CEE 3804
Simulink and Differential Equations

Dr. A. Trani
Professor
Virginia Tech

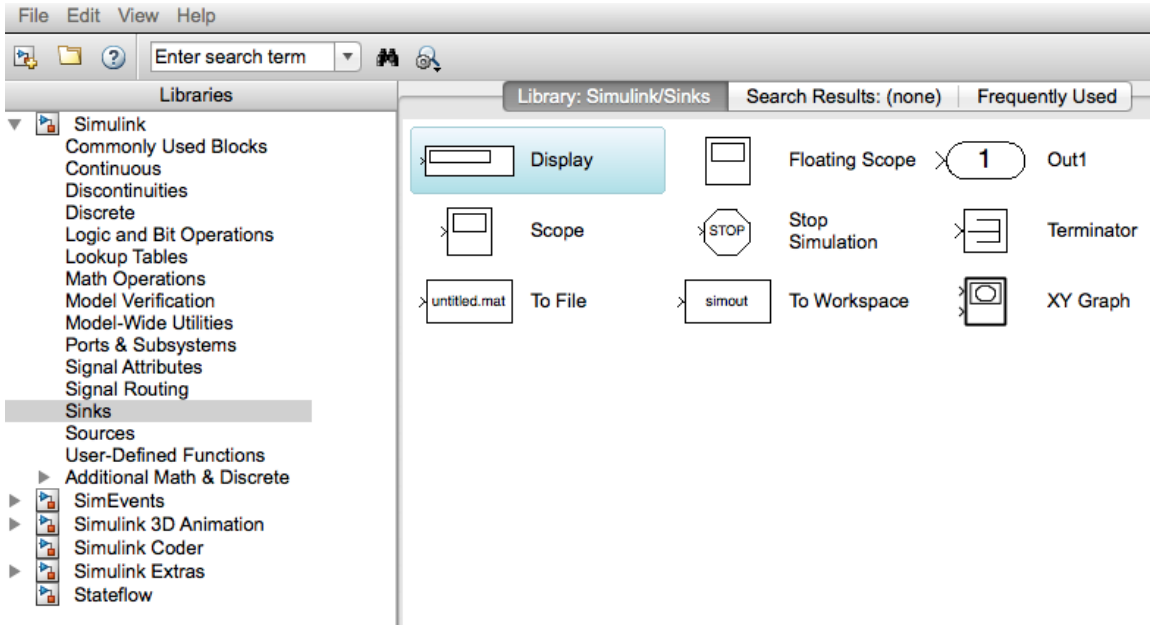
Spring 2024

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 Ordinary Differential Equations 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

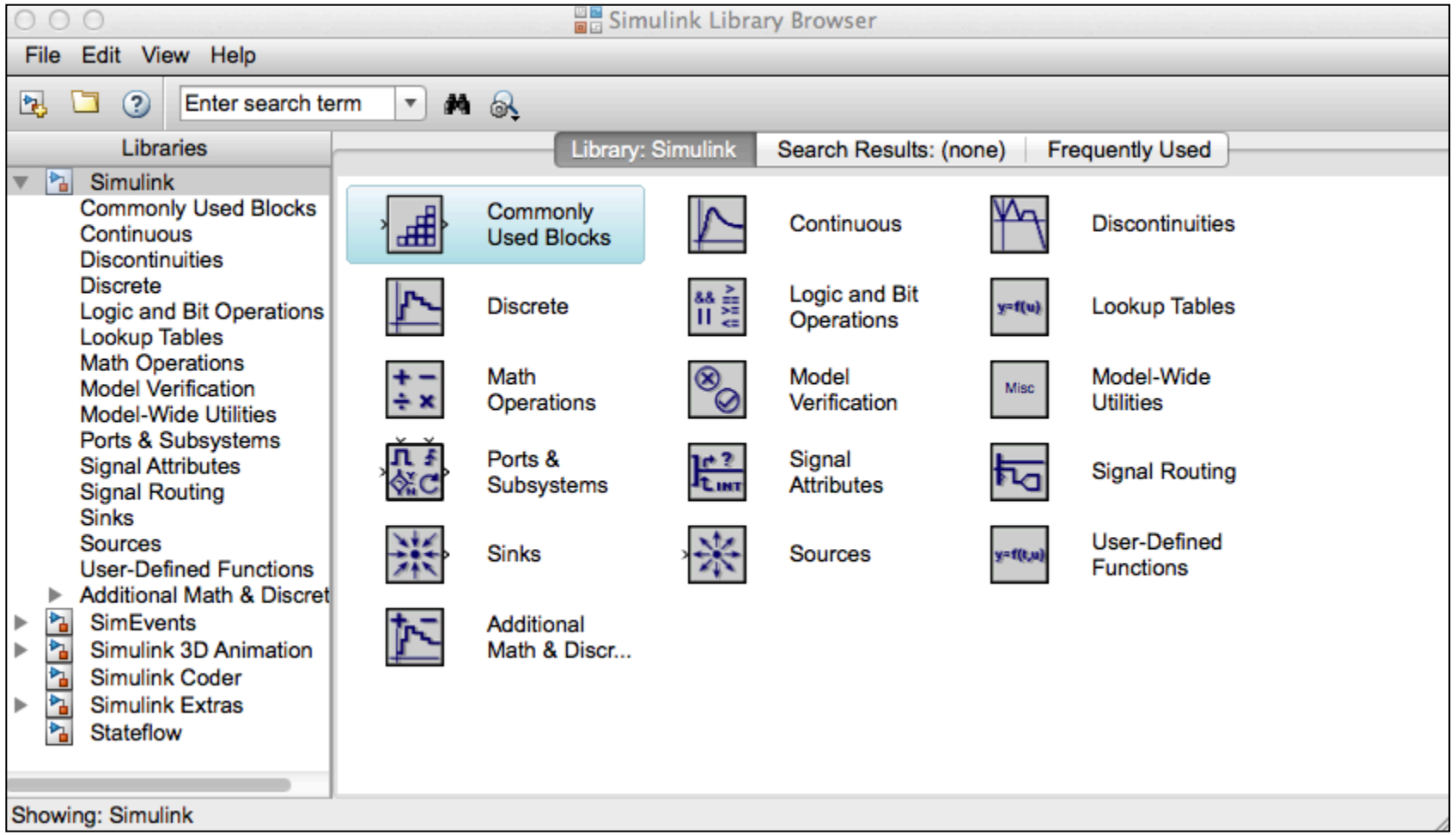
Simulink Building Blocks

- 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

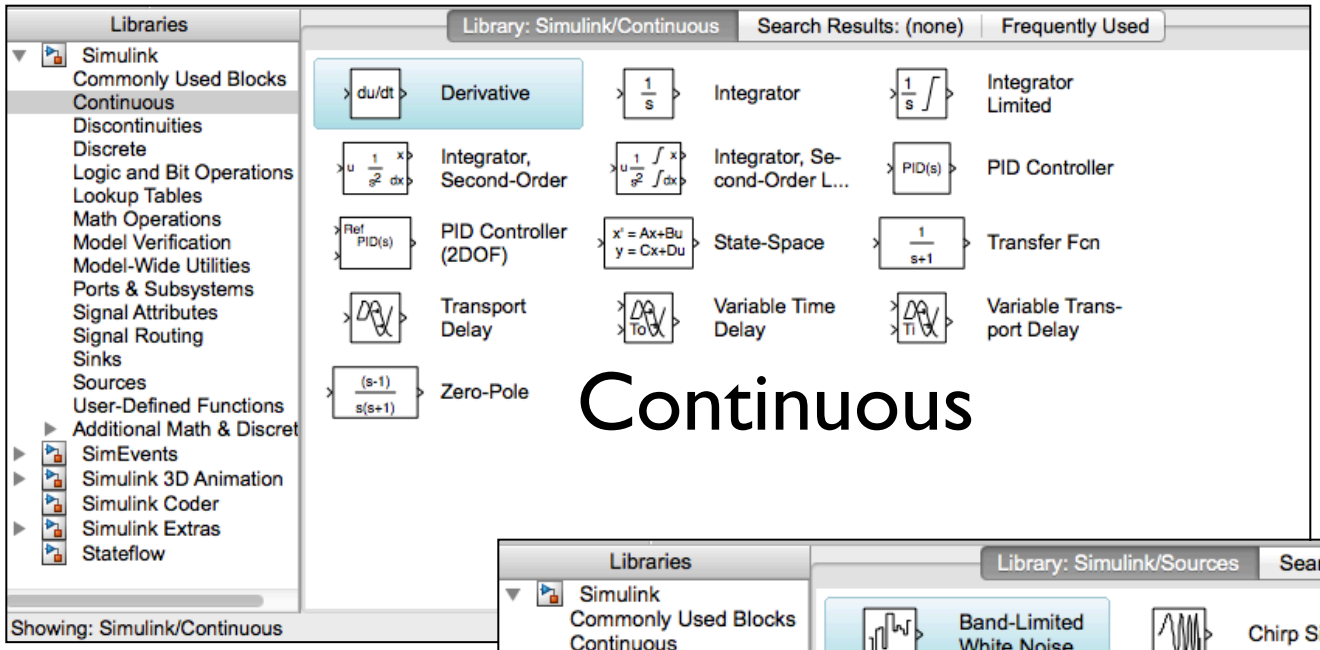


Simulink Interface

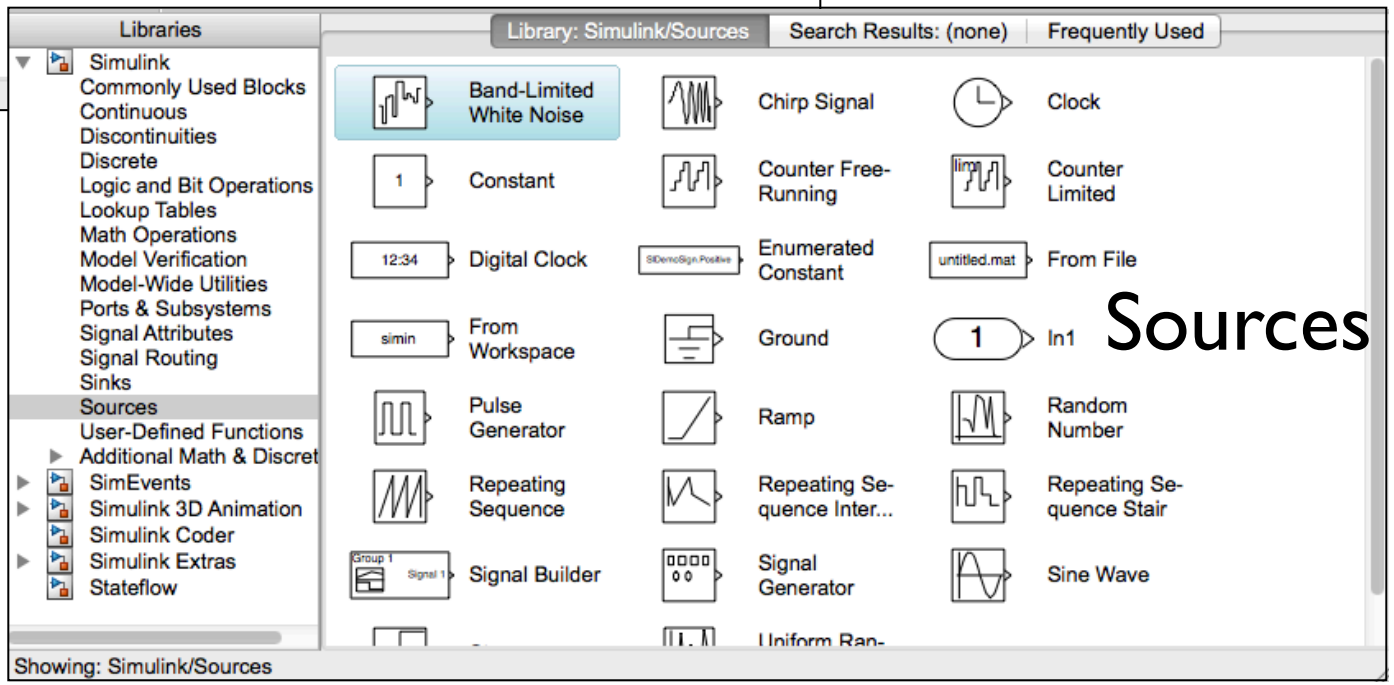
- Simulink blocks are contained and organized in libraries
  - Four of the most useful libraries are: 1) Continuous, 2) Sources, 3) Sinks and 4) Math Operations



Typical Simulink Libraries

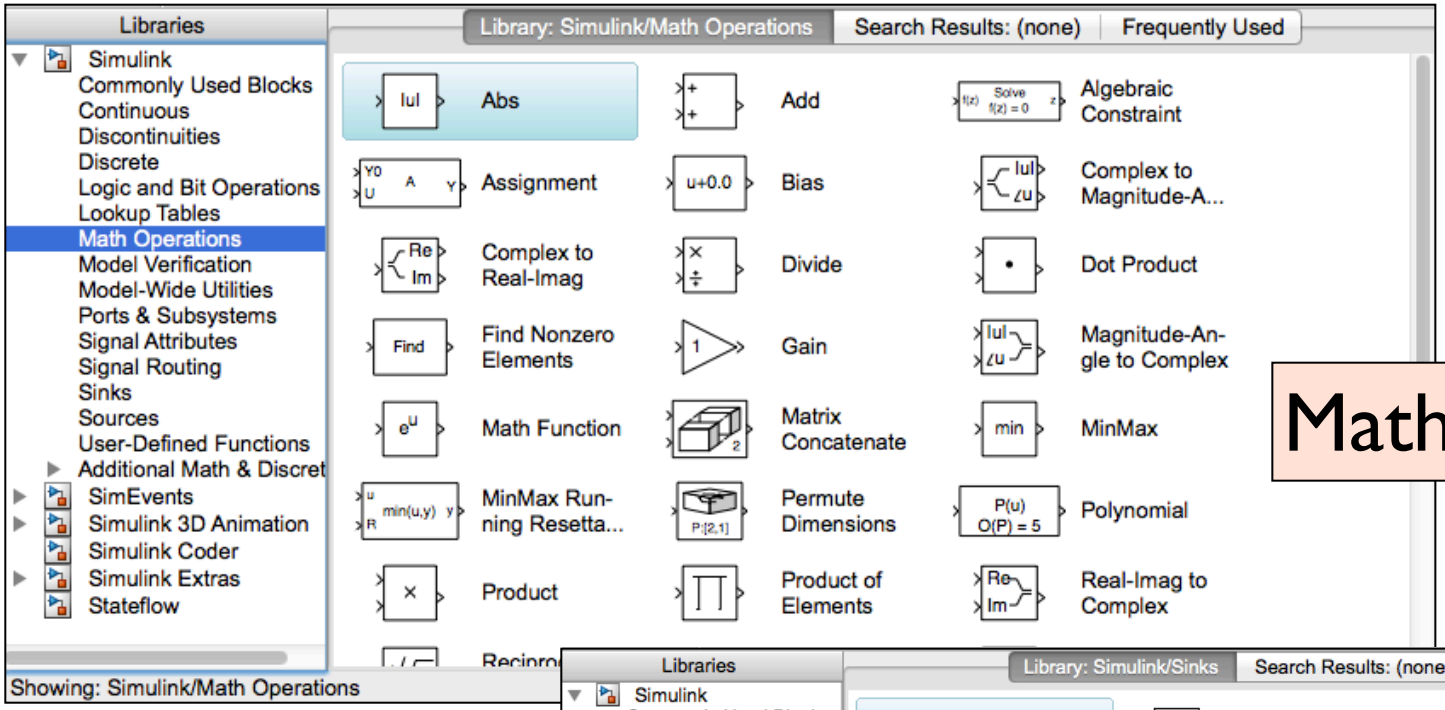


Continuous

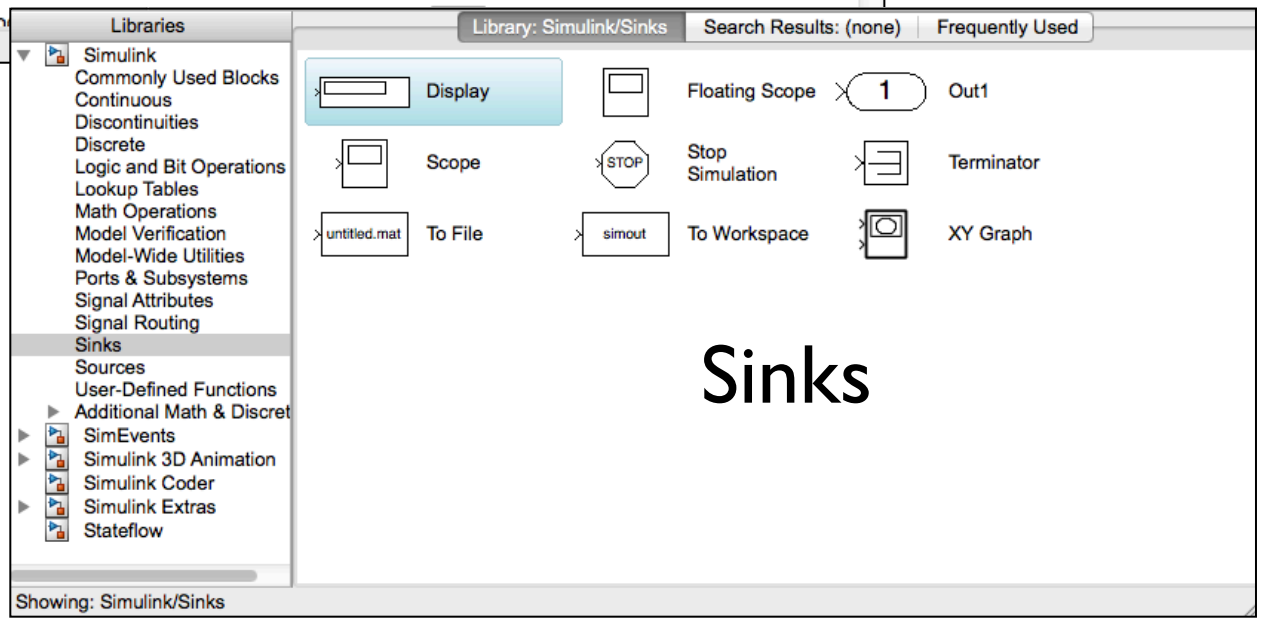


Sources

Typical Simulink Libraries



Math Operations



Sinks

Example 1. First-Order Kinematic Model

We would like to solve the first-order differential equation shown below in Simulink

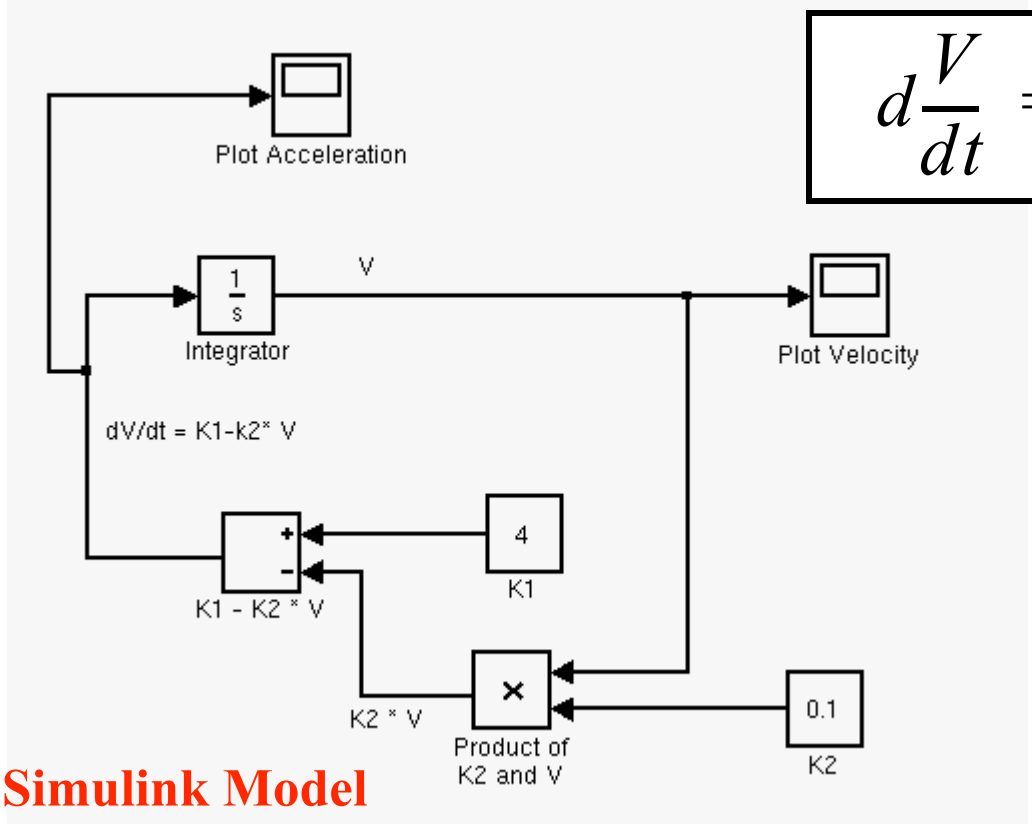
$$d\frac{V}{dt} = k_1 - k_2 V \tag{1}$$

where: V is the speed of the vehicle, k_1 and k_2 are model constants.

The values of the model parameters are: $k_1 = 4.0$ and $k_2 = 0.1$ with units for V in m/s and for $\frac{\mathrm{d}V}{\mathrm{d}t}$ in m/s^2.

Simulink Model

The following plot shows the Simulink model solution for the first order differential equation



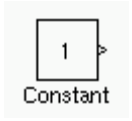
Simulink Model

$$d\frac{V}{dt} = k_1 - k_2 V$$

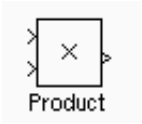
Equation

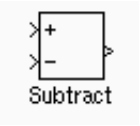
Procedure to Create a Simulink Model

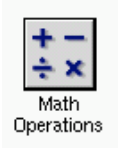
The Simulink blocks needed for this model are found in four distinct Simulink libraries:

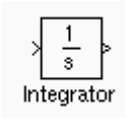
- **Constant block** 

in the Simulink Sources library 

- **Product** 

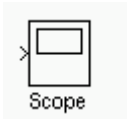
and **Subtraction** 

blocks in the Simulink Math Operations library 

- **Integrator block** 

in Simulink Continuous library 

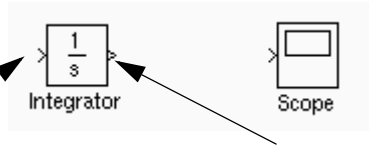
Procedure to Create a Simulink Model (cont.)

- **Scope block** 

in Simulink Sinks library 

Once the blocks are available in the new model window, they can be "wired"

Wiring a model inplies connecting the output connection of each block with the input connection of another block



Input connection of integrator block

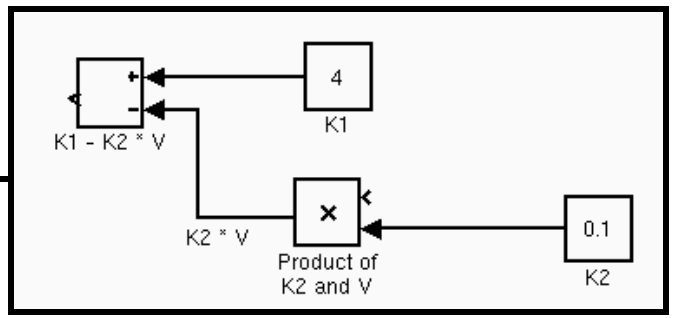
Output connection of integrator block

After wiring

Creating the Simulink Model

- Recognize the number of terms inside the differential equation to be solved. For the example below we have two terms in the right-hand side
- Each term requires a series of operations to evaluate it. Your Simulink model will require one mathematical block for each operation. For example, the product operation between k_2 and V requires a Simulink block

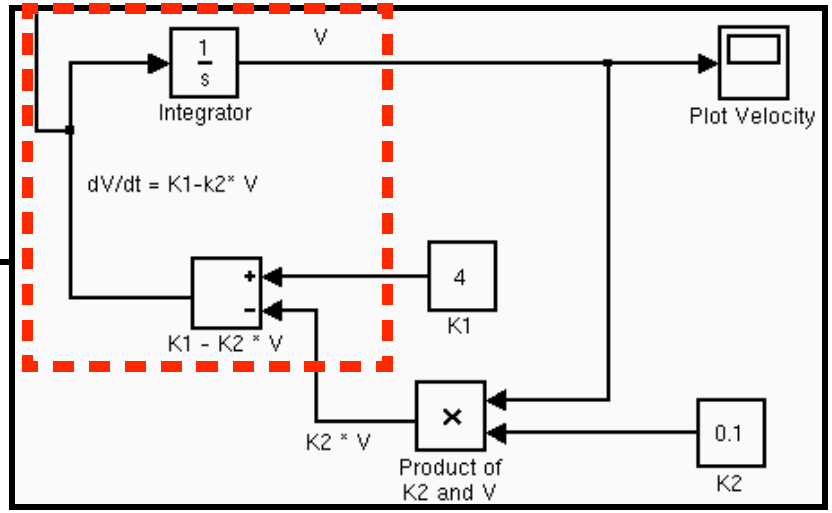
$$d\frac{V}{dt} = k_1 - k_2 V$$



Completing the Simulink Model

- In the model of interest we need to integrate the output of the term $k_1 - k_2 V$ because this is the rate of change of vehicle speed with respect to time. We accomplish this using an integrator block (see figure)

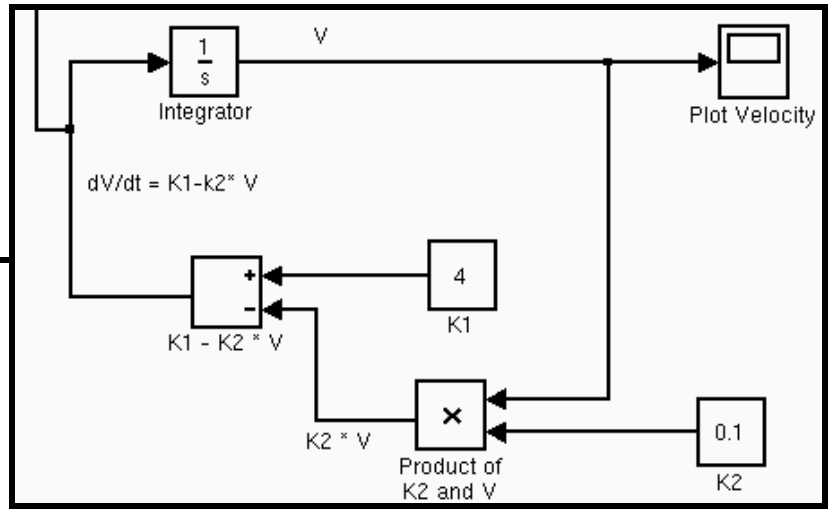
$$d\frac{V}{dt} = k_1 - k_2 V$$



Completing the Simulink Model

- The output of the integrator block is V (the vehicle speed) which is the signal needed by the product block multiplying k_2 and V. This forms a first-order negative feedback loop.

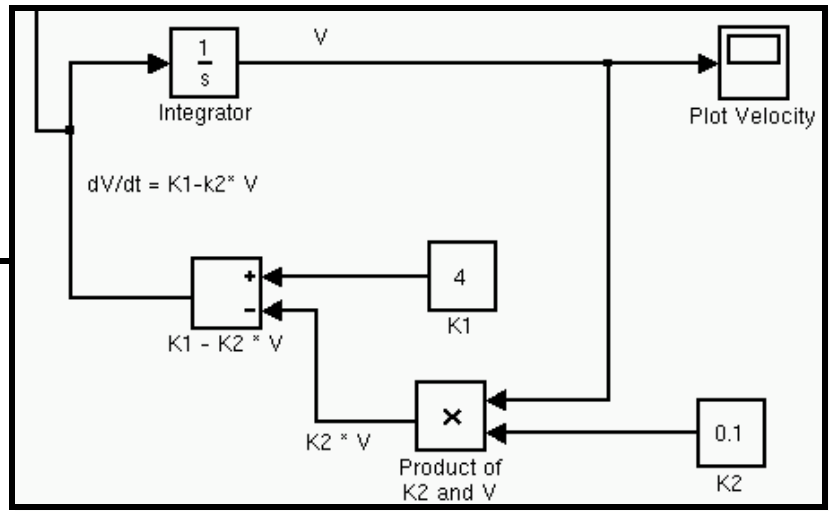
$$d\frac{V}{dt} = k_1 - k_2 V$$



Completing the Simulink Model

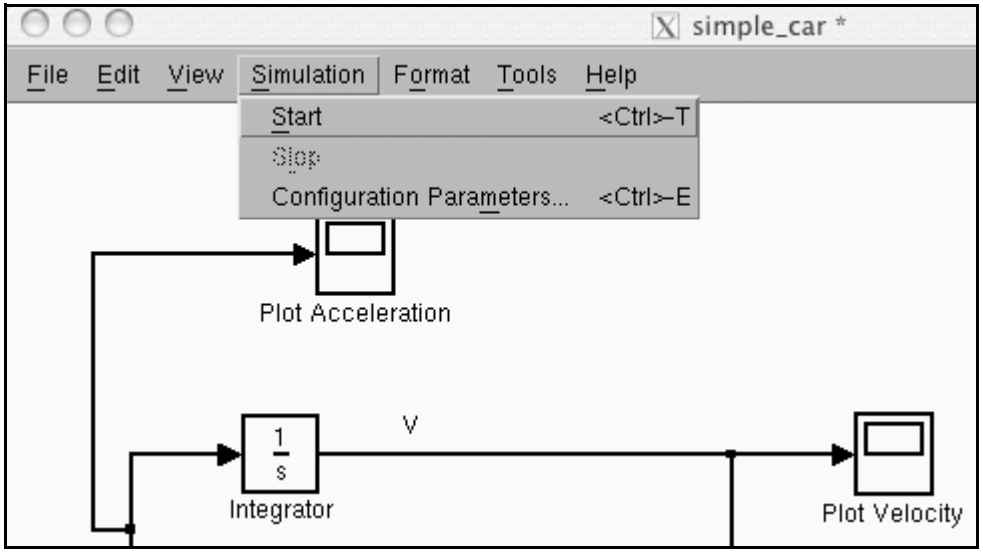
- The output of the integrator block (V) can be displayed in a scope block. This scope block is labeled "Plot Velocity" in the model

$$d\frac{V}{dt} = k_1 - k_2 V$$



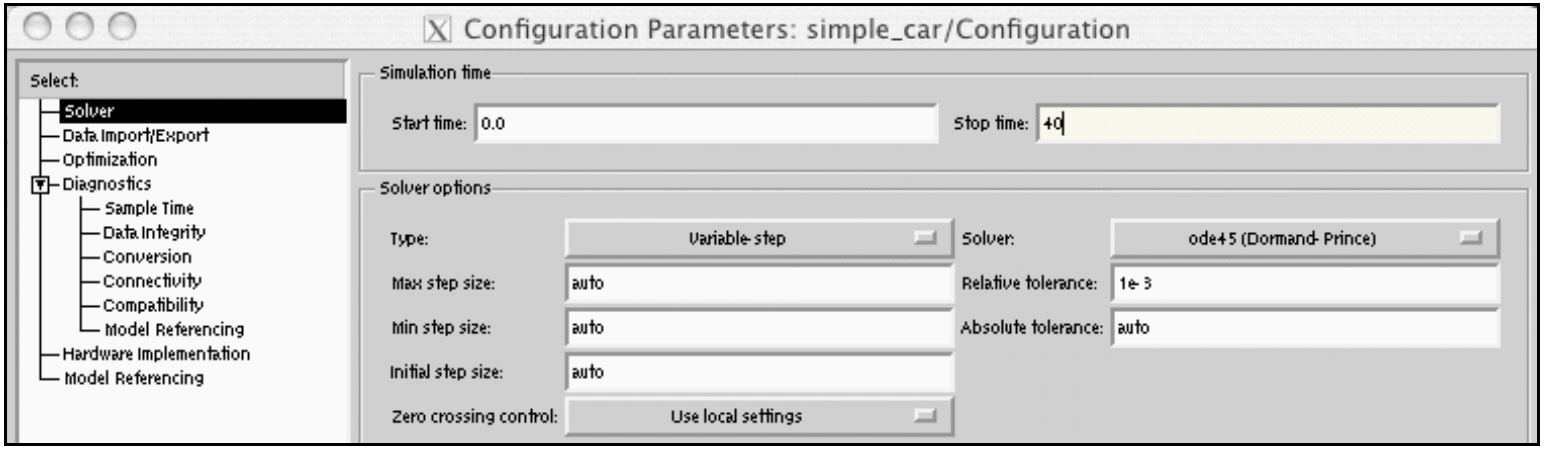
Simulation Model Settings

- Before a differential equation is solved numerically we need to define the simulation configuration parameters
- These parameters are needed to tell the solver the initial and final times of the simulation and the numerical integration procedure to be used



Simulation Model Settings (cont.)

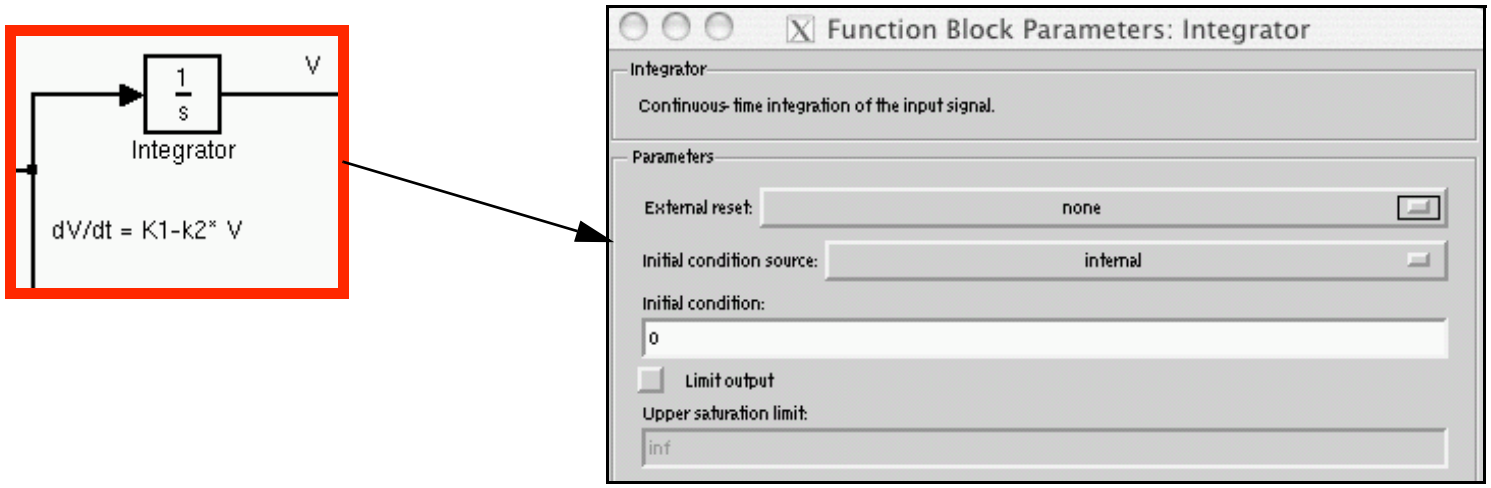
- For this simple differential equation we use a start time of zero (t=0) and 40 seconds as the stop time (t=40). This means the equation will be solved between these limits.
- The numerical differential equation solver used is a variable step ODE45 solve (default)



Initial Conditions

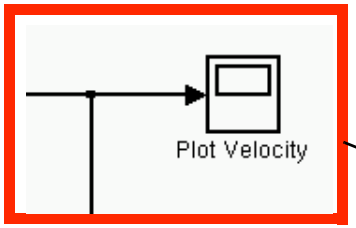
- Solving a differential equation requires the specification of initial conditions (Initial Value Problems). Initial conditions are required for all state variables of the system (e.g., variables to be integrated)

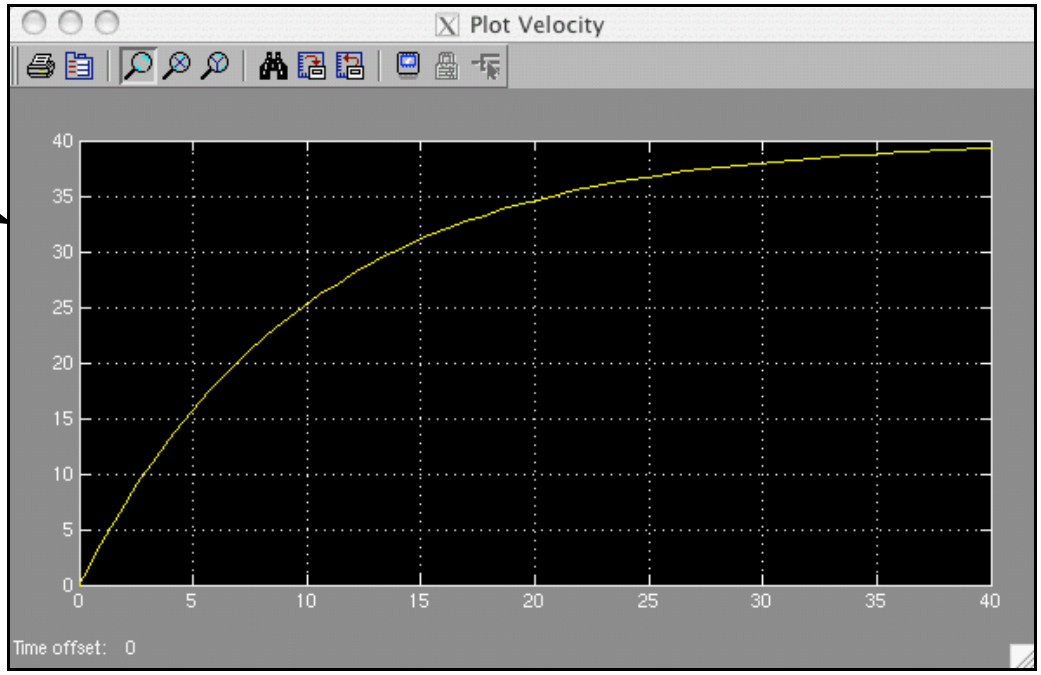
- There is only one state variable in this system (V). Specify initial conditions by opening the integrator block



Simulating and Getting Results

- To solve the differential equation just "start" the model from the "Simulation" pull down menu (in Windows there is "play" button in the Simulink window)
- See the model results opening each scope block

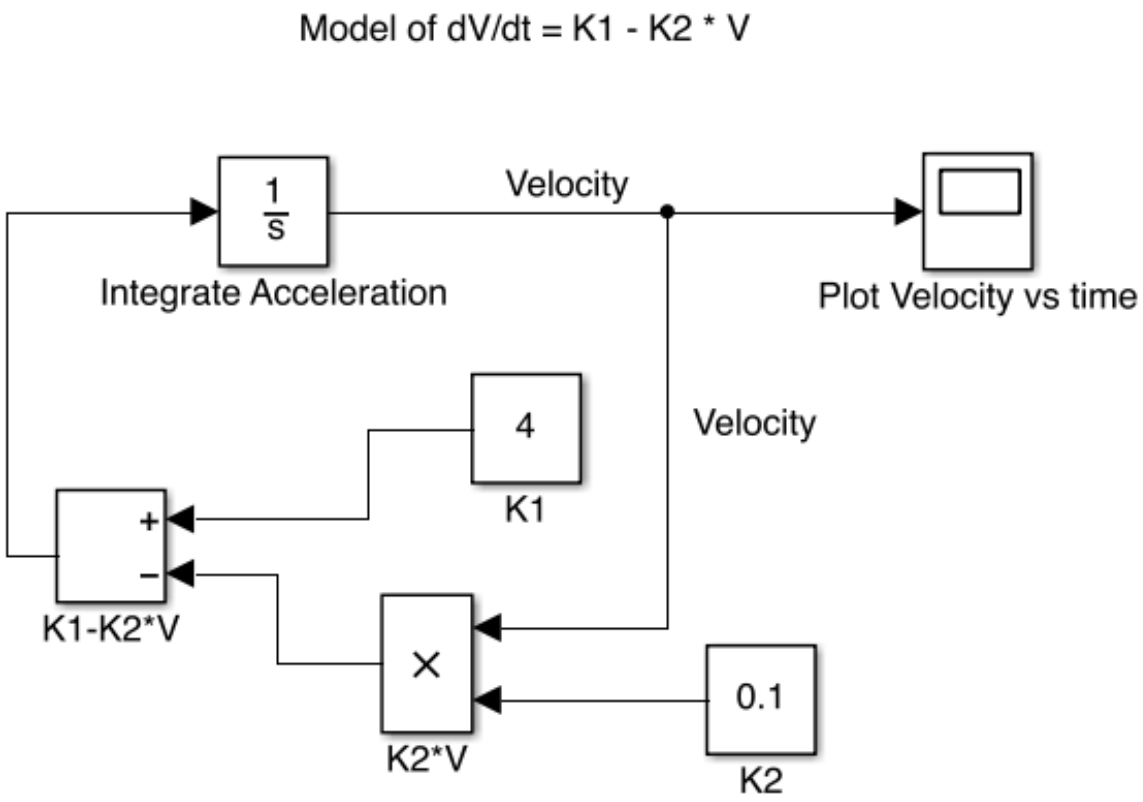




Note: All plots in a scope are functions of time

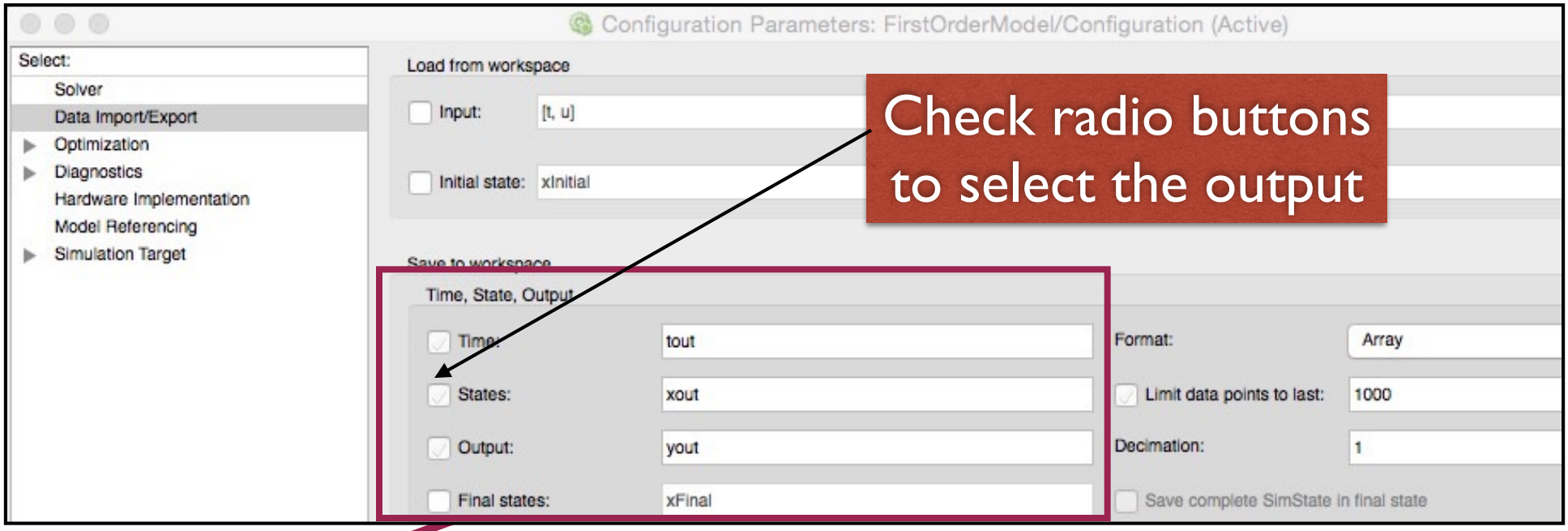
Obtaining Data from a Simulink Model

- Using the Data Export panel in Simulink
- Using a "Simout" block in the Simulink model
- Both methods will be demonstrated



Data Export Panel (in Simulink Preferences)
Matlab Release 2014

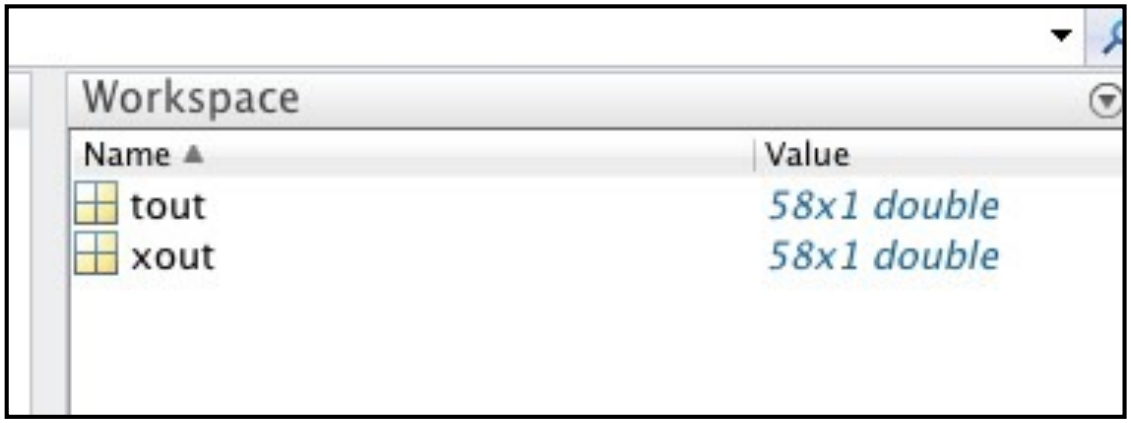
- Select the States and Output in the Data Export panel



Time = **tout** = contains values of time for state variables
States = **xout** = contains an array of state variables calculated by the model
Output = **yout** = a Simulink data structure containing both the time and state variable results

Exported State Variables

- The workspace variables out and tout are typically produced as arrays in the newer version of Matlab
- If a data structure is produced instead go back to the configuration parameters and set "arrays" as the output method



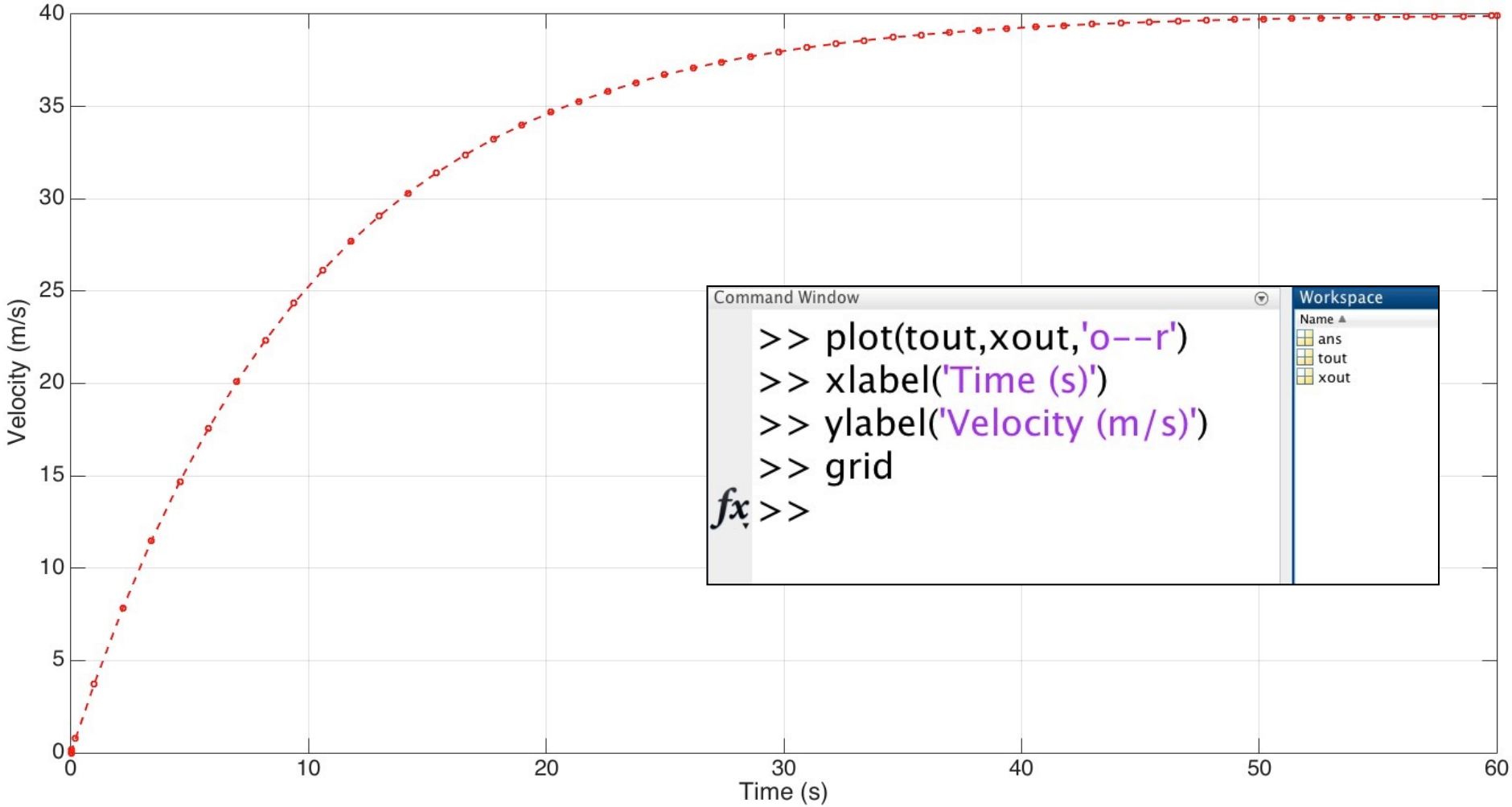
Command Window

```
>> [tout xout]

ans =

         0         0
    0.0001    0.0002
    0.0003    0.0012
    0.0016    0.0062
    0.0078    0.0313
    0.0392    0.1566
```

tout = contains values of time for state variables
xout = contains an array of state variables calculated by the model

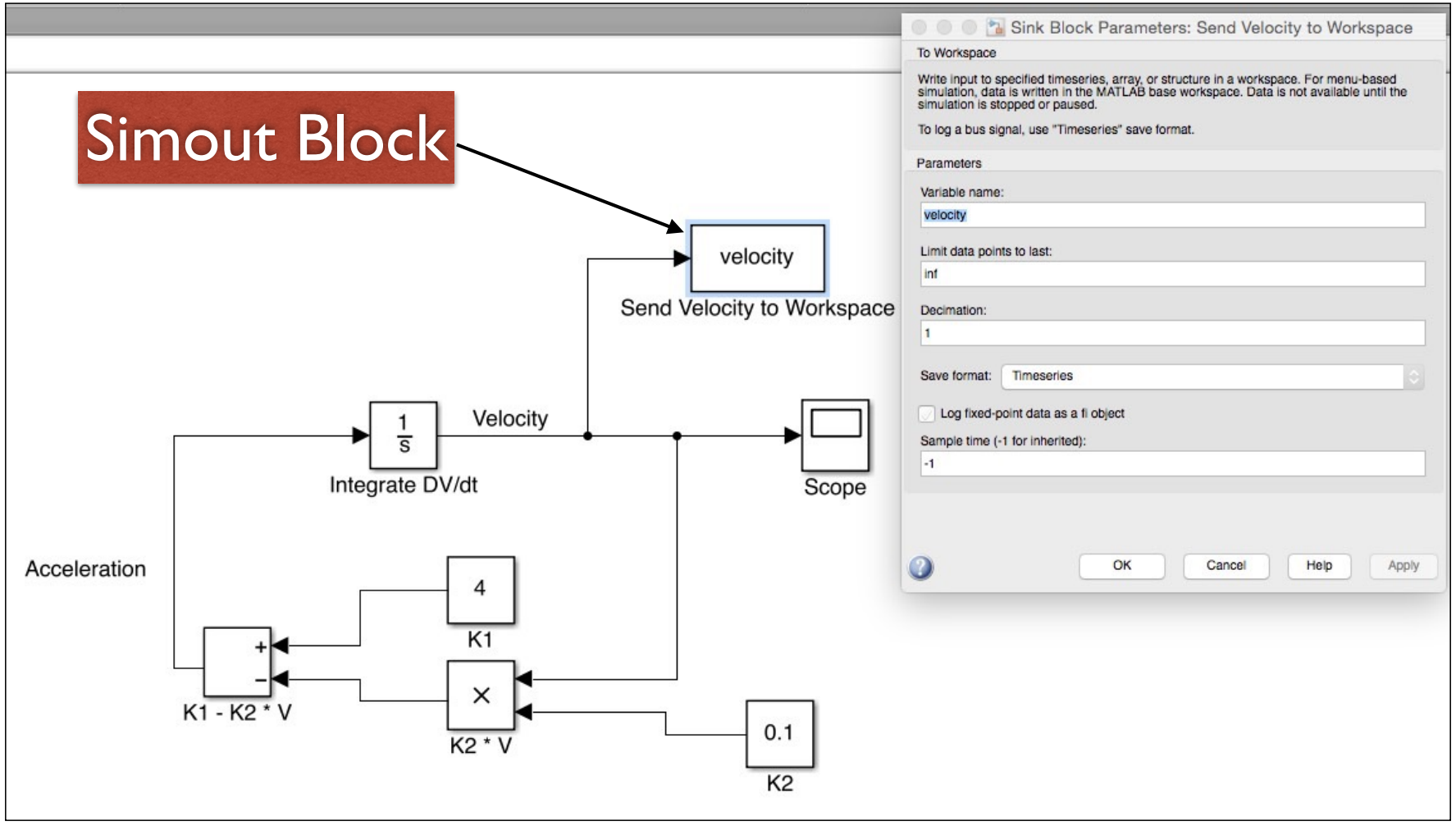
Manipulation of State Variables

- Make a plot of time vs. Velocity



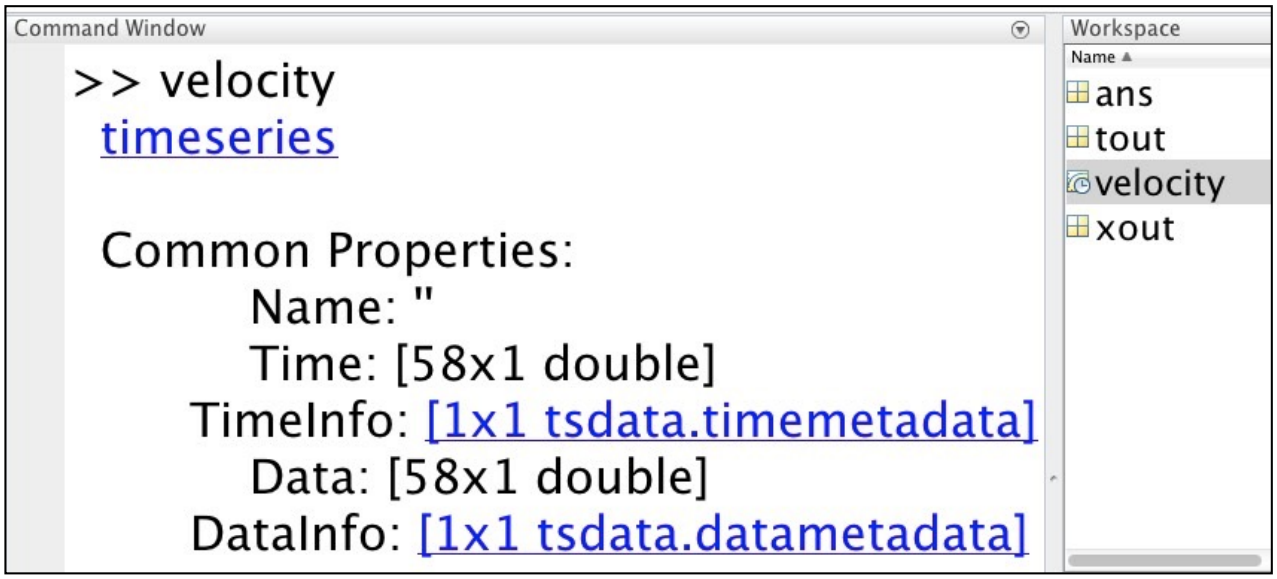
Using the Simout Block

- Add a Simout block (located in the "Sinks" library) to export the velocity signal



Using the Simout Block

- Outcome of using the Simout block
- A structure array (**velocity**) is generated in the workspace



velocity

1x1 double timeseries

Time series name:

Time	Data:1
0	0
5.0238e-05	2.0095e-04
3.0143e-04	0.0012
0.0016	0.0062
0.0078	0.0313
0.0392	0.1566
0.1962	0.7773
0.9812	3.7384
2.1812	7.8388
3.3812	11.4756
4.5812	14.7011

Manipulating the Variables

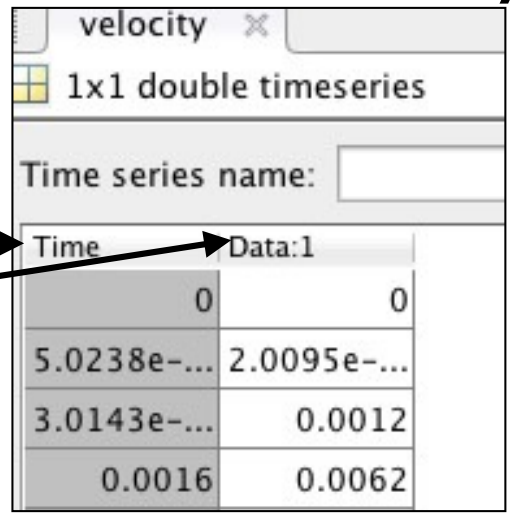
- The variable **velocity** is a structure array (a structure that saves various types of data inside)
- **Velocity** has two data sets available: a) Time and b) Data
- Use the dot notation to point to each one of these data sets

```
>> velocity.Time
>> velocity.Data
```

points to the time vector

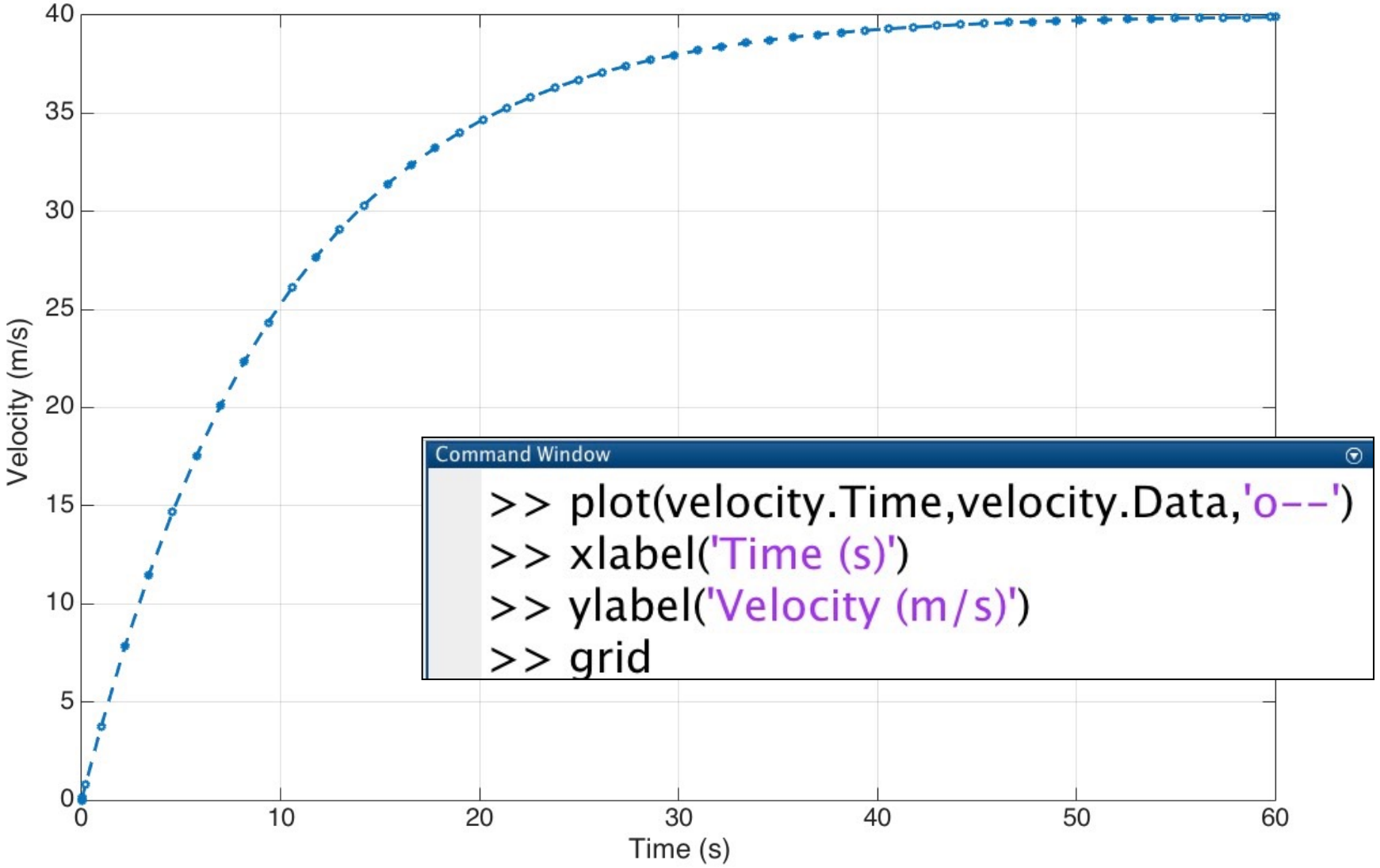
points to the velocity data

```
>> velocity.Time
>> velocity.Data
```



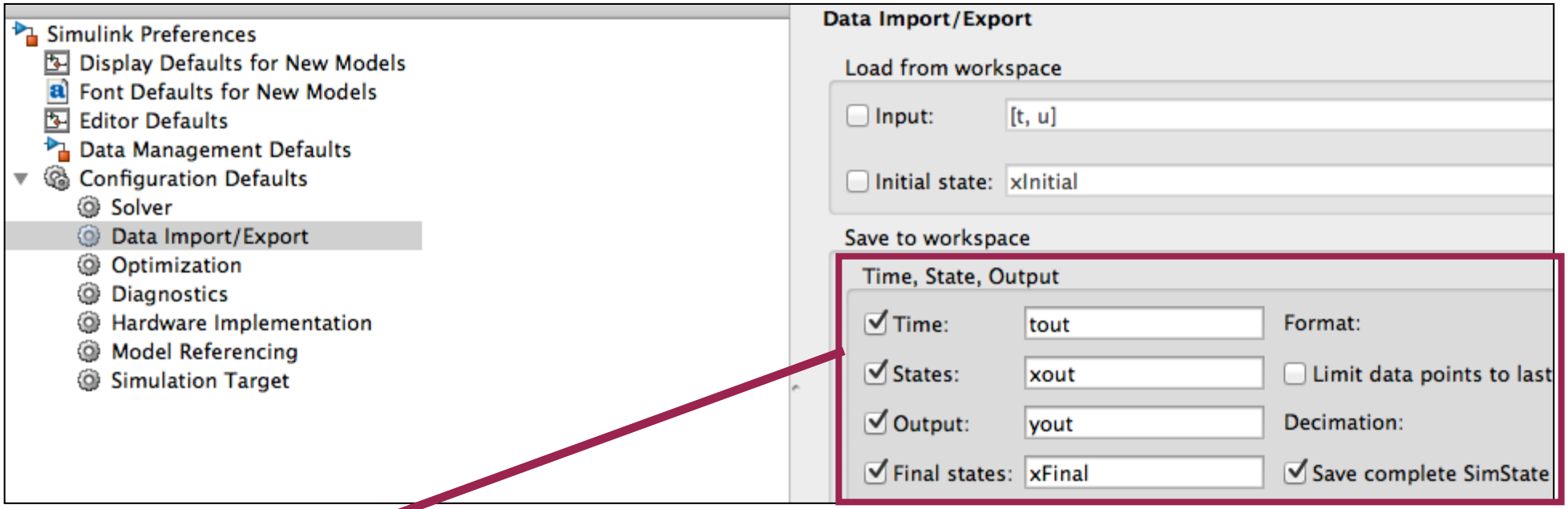
Manipulation of State Variables

- Data inside the structured array can be manipulated like any other variable (see plot below)



Data Export Panel (in Simulink Preferences)
Matlab Release 2012

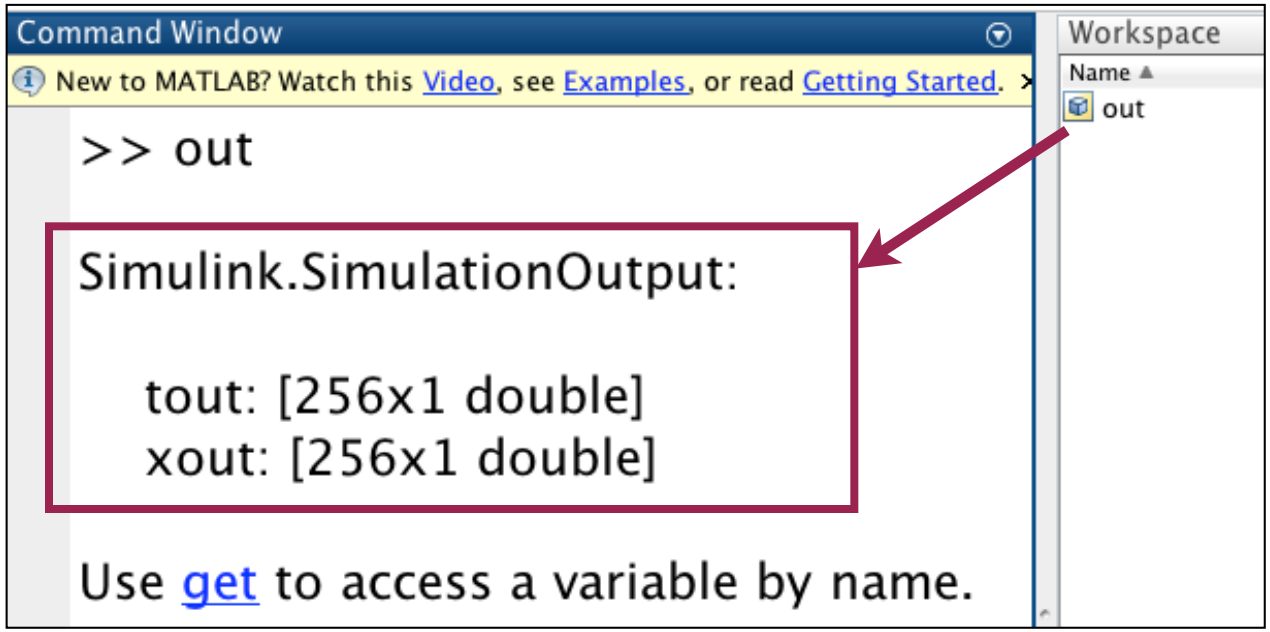
- Select the States and Output in the Data Export panel



Time = **tout** = contains values of time for state variables
States = **xout** = contains an array of state variables calculated by the model
Output = **yout** = a Simulink data structure containing both the time and state variable results

Exported State Variables

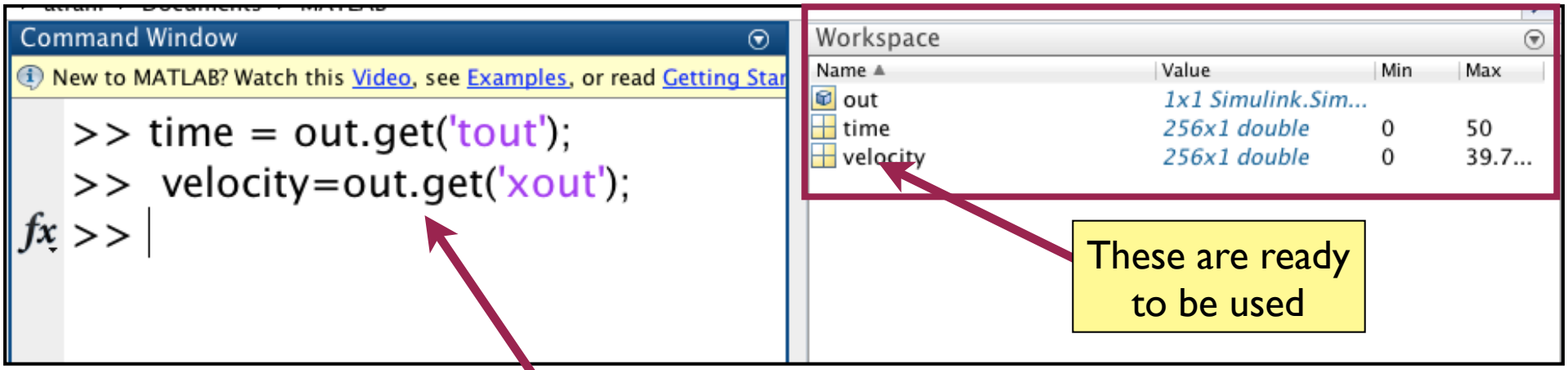
- The workspace variable "out" is a Simulink class called SimulationOutput that contains three variables:



tout = contains values of time for state variables
xout = contains an array of state variables calculated by the model

"Get" the State Variables

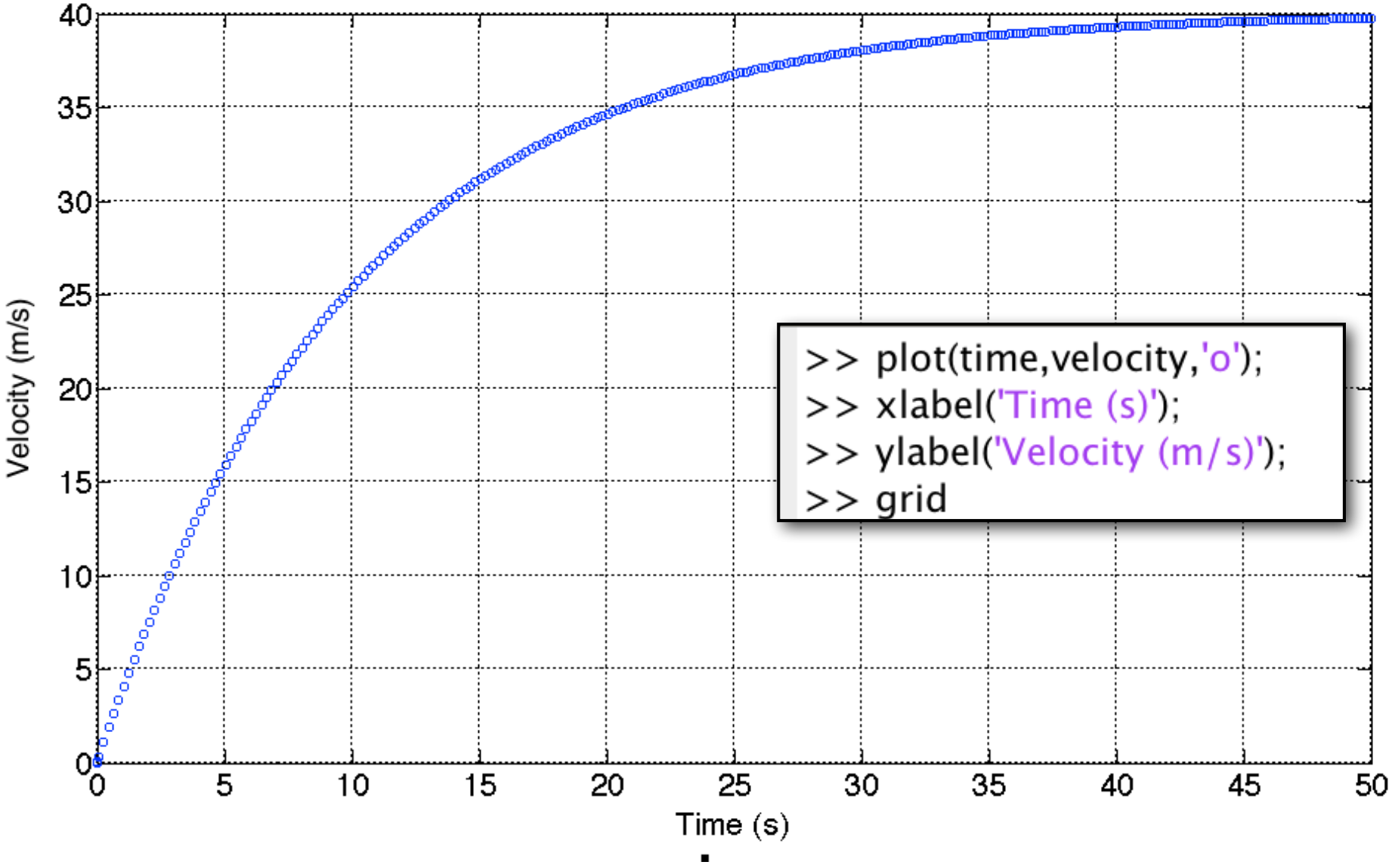
- Use the get command to obtain the variables inside the Simulink class SimulationOutput



- I created two new workspace variables: time and velocity to extract the numerical values of the state variables produced by Simulink
- Once the variables are create, proceed to make a plot

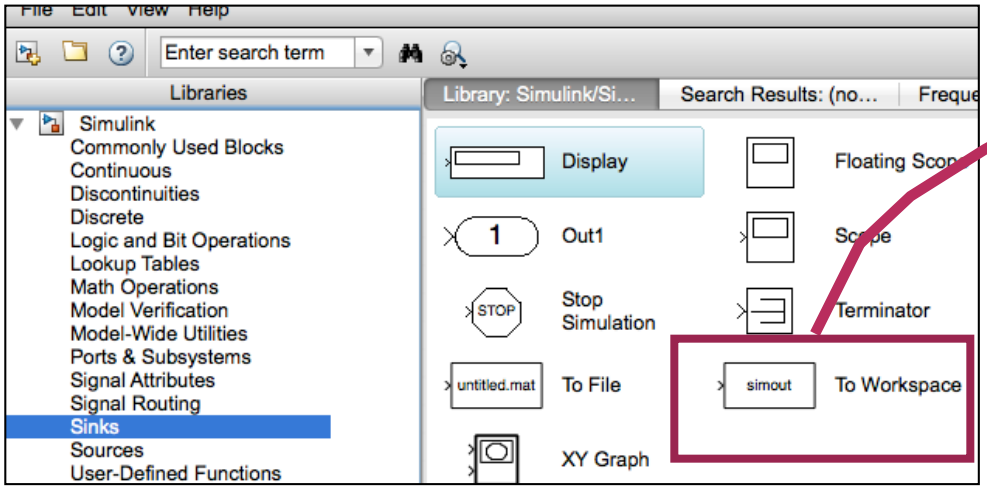
Manipulation of State Variables

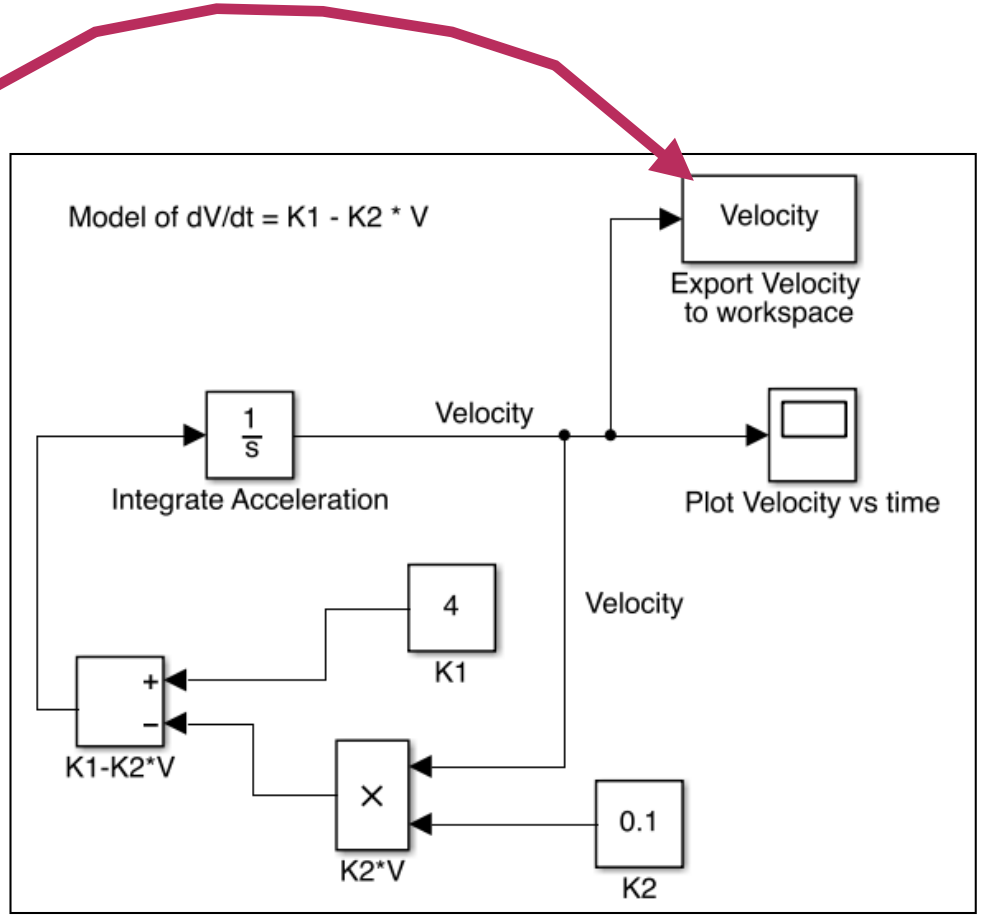
- Make a plot of time vs. Velocity



Using the Simout Block

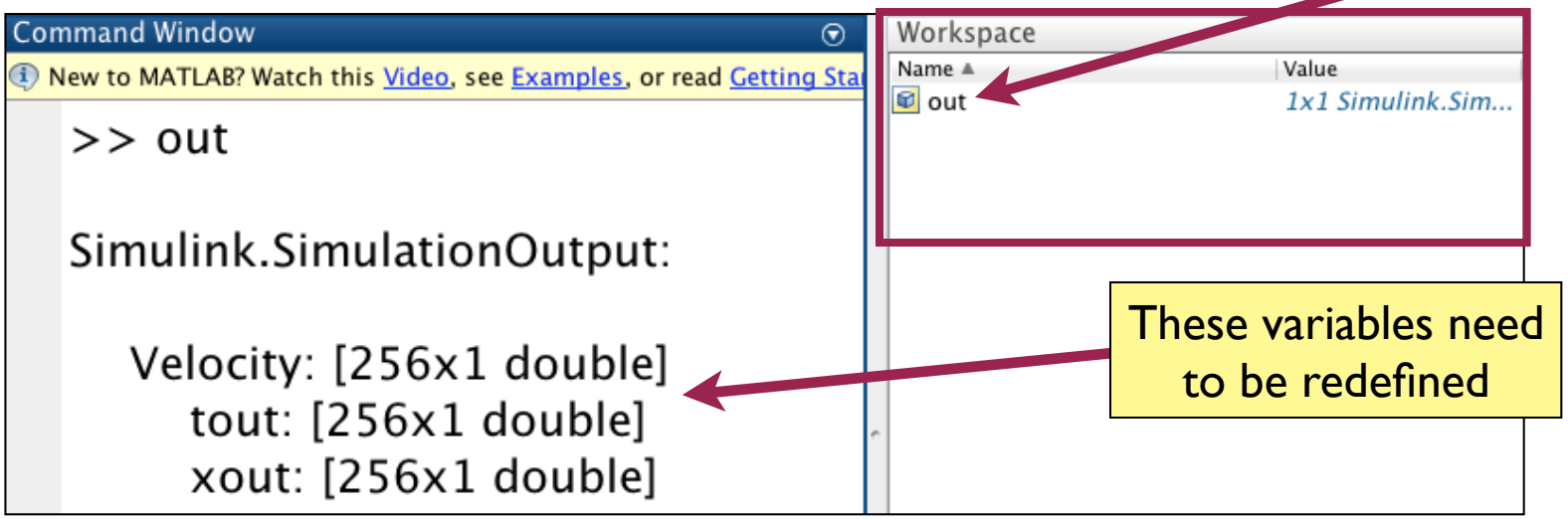
- Add a Simout block (located in the "Sinks" library) to export the velocity signal





"Get" the State Variables

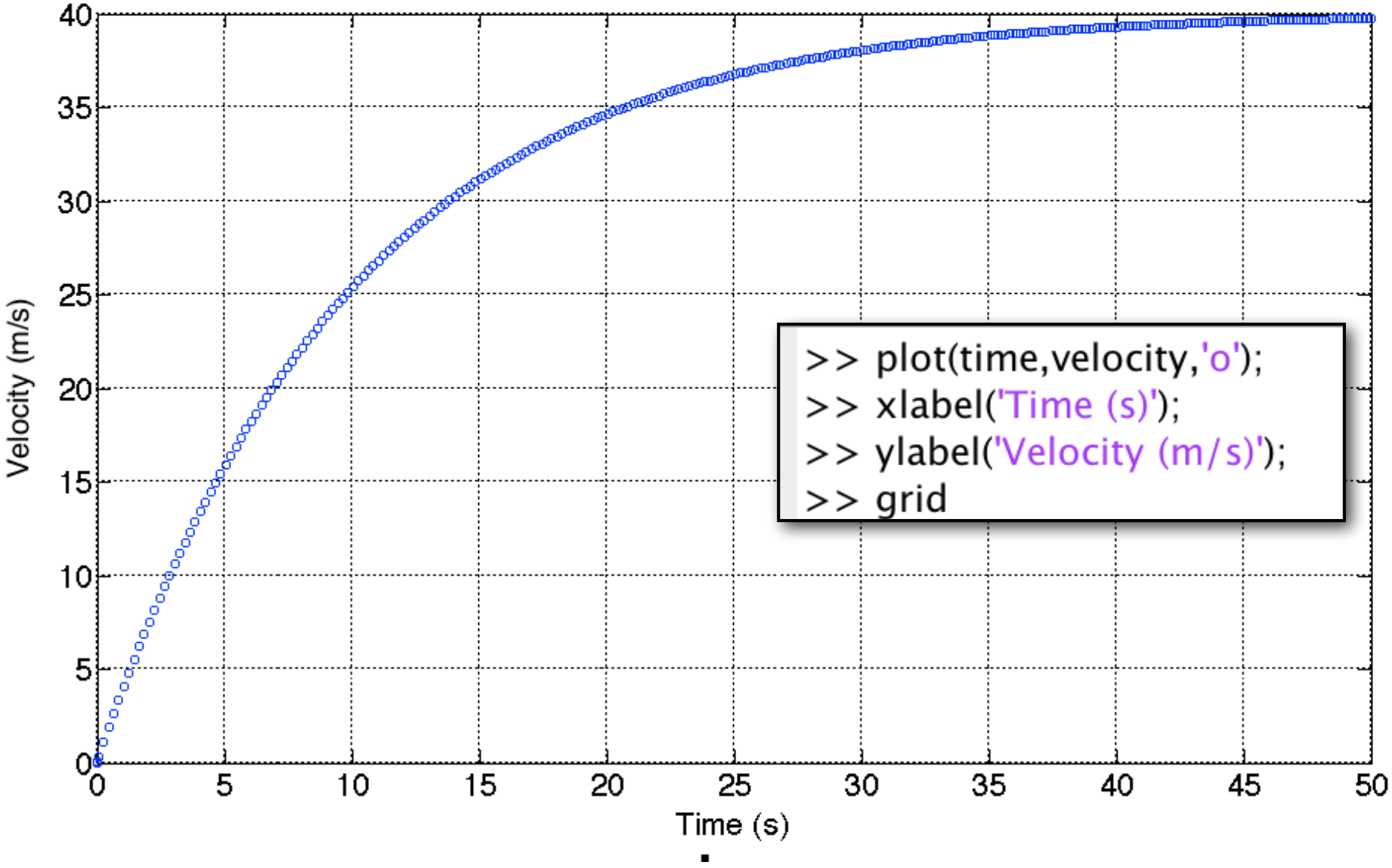
- Note that all variables produced by the model are contained in the Simulink class SimulationOutput ("**out**")



- Use the "Get" class to obtain the numerical values of Velocity, xout and tout

```
>> time=out.get('tout');
>> velocity=out.get('Velocity');
>> plot(time,Velocity,'o--'
```

Manipulation of State Variables



Car Following Model (Another Example)

- This problem has been studied for the past 45 years in traffic flow theory
- Started with the work of Gazis and Herman
- Models vehicles individually
- The vehicle and human interactions are modeled explicitly
- A driver follows another vehicle by judging:

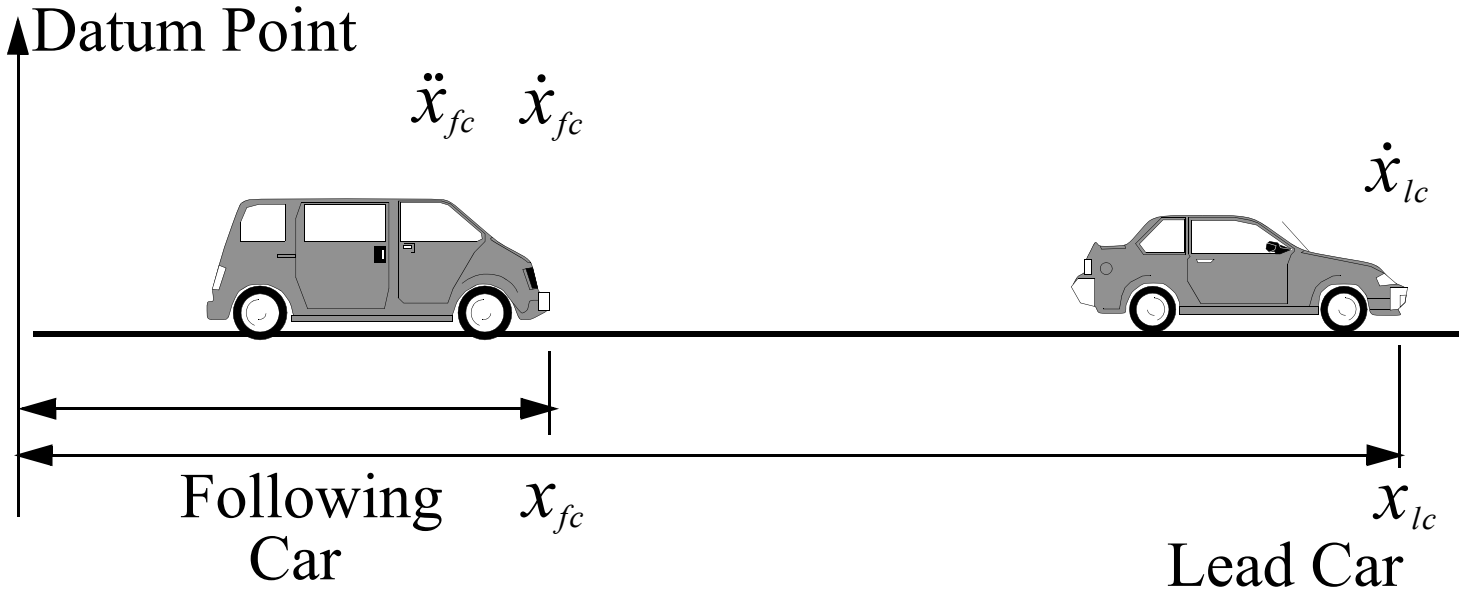
  a) Distance

  b) Speed difference

  c) Reaction time

  d) Vehicle performance

Car Following Model



- Two cars follow each other on the same road
- The driver in the lead car has a speed known speed profile independent of the following car
- The driver in the following car adjusts to the behavior of the front car

Car Following Models

Nomenclature:

x_{fc} and x_{lc} are the positions of the following and lead cars, respectively

$\dot{x}_{fc}$ and $\dot{x}_{lc}$ are the speeds of the following and lead cars, respectively

$\ddot{x}_{fc}$ is the acceleration of the following car. This is the control variable that the driver adjusts to keep up with the lead car (and avoid a collision)

Car Following Models

Assume the velocity profile of the lead car is known as a function of time.

Proposed Driving Rule # 1

The acceleration profile of the "following" car is just a function of the relative speeds of the two cars;

$$\ddot{x}_{fc}(t+\tau) = k\left(\dot{x}(t)_{lc} - \dot{x}(t)_{fc}\right) \qquad (2)$$

where: k is a gain constant of the response process

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

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

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

Car Following Models

Assume the velocity profile of the lead car is known as a function of time.

Proposed Driving Rule # 2

The acceleration profile of the "following" car is a function of the relative speeds and the relative distance between the two vehicles;

$$\ddot{x}_{fc}(t+\tau) = \frac{k\left(\dot{x}_{lc}(t)-\dot{x}_{fc}(t)\right)}{\left(x_{lc}(t)-x_{fc}(t)\right)} \qquad (3)$$

where: k is a gain constant of the response process

$\ddot{x}_{fc}(t + \tau)$ is the acceleration of the following vehicle adjusted for time lag τ

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

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

$x_{lc}(t)$ is the position of the leading vehicle

$x_{fc}(t)$ is the position of the following vehicle

How Do We Use This Model?

- The models presented in equations (1) and (2) can be tested against critical maneuvers performed by the lead vehicle
- The critical maneuvers tests conditions where a driver in the lead vehicle “brakes hard”, accelerates quickly, or follows an erratic velocity profile
- The second order differential equation can be solved numerically if desired of with the use of the Matlab toolbox called Simulink
- Simulink is a toolbox designed to solve differential equations of motion

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 ($t = 0$)
- 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

Example 2 - Car Following Problem

Solve the second-order differential equation of motion representing the acceleration profile of the "following" vehicle (acceleration is function of the relative speeds of the two cars)

$$\ddot{x}_{fc}(t + \mathtt{t}) = k(\dot{x}(t)_{lc} - \dot{x}(t)_{fc}) \quad (4)$$

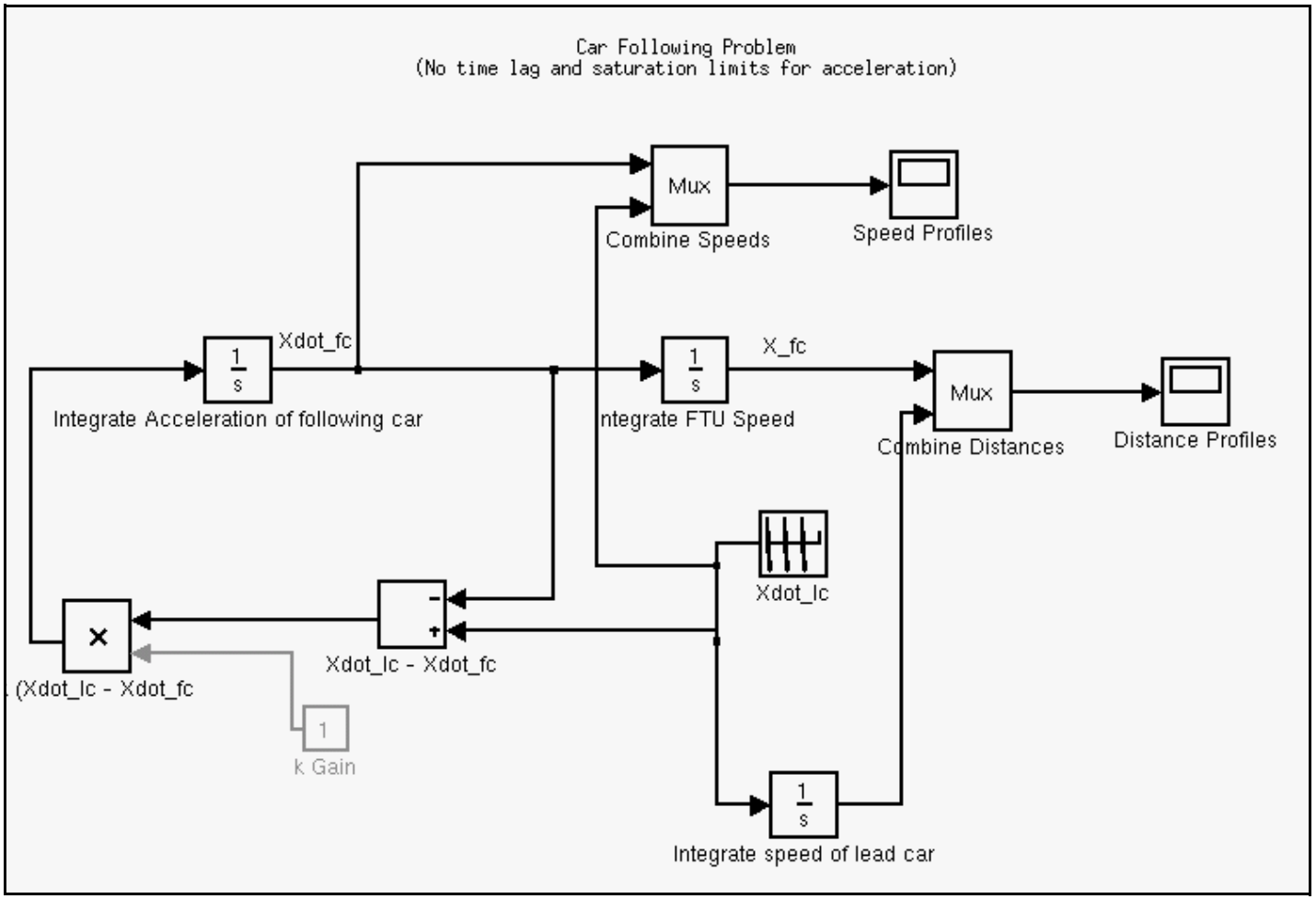
where: k is a gain constant of the response process

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

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

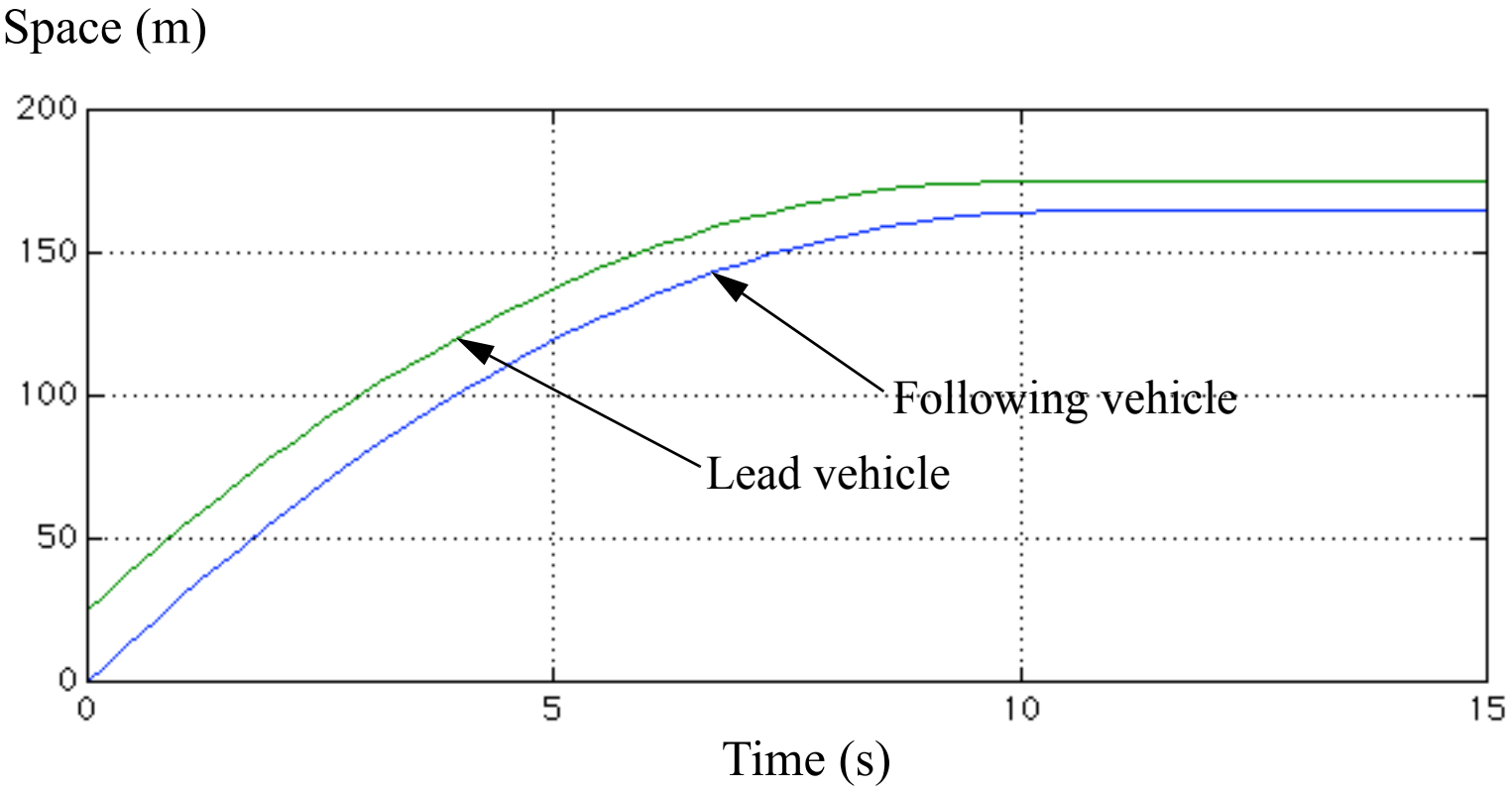
$\dot{x}_{fc}(t)$ is the speed of the following vehicle

Simulink Representation of the Car-Following Problem



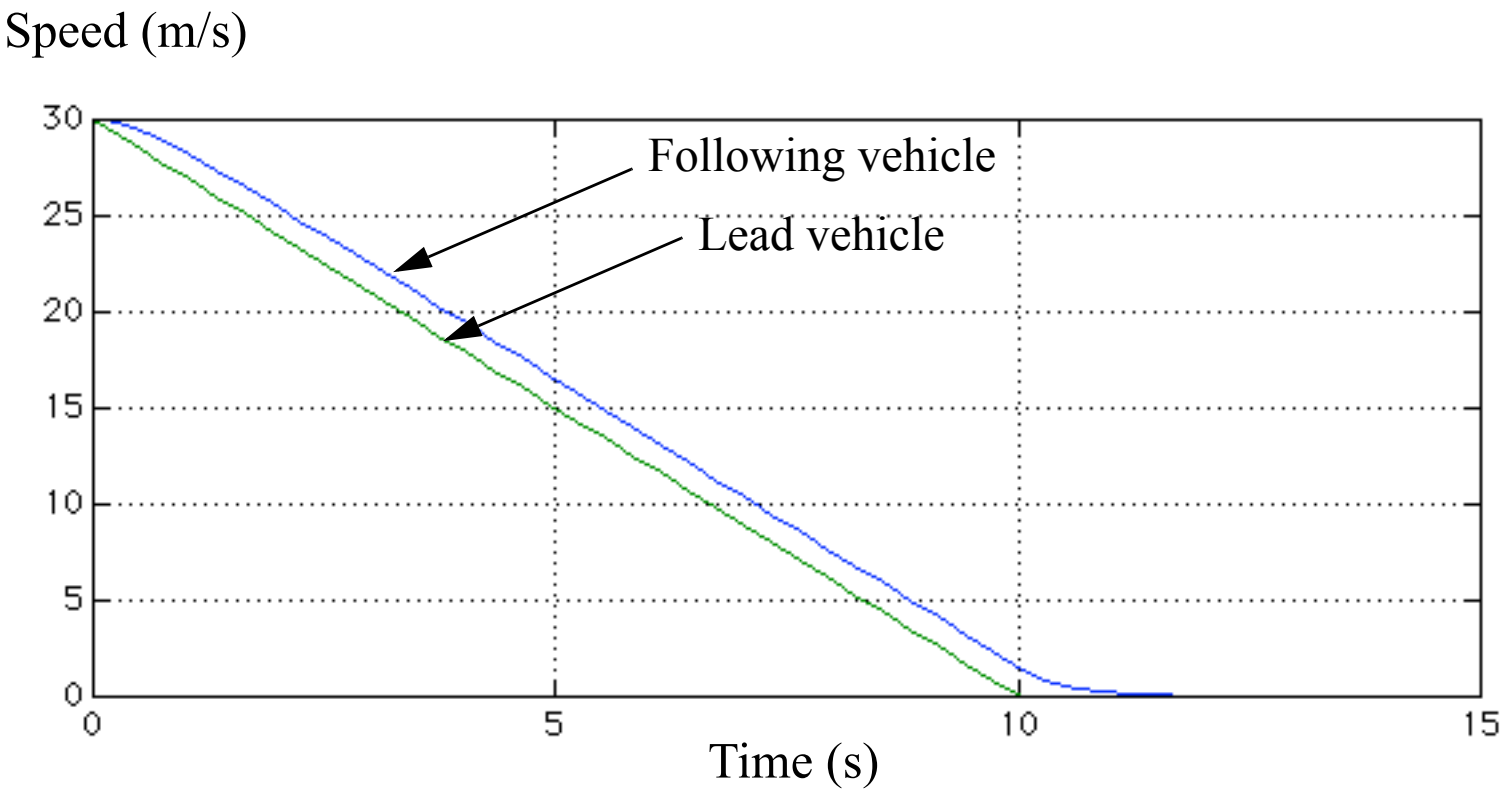
Car-Following Model Output

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



Car-Following Model Output

Velocity profiles for lead and following vehicles



Car-Following Experiment

Goal: to determine the practical capacity of an interstate highway using a car-following model

Method:

a) Simulate a critical maneuver (such as hard braking)

b) Determine if vehicles collide given known initial positions and speeds (initial conditions)

c) Repeat the experiment in part (b) for various initial conditions

Car-Following Experiment

d) Assuming drivers use common sense, the minimum spacing is the distance between cars that avoids the collision by some desired margin

NOTE: In real practice, several critical manuevers could be tried to obtain a safe spacing between successive cars. The issue is whether people driving are so predictable!

Car Following Experiment

Assumptions:

a) Cars travel at 30 m/s initially

b) Lead car initial position is at datum point (0 meters)

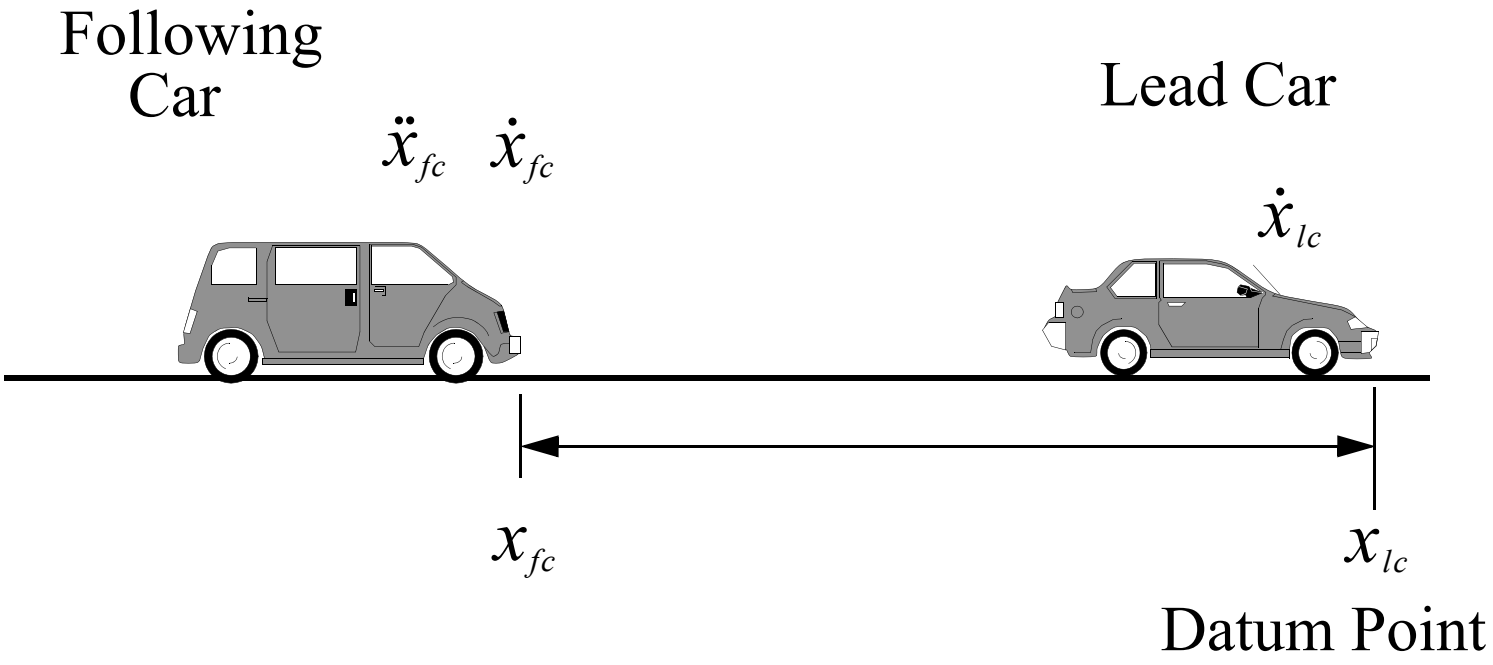
c) Lead car brakes at 3 m/s^2 to avoid an obstacle

d) Following car brakes behind lead car

e) Car size is 5.8 meters in length

f) Minimum desired distance between cars at end of critical maneuver is one car length (5.8 meters)

Car Following Example



Minimum safe distance between two stopped vehicles is (11.6 meters - two car lengths)

Car Following Experiment

Table containing outcomes of car-following experiment

Reaction time = 1.5 seconds

Following Car Initial Position (m) from Datum Point	Distance Between Front Bumpers at Closest Point of Approach (m)	Remark
-100	70	Safe
-90	60	Safe
-80	50	Safe
-70	40	Safe
-60	30	Safe
-50	20	Safe
-40	10	No
-30	0	No

Interpretation of Results

The experiment suggest that the critical spacing to avoid an accident is above 40 meters and less than 50 meters

Further refinement of the experiment suggests 42 meters is the critical spacing to avoid an accident

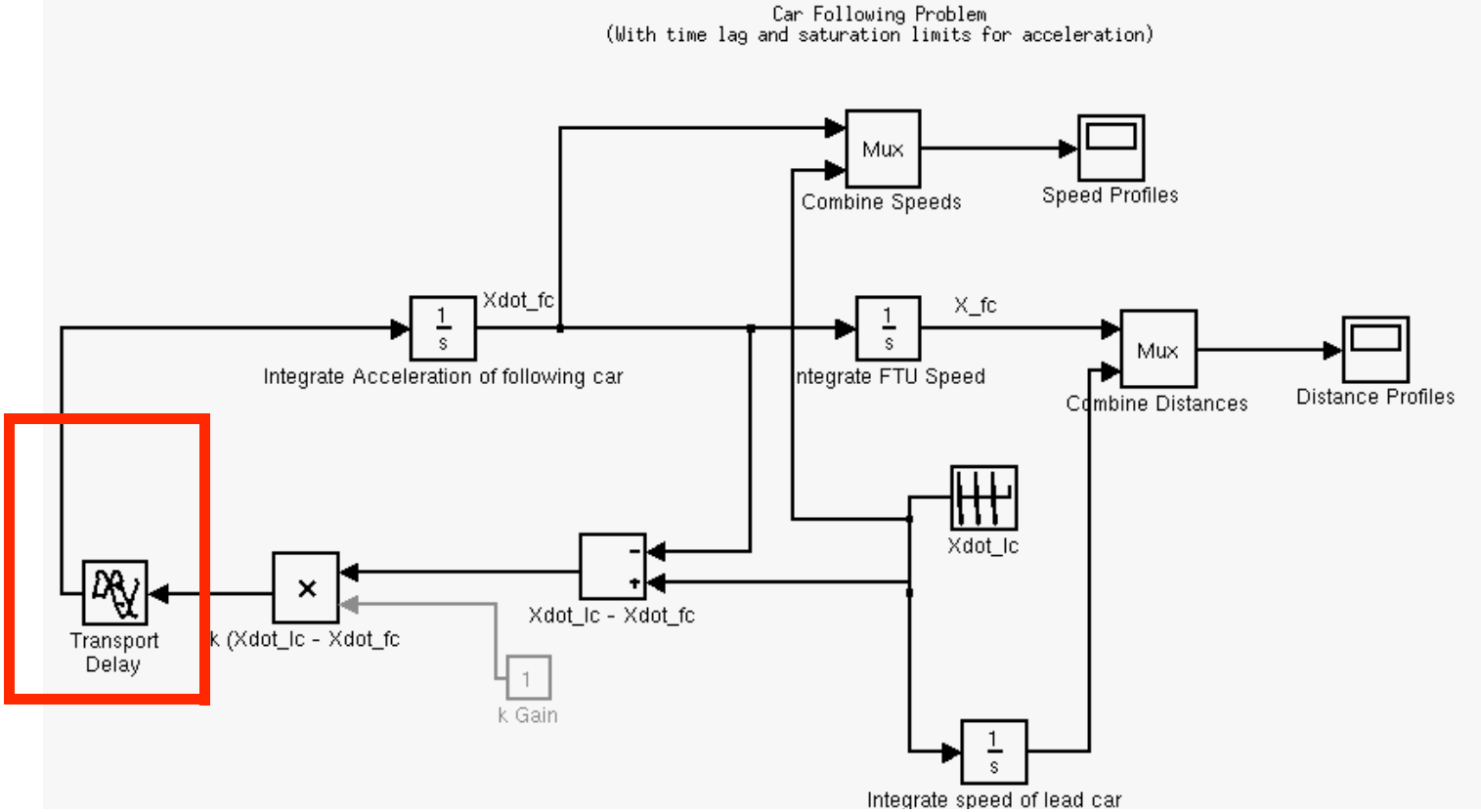
Starting with an initial condition of 30 m/s for the lead and following cars, this implies a headway of 1.4 seconds per successive vehicle

The approximate capacity of the highway for this critical maneuver would be 2,514 vehicles per hour per lane

NOTE: This value is optimistic since no time lags have been factored in the man-machine system

Car-Following Model with Delay

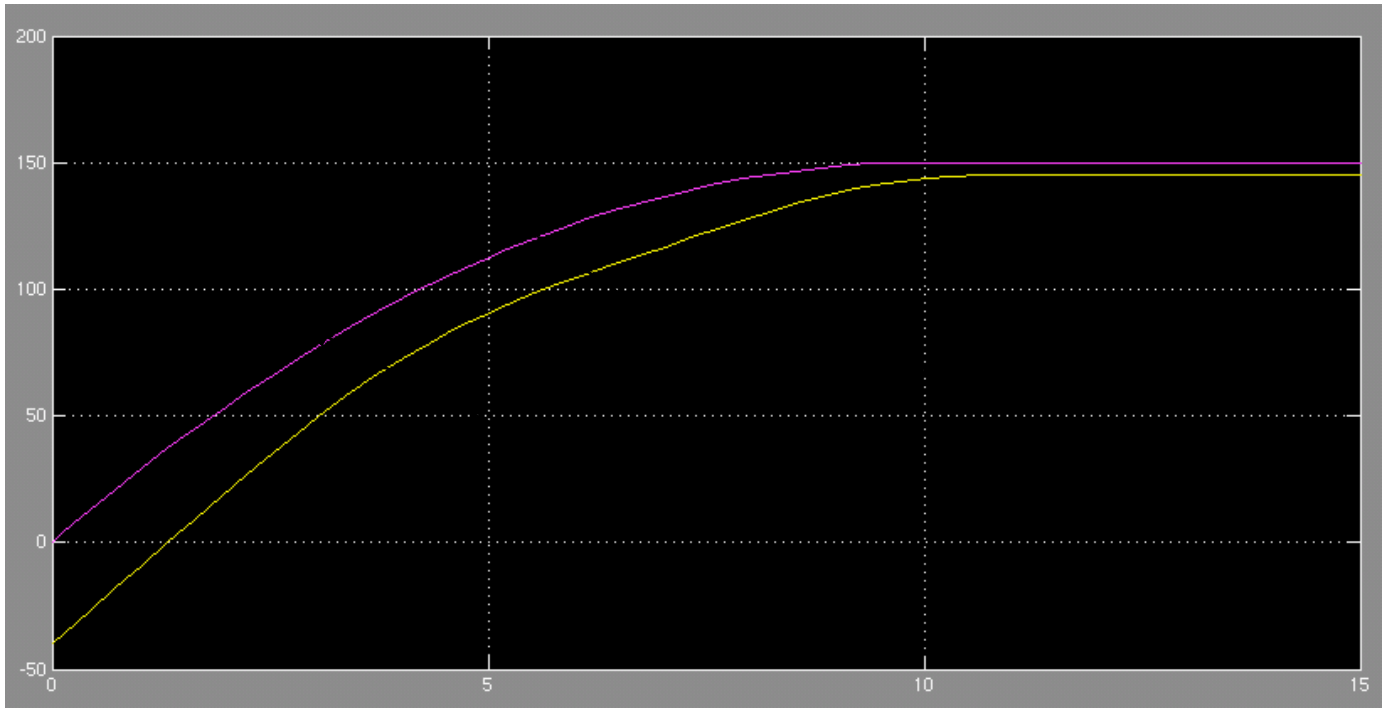
A pure transport delay block is added to the original model simulating the transport lag dynamics of a man-machine system



Output of the Car-Following Model with Delay

- If the following car is 50 meters behind the lead car, clearly a collision (spacing < 11.6 m.) occurs if t =1.5s.

Space (m)



Time (s)

Car Following Experiment (Again)

Repeat (for homework the experiment) including the time lag (to be conservative use a 2.5 second reaction time)

Other Assumptions:

a) Cars travel at 30 m/s initially

b) Lead car initial position is at datum point (0 meters)

c) Lead car brakes at 3 m/s^2 to avoid an obstacle

d) Following car brakes behind lead car

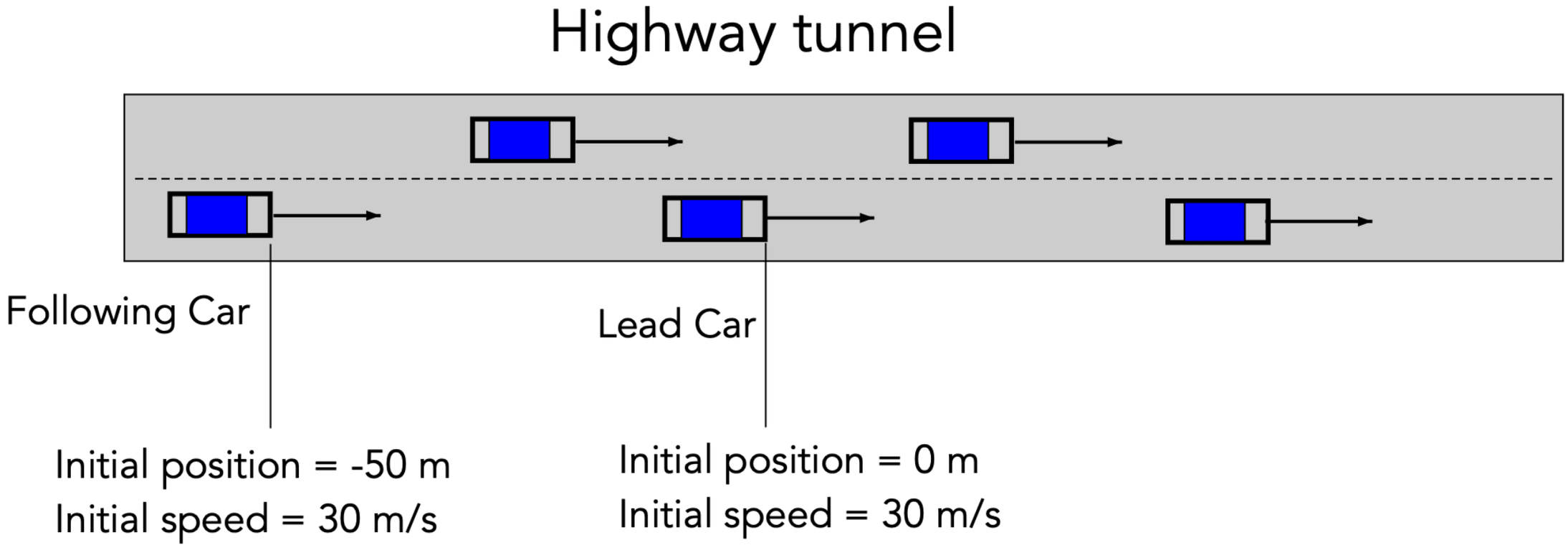
Car Following Experiment (Again)

e) Car size is 5.8 meters in length

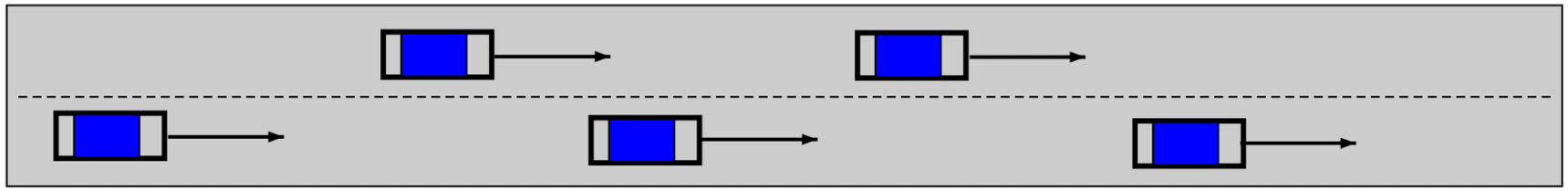
f) Minimum desired distance between cars at end of critical maneuver is one car length (5.8 meters)

- Early experiments to determine road capacity focused on highway tunnels in New York
- Work by Greenshields (1936) developed a simple macroscopic model to predict road/highway capacity

Source: Port Authority of New York



Highway tunnel



Car Following Problem
(With time lag and saturation limits for acceleration)

Following Car Blocks

AccelerationProfiles1

Limit Acceleration

Saturation Limits

Integrate Acceleration of following car

Xdot_fc

Speed of FC

Integrate Following Car Speed

X_fc

Distance of FC

Mux

Combine Speeds

Speed Profiles

Mux

Combine Distances

Distance Profiles

Highway design practice uses 2.5 seconds as reaction time

Time Lag

k (Xdot_lc - Xdot_fc)

k Gain

Xdot_lc - Xdot_fc

Xdot_lc

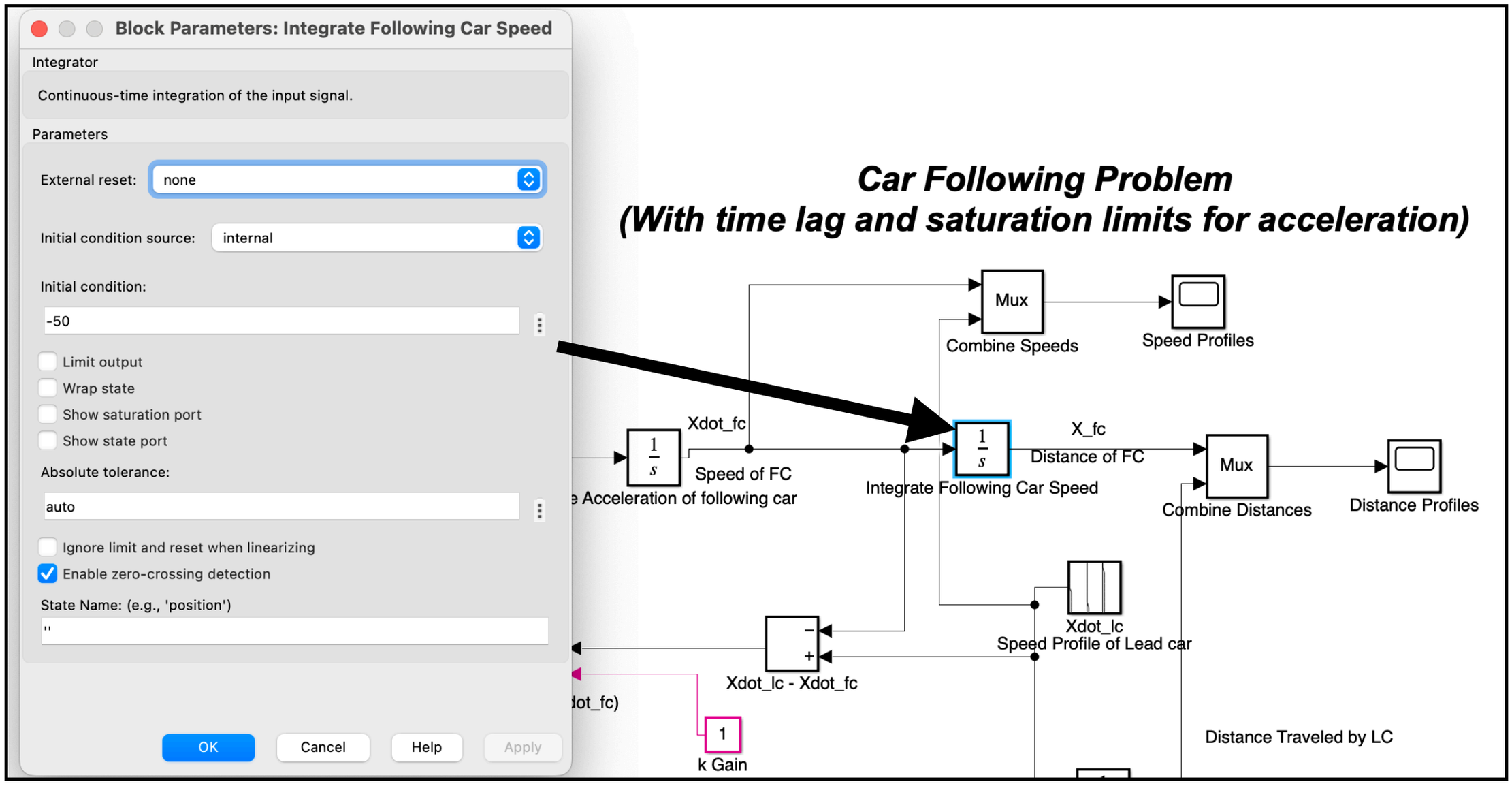
Speed Profile of Lead car

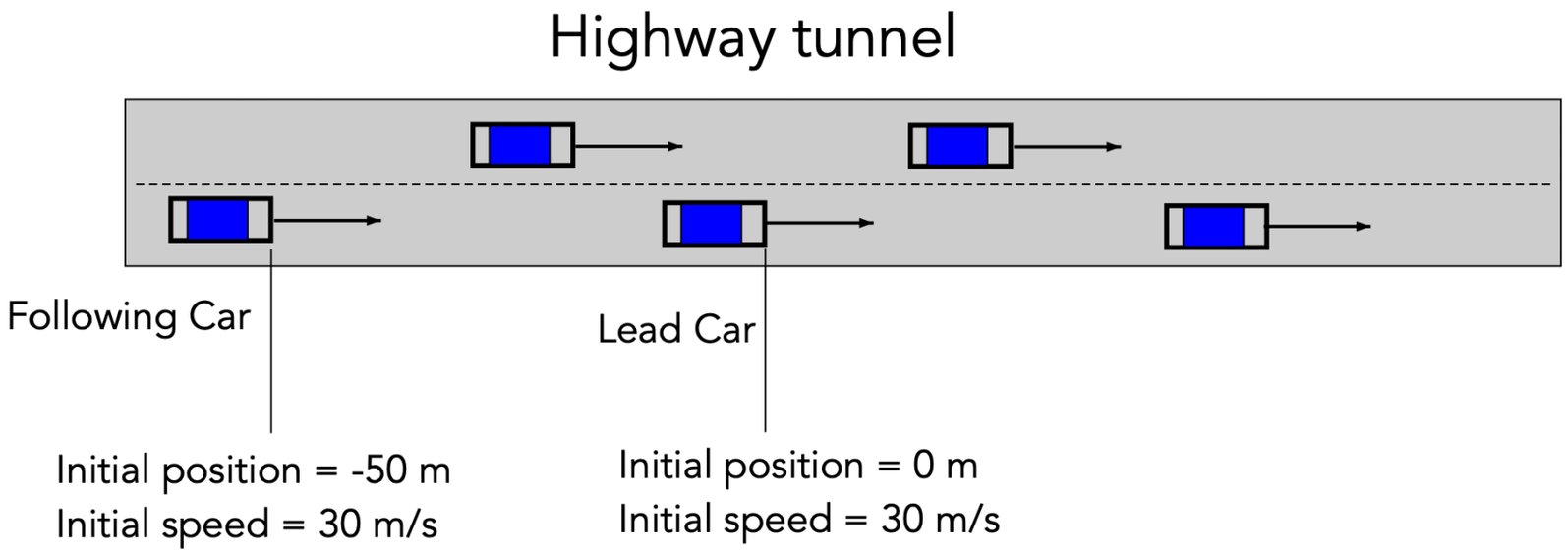
Lead Car Blocks

Distance Traveled by LC

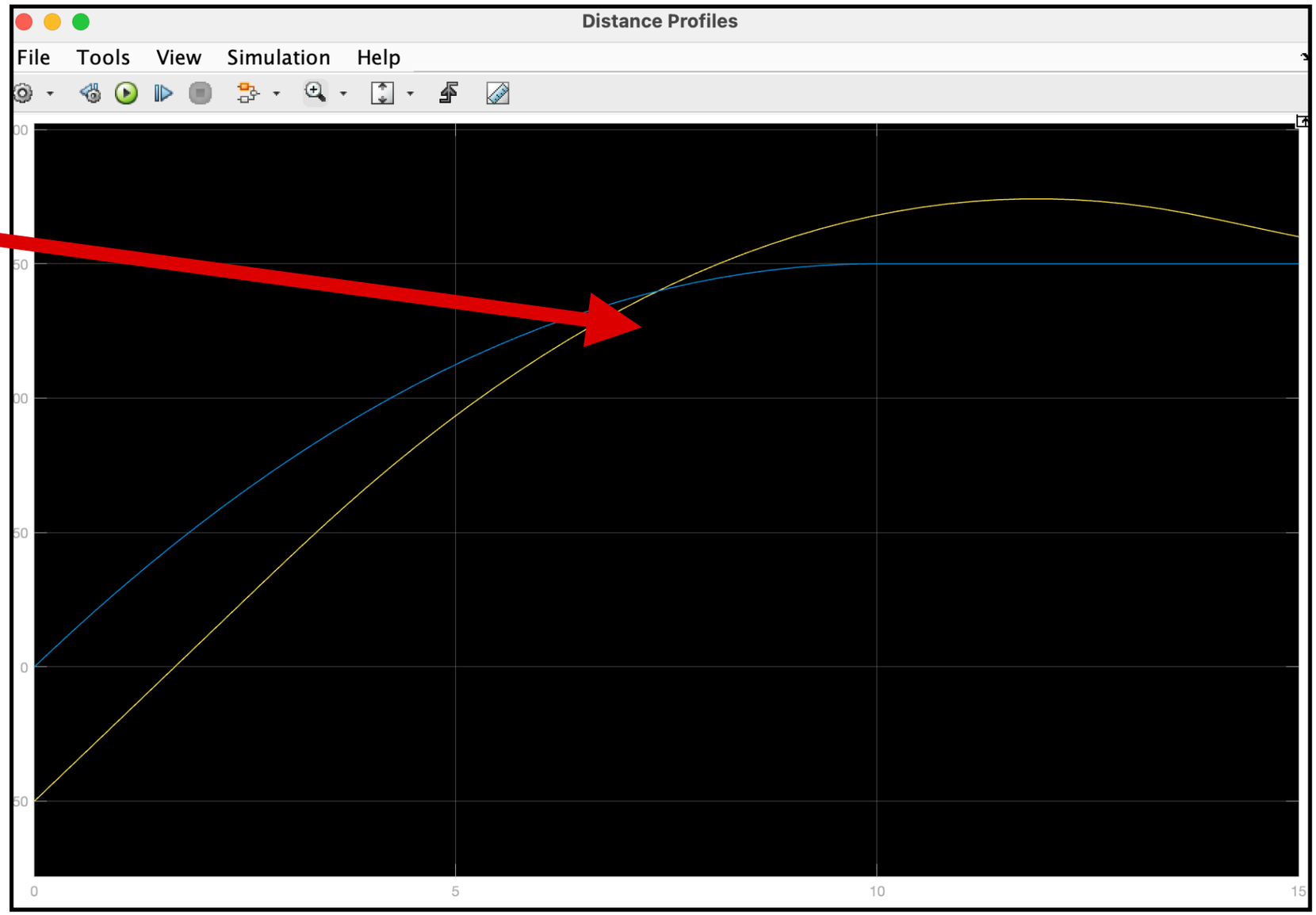
Integrate speed of lead car

The initial distance of the following car is specified in the integrator block for the following car

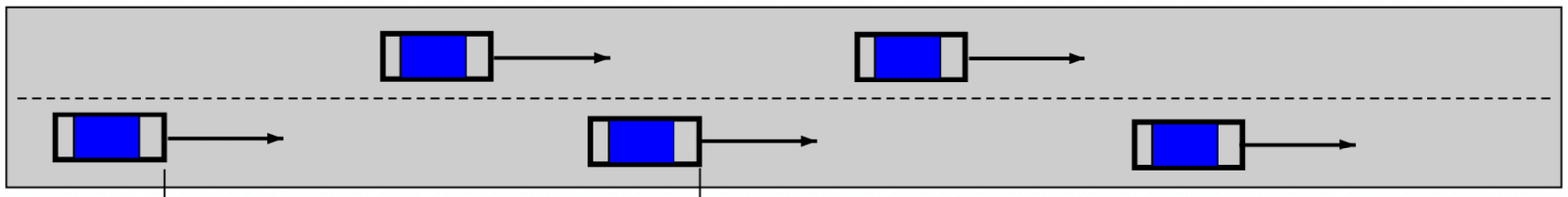




- Collision occurs at 7 seconds after the lead car starts braking
- Separating cars 50 meters with a distracted driver is not safe



Highway tunnel



Following Car

Lead Car

Initial position = -100 m
Initial speed = 30 m/s

Initial position = 0 m
Initial speed = 30 m/s

- No collision is observed
- Separating cars 100 meters with a distracted driver is safe