope` box
2. Now lets run the simulation...
3. First set the simulation parameters typing `Ctrl + E`

A First Run

1. Open a useful monitor of the simulation by double clicking on the `scope` box
2. Now lets run the simulation...
3. First set the simulation parameters typing `Ctrl + E`
4. Many options; lets set the time from 0 to 25

A First Run

1. Open a useful monitor of the simulation by double clicking on the `scope` box
2. Now lets run the simulation...
3. First set the simulation parameters typing `Ctrl + E`
4. Many options; lets set the time from 0 to 25
5. Now click on `Simulation`, `Start`

Output to Matlab

Output to Matlab

1. We often want the simulation data in Matlab

Output to Matlab

1. We often want the simulation data in Matlab
2. Go to the library main, click on `sinks`

Output to Matlab

1. We often want the simulation data in Matlab
2. Go to the library main, click on `sinks`
3. Drag `simout` into the model and connect before scope

Output to Matlab

1. We often want the simulation data in Matlab
2. Go to the library main, click on `sinks`
3. Drag `simout` into the model and connect before scope
4. Right click and call the object output

Output to Matlab

1. We often want the simulation data in Matlab
2. Go to the library main, click on `sinks`
3. Drag `simout` into the model and connect before scope
4. Right click and call the object output
5. Returns a matlab structure...run the sim and lets take a look

Output More to Matlab

Output More to Matlab

1. Suppose we want to get the signal at different points in the simulation

Output More to Matlab

1. Suppose we want to get the signal at different points in the simulation
2. Drag another `simout` onto the screen

Output More to Matlab

1. Suppose we want to get the signal at different points in the simulation
2. Drag another `simout` onto the screen
3. Lets grab the original gaussian signal

Output More to Matlab

1. Suppose we want to get the signal at different points in the simulation
2. Drag another `simout` onto the screen
3. Lets grab the original gaussian signal
4. Name the second simout `output`

Output More to Matlab

1. Suppose we want to get the signal at different points in the simulation
2. Drag another `simout` onto the screen
3. Lets grab the original gaussian signal
4. Name the second simout `output`
5. Connect to the line between the signal and the function..run the simulation

Output to File

Output to File

1. The best way to assure your simulated data is not lost is to save it immediately

Output to File

1. The best way to assure your simulated data is not lost is to save it immediately
2. Go into `sinks` again and select a `untitled.mat` file.

Output to File

1. The best way to assure your simulated data is not lost is to save it immediately
2. Go into `sinks` again and select a `untitled.mat` file.
3. Right click to set file extension...will write to working directory if no extension is provided

Output to File

1. The best way to assure your simulated data is not lost is to save it immediately
2. Go into `sinks` again and select a `untitled.mat` file.
3. Right click to set file extension...will write to working directory if no extension is provided
4. The signal can be saved in different times along the model path

Output to File

1. The best way to assure your simulated data is not lost is to save it immediately
2. Go into `sinks` again and select a `untitled.mat` file.
3. Right click to set file extension...will write to working directory if no extension is provided
4. The signal can be saved in different times along the model path
5. Somewhat inconvenient, every time you want a new position on the path, you need a new file.

Using Data From Matlab Workspace I

There are many reasons we could want to use data from the matlab workspace

Using Data From Matlab Workspace I

There are many reasons we could want to use data from the matlab workspace

1. We may want to use data in memory rather than generate it

Using Data From Matlab Workspace I

There are many reasons we could want to use data from the matlab workspace

1. We may want to use data in memory rather than generate it
2. It may be useful to use real-world data in the model.

Using Data From Matlab Workspace I

There are many reasons we could want to use data from the matlab workspace

1. We may want to use data in memory rather than generate it
2. It may be useful to use real-world data in the model.
3. It may be difficult to use specific generators/rng streams in the simulation

Using Data From Matlab Workspace I

There are many reasons we could want to use data from the matlab workspace

1. We may want to use data in memory rather than generate it
2. It may be useful to use real-world data in the model.
3. It may be difficult to use specific generators/rng streams in the simulation
4. We could use the inputs/outputs of previous simulations as inputs to the current simulation

Using Data From Matlab Workspace II

All of the objects in the workspace are available

Using Data From Matlab Workspace II

All of the objects in the workspace are available

1. First, lets put something in the workspace, issue the command
   `x = [1:100; rand(1,100)]'`

Using Data From Matlab Workspace II

All of the objects in the workspace are available

1. First, lets put something in the workspace, issue the command `x = [1:100; rand(1,100)]'`
2. Creates a 100x2 matrix with the first column indicating the time stamp

Using Data From Matlab Workspace II

All of the objects in the workspace are available

1. First, lets put something in the workspace, issue the command `x = [1:100; rand(1,100)]'`
2. Creates a 100x2 matrix with the first column indicating the time stamp
3. Right Click and delete the random generator

Using Data From Matlab Workspace II

All of the objects in the workspace are available

1. First, lets put something in the workspace, issue the command `x = [1:100; rand(1,100)]'`
2. Creates a 100x2 matrix with the first column indicating the time stamp
3. Right Click and delete the random generator
4. Now go to `Sources` and drag `simin`

Using Data From Matlab Workspace II

All of the objects in the workspace are available

1. First, lets put something in the workspace, issue the command `x = [1:100; rand(1,100)]'`
2. Creates a 100x2 matrix with the first column indicating the time stamp
3. Right Click and delete the random generator
4. Now go to `Sources` and drag `simin`
5. Right click on `simin` and tell it to use x

Using Data From Matlab Workspace II

All of the objects in the workspace are available

1. First, lets put something in the workspace, issue the command `x = [1:100; rand(1,100)]'`
2. Creates a 100x2 matrix with the first column indicating the time stamp
3. Right Click and delete the random generator
4. Now go to `Sources` and drag `simin`
5. Right click on `simin` and tell it to use x
6. Now lets set the simulation time to 0-250 and run it

Using Data From Matlab Workspace II

All of the objects in the workspace are available

1. First, lets put something in the workspace, issue the command `x = [1:100; rand(1,100)]'`
2. Creates a 100x2 matrix with the first column indicating the time stamp
3. Right Click and delete the random generator
4. Now go to `Sources` and drag `simin`
5. Right click on `simin` and tell it to use x
6. Now lets set the simulation time to 0-250 and run it
7. Notice anything odd?...extrapolation

Scope Plot of Multiple Characteristics

It is easy to visualize how things move together

Scope Plot of Multiple Characteristics

It is easy to visualize how things move together

1. Go to search, enter `mux`

Scope Plot of Multiple Characteristics

It is easy to visualize how things move together

1. Go to search, enter `mux`
2. This block concatenates multiple signals into larger vector-valued signals

Scope Plot of Multiple Characteristics

It is easy to visualize how things move together

1. Go to search, enter `mux`
2. This block concatenates multiple signals into larger vector-valued signals
3. Drag to the graphic

Scope Plot of Multiple Characteristics

It is easy to visualize how things move together

1. Go to search, enter `mux`
2. This block concatenates multiple signals into larger vector-valued signals
3. Drag to the graphic
4. Connect the input and output to the two `mux` input wedges

Scope Plot of Multiple Characteristics

It is easy to visualize how things move together

1. Go to search, enter `mux`
2. This block concatenates multiple signals into larger vector-valued signals
3. Drag to the graphic
4. Connect the input and output to the two `mux` input wedges
5. Run the simulation and observe

Scope Plot of Multiple Characteristics

It is easy to visualize how things move together

1. Go to search, enter `mux`
2. This block concatenates multiple signals into larger vector-valued signals
3. Drag to the graphic
4. Connect the input and output to the two `mux` input wedges
5. Run the simulation and observe
6. Look at the Scope

Scope Plot of Multiple Characteristics

It is easy to visualize how things move together

1. Go to search, enter `mux`
2. This block concatenates multiple signals into larger vector-valued signals
3. Drag to the graphic
4. Connect the input and output to the two `mux` input wedges
5. Run the simulation and observe
6. Look at the Scope
7. Which is which? We need a legend.

An Alternative Multiple Characteristic Plot

It is easy to visualize how things move together..again

An Alternative Multiple Characteristic Plot

It is easy to visualize how things move together..again

1. Right click in model area and choose Signal & Scope manager

An Alternative Multiple Characteristic Plot

It is easy to visualize how things move together..again

1. Right click in model area and choose Signal & Scope manager
2. Under viewers expand Simulink

An Alternative Multiple Characteristic Plot

It is easy to visualize how things move together..again

1. Right click in model area and choose Signal & Scope manager
2. Under viewers expand Simulink
3. Double Click Scope

An Alternative Multiple Characteristic Plot

It is easy to visualize how things move together..again

1. Right click in model area and choose Signal & Scope manager
2. Under viewers expand Simulink
3. Double Click Scope
4. Right click on the scope and choose `Edit Signal Connections`

An Alternative Multiple Characteristic Plot

It is easy to visualize how things move together..again

1. Right click in model area and choose Signal & Scope manager
2. Under viewers expand Simulink
3. Double Click Scope
4. Right click on the scope and choose `Edit Signal Connections`
5. Select the inputs and outputs then close.

An Alternative Multiple Characteristic Plot

It is easy to visualize how things move together..again

1. Right click in model area and choose Signal & Scope manager
2. Under viewers expand Simulink
3. Double Click Scope
4. Right click on the scope and choose `Edit Signal Connections`
5. Select the inputs and outputs then close.
6. Run the simulation and double click on the glasses

An Alternative Multiple Characteristic Plot

It is easy to visualize how things move together..again

1. Right click in model area and choose Signal & Scope manager
2. Under viewers expand Simulink
3. Double Click Scope
4. Right click on the scope and choose `Edit Signal Connections`
5. Select the inputs and outputs then close.
6. Run the simulation and double click on the glasses
7. Now right click on the plot space and select legend to get a legend