

EENG498BC: Systems Exploration, Engineering, and Design Laboratory

Spring 2016

Exercise 2: System Simulation

Vibhuti Dave and Tyrone Vincent*

1. Introduction

The purpose of this exercise is to create a simulation of a vehicle using MATLAB and Simulink.

2. Simulink Introduction

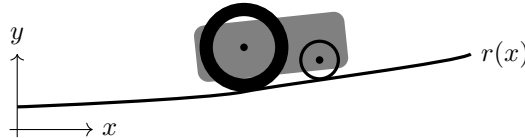
You will need to learn how to enter and simulate differential equations in Simulink. You should do this *before* coming to lab. The following resources may be useful:

- Simulink video tutorials: http://www.mathworks.com/academia/student_center/Tutorials/simulink-launchpad.html (Suggest viewing Simulink On-Ramp and Using Simulink to Model Continuous Dynamical Systems)
- Michigan/CMU Simulink Tutorial: <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=SimulinkModeling>

Note that you can get access to Matlab and Simulink at most any computer lab on campus, and in addition, you can utilize the virtual lab at matlab.mines.edu. It is expected that you are already familiar with MATLAB and creating MATLAB functions. If not, you may also want to look at the Mathworks tutorials for MATLAB.

3. The System

The vehicle to be simulated is shown in cartoon form below. It has wheels which are driven by a motor. The position of the vehicle is measured relative to an (arbitrary) fixed point, with horizontal location x and vertical location y . (We assume the vehicle moves in a straight line, so only two axes are necessary). The vehicle drives along a road with road surface height given by the function $r(x)$.



The differential equation that describes the movement of the vehicle in the x direction is given by

$$\ddot{x} = \cos(\phi) \left[-\frac{d^2 r}{dx^2} (\dot{x})^2 \sin(\phi) - g \sin(\phi) + \frac{1}{m} f - \frac{b}{m \cos(\phi)} \dot{x} \right].$$

The parameters in this equation are the following:

Parameter	Description
f	force on the vehicle due to motor/tires (N)
g	gravitational acceleration 9.81 m/s^2
m	mass of the vehicle (kg)
b	translational drag constant (Ns/m)
$\phi = \tan^{-1}(\frac{dr}{dx})$	the angle that the road makes with the horizontal (rad)

* Developed and edited by Tyrone Vincent and Vibhuti Dave

The assumptions behind and derivation of this equation can be found in the Appendix.

You will use Simulink to solve this differential equation. Once you find $x(t)$, $y(t)$ is given by $y(t) = r(x(t))$.

4. Creating your road

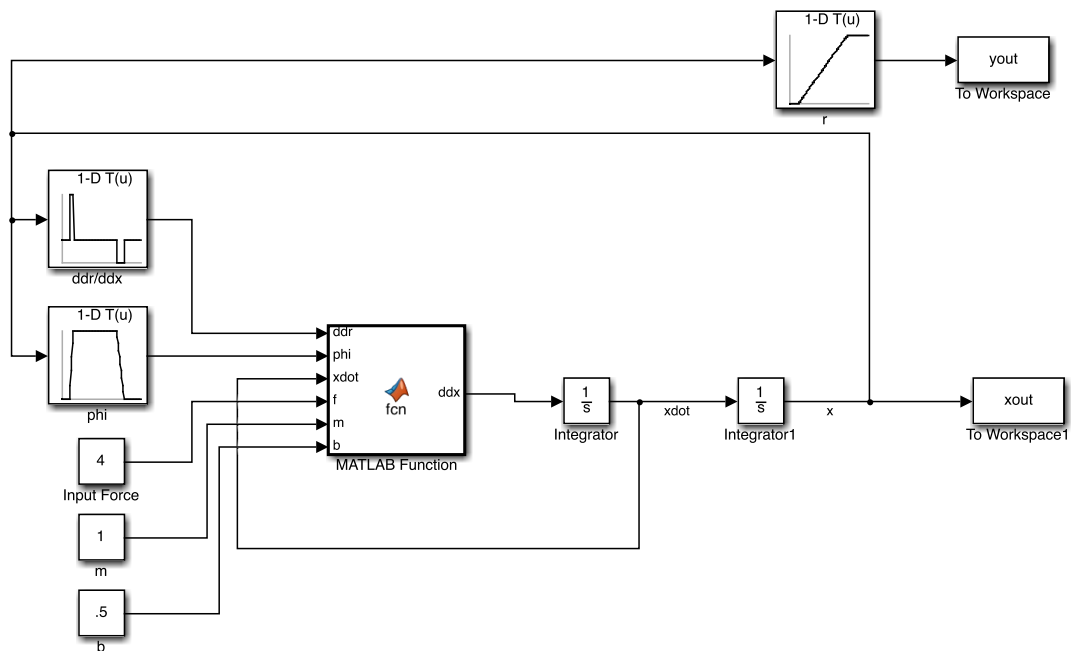
To implement the simulation, you will need a function to return various characteristics of the road at location x , specifically $r(x)$, $\phi = \tan^{-1} \left(\frac{dr}{dx} \right)$ and $\frac{d^2r}{dx^2}$. If you define $r(x)$ first, you run the risk of creating a function that is not differentiable twice. Thus, it is easier to define $\frac{d^2r}{dx^2}$ first, and then integrate twice to get the other functions. The following MATLAB function is an example of using this approach. Note that the line `cumsum(ddr)*dx` is a cumulative sum that implements a discrete approximation of the integral. Type `help cumsum` for more information.

```
%
% Grid on which to define d^2r/dx^2
%
dx=.1;
x = 0:dx:10; % m
%
% Initialize d^2r/dx^2 as zero
%
ddr = zeros(size(x));
%
% Increasing slope between 1 m and 1.5 m
% To end at a slope of 1/3, magnitude of d^2r/dx^2 = (1/3)/0.5
%
index = find(x>=1 & x<1.5);
ddr(index) = ((1/3)/0.5)*ones(size(index));
%
% Decreasing slope between 7 m and 8 m
% want to end back a slope of 0
%
index = find(x>=7 & x<8);
ddr(index) = -((1/3)/1)*ones(size(index));
%
% Find dr/dx, phi, and r
%
dr = cumsum(ddr)*dx;
phi = atan(dr);
r = cumsum(dr)*dx;
%
% Plot the results
%
figure(1)
clf
set(gca,'fontsize',16) % make sure text is readable
subplot(3,1,1)
plot(x,ddr)
subplot(3,1,2)
plot(x,dr)
ylabel('$\frac{dr}{dx}$','interpreter','latex')
subplot(3,1,3)
plot(x,r)
ylabel('$r$','interpreter','latex')
xlabel('$x$ (m)','interpreter','latex')
```

```
%
% Save the results in a .mat file
%
save road.mat x r dr ddr phi
```

5. Deliverables

- Implement a simulation of the vehicle model in Simulink. Because the equations are fairly complicated, the Matlab function block, found in the User-Defined Functions library, could be useful. (Use the MATLAB help to get more information on how to use this block.) In addition, you will want to use a 1-D lookup table block, found in the Lookup Tables library, to implement $r(x)$, $\phi(x)$, and $\frac{d^2r}{dx^2}(x)$. If the results of the above script are in the local workspace, then setting Breakpoints 1: to x and Table Data: to r , ϕ , or ddr , as appropriate, will implement a linear interpolation of the road data. Implement your simulation so that the parameters (i.e. m , b , etc.) are easy to change. The following example shows a typical configuration



Note that `to workspace` blocks are used to save the results in the variables `yout` and `xout`. The default is for these variables to be structures called Timeseries. After running the simulation, type the name of the variable (i.e. `yout`) at the command line to see the elements of the structure. `yout.Data` will give you access to the data, while `yout.Time` to the corresponding time points. However, the `plot` command will recognize a Timeseries, so you can just type `plot(yout)` to get a plot of y versus time.

- Create an animation that shows the results of the simulations.
 - Click on the Matlab Help (located on the home tab of the command window), then click on Matlab, Graphics, 2-D and 3-D Plots, and Animation. Try each of the “Examples and How To”.
 - This link gives a detailed example of an animation, and the example of animating a bouncing ball is very close to animating the motion of the vehicle. <https://zerocrossraptor.wordpress.com/matlab-undercover/creating-simple-animation-in-matlab/>

- Re-read the EENG307 lecture notes on PID control, posted on the blackboard site. In simulink, implement a PI controller to regulate the speed of the robot, assuming that you can directly control the applied force f . Use the linearized system dynamics

$$\ddot{x} + \frac{b}{m}\dot{x} = \frac{1}{m}f.$$

as the design model. These dynamics assume a flat road, but the I term in your controller should help account for the force due to a non-flat road. Show that the controller regulates the speed even as the road slope varies.

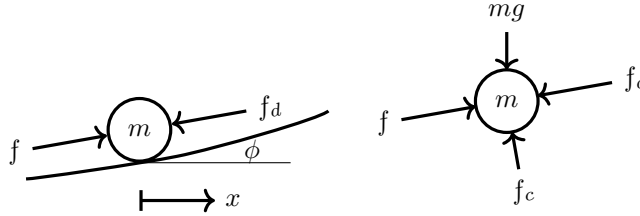
- Create a document that discusses the results of your Simulink simulation, with enough detail that someone else could recreate the results. This document should include the Simulink block diagram, system and simulation parameters, plots of the results, and an interpretation of the results.
- Store your simulink model and animation code for future use.

6. Demonstrations

- Be prepared to demonstrate your simulation and animation.

7. Appendix: Vehicle Model

To model this system, we have to decide what detail we would like to include. For this exercise, let's model the robot as a single mass of m kg, ignoring rotational effects (including the inertia in the rotating tires). When the motor spins the tires, (assuming they don't slip) a force of f N is applied to the mass. Due to wind or other friction forces, a damping force of $f_d = b\dot{v}$ N acts against the vehicle motion, where \dot{v} is the vehicle velocity. The angle of the road at the location of the vehicle is ϕ . This simplified model is shown below, to the left. The road surface supports the vehicle with contact force f_c , and gravity exerts a force of mg N, where g is gravitational acceleration. The free body diagram is shown below, to the right.



We assume that the robot is always in contact with the road, and thus the position along the y axis is always $r(x)$. Newton's law in the x and y axes gives the following:

$$\begin{aligned} \text{acceleration in } y \text{ axis} \quad m\ddot{y} &= f_c \cos(\phi) - mg + (f - f_d) \sin(\phi) \\ \text{acceleration in } x \text{ axis} \quad m\ddot{x} &= (f - f_d) \cos(\phi) - f_c \sin(\phi) \end{aligned}$$

Since the movement along the y axis is prescribed by the road, we have $y = r(x)$ and

$$\begin{aligned} \dot{y} &= \frac{dr}{dx} \dot{x} \\ \ddot{y} &= \frac{d^2r}{dx^2} (\dot{x})^2 + \frac{dr}{dx} \ddot{x}. \end{aligned}$$

We plug this into the first equation, multiply by $\sin(\phi)$ and multiply the second equation by $\cos(\phi)$.

$$\begin{aligned} m \left(\frac{d^2r}{dx^2} (\dot{x})^2 + \frac{dr}{dx} \ddot{x} \right) \sin(\phi) &= f_c \cos(\phi) \sin(\phi) - mg \sin(\phi) + (f - f_d) \sin^2(\phi) \\ m\ddot{x} \cos(\phi) &= (f - f_d) \cos^2(\phi) - f_c \sin(\phi) \cos(\phi) \end{aligned}$$

By adding the two equations, we eliminate f_c

$$m \left(\frac{d^2 r}{dx^2} (\dot{x})^2 \sin(\phi) + \frac{dr}{dx} \ddot{x} \sin(\phi) + \ddot{x} \cos(\phi) \right) = -mg \sin(\phi) + (f - f_d)(\cos^2(\phi) + \sin^2(\phi))$$

Solving for \ddot{x} and simplifying (using $\sin^2(\phi) + \cos^2(\phi) = 1$),

$$\ddot{x} = \frac{1}{\frac{dr}{dx} \sin(\phi) + \cos(\phi)} \left[-\frac{d^2 r}{dx^2} (\dot{x})^2 \sin(\phi) - g \sin(\phi) + \frac{1}{m} (f - f_d) \right]$$

Now, the damping force is given by $f_d = b\dot{v}$. Since the car moves along the road, the velocity v is given by $\dot{v} = \frac{\dot{x}}{\cos(\phi)}$, so that $f_d = \frac{b}{\cos \phi} \dot{x}$. Some further simplification is possible since the slope $\frac{dr}{dx}$ and the angle ϕ are related. In fact, $\tan(\phi) = \frac{dr}{dx}$. Then

$$\begin{aligned} \frac{dr}{dx} \sin(\phi) + \cos(\phi) &= \tan(\phi) \sin(\phi) + \cos(\phi) \\ &= \frac{\sin(\phi)}{\cos(\phi)} \sin(\phi) + \cos(\phi) \\ &= \frac{\sin^2(\phi) + \cos^2(\phi)}{\cos(\phi)} \\ &= \frac{1}{\cos(\phi)} \end{aligned}$$

and our final equation is

$$\ddot{x} = \cos(\phi) \left[-\frac{d^2 r}{dx^2} (\dot{x})^2 \sin(\phi) - g \sin(\phi) + \frac{1}{m} f - \frac{b}{m \cos(\phi)} \dot{x} \right]$$