Assignment 8: Simulink and Modeling Systems

Sample Solutions Instructor: Trani

Problem 1

Model the water distribution system shown in Figure 1 using Simulink. The parameters of the reservoirs and pumps are shown in Table 1. The demand functions for both cities are shown in Table 2.

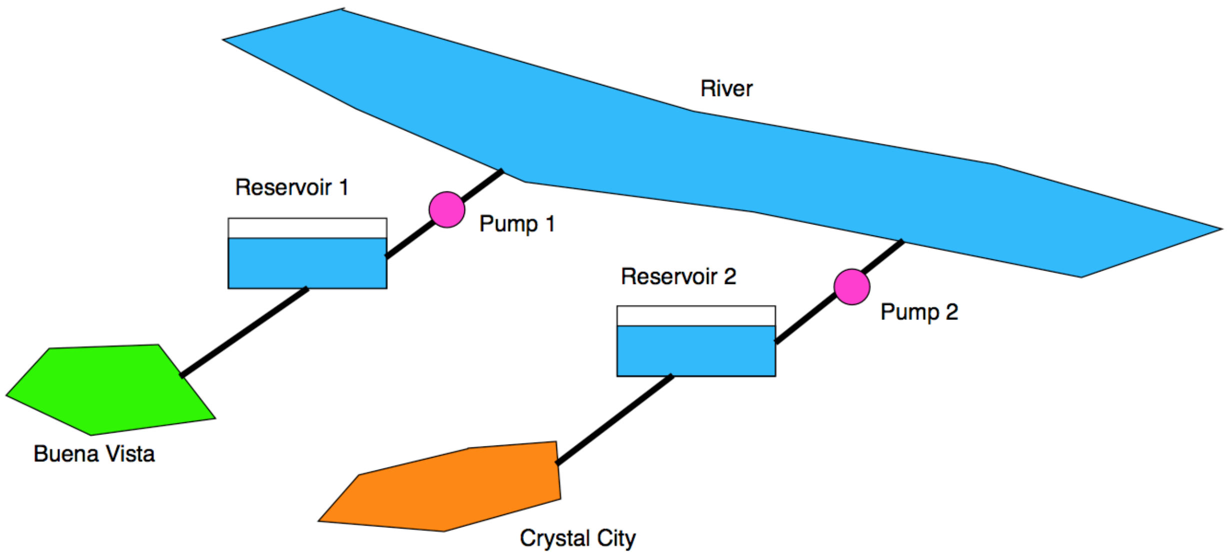

Figure 1. Water Distribution System.

Table 1. Parameters for Water Distribution System.

Parameter	Values	Remarks
Reservoir 1	Area = 3.1e5 (sq. meters) Desired head = 10 meters	10 meters is the desired head to maintain water pressure
Reservoir 2	Area = 3.6e5 (sq. meters) Desired head = 10 meters	10 meters is the desired head to maintain water pressure
Pump 1 constant	2100 (sq. meters/min)	
Pump 2 constant	2600 (sq. meters/min)	
Max. Pipe volume into cities (Qin)	0.8e4 cu.meter/min)	Saturation value of Qin for water pipes taking water from reservoir to each city

Table 2. Historical Time Demand Values for Water Distribution System.

Time of Day (minutes)	Buena Vista (cu. meters/min)	Crystal City (cu. meters/min)
0	120	230
100	890	1100
200	1450	2140
300	1780	2300
500	540	867
750	450	1200
900	890	1300
1100	760	1020
1250	505	860
1440	120	230

Task 1

Simulate the water system for 5 days continuously (i.e., 7,200 minutes) to observe the dynamic behavior of the system. Using Matlab plot commands (not Simulink) show graphically the variations of headway of each reservoir as a function of time.

Task 2

Explain what are the approximate steady-state values of the reservoir volumes.

Task 3

If the demand function of Buena Vista in 20 years increases by a factor of three compared to today, can the system function correctly? Comment.

Solution

The solution to this problem is similar to that solved in class.

Task 4

Suppose we want to make the water distribution system more reliable. Figure 2 illustrates a new network configuration suggested. In this new system, water can be transferred between reservoirs to supply any deficit if the water demands at either city are excessive. Moreover, the system can cope with a failure of one of the pumps feeding either one of the reservoirs. Discuss (do not model) in detail the logic of how would model the interactions of the new system. Tell me using logical arguments (use a flowchart to explain) how would you control the activation of pump 3 based on the states of reservoirs 1 and 2.

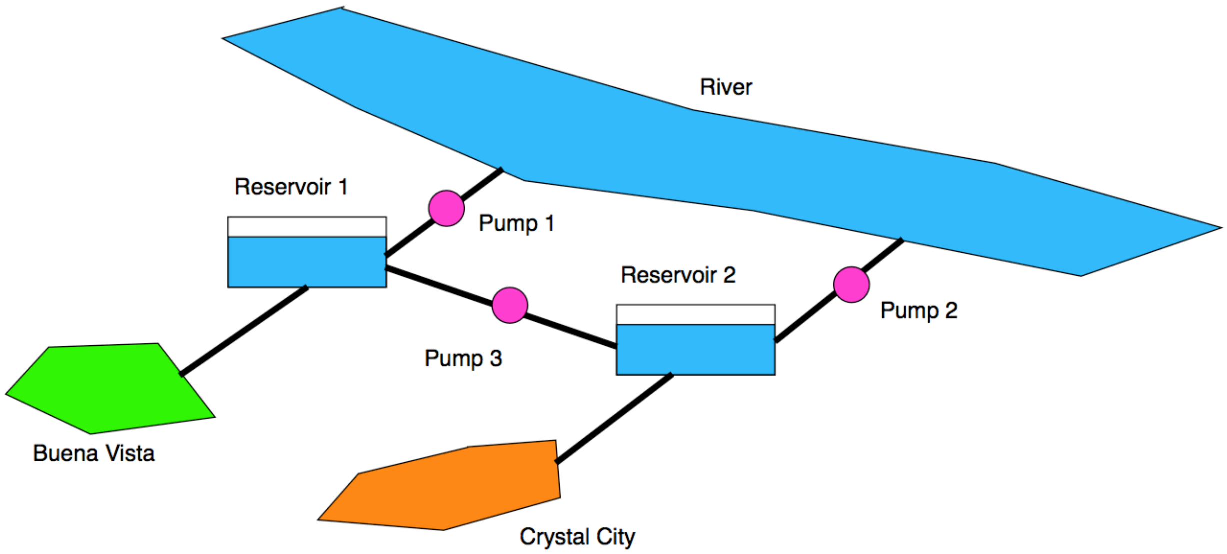

Figure 2. Modified Water Distribution System.

Problem 2

An Excel file contains data for thousands of public airports in the U.S. A sample of the file is shown below. The file is called airports_2011.xls.

Airport ID	Name	State	Passengers per Year	Latitude (deg)	Longitude (deg)
00V	MEADOW LAKE	CO	0	38.94574889	-104.5698933
01G	PERRY-WARSAW	NY	0	42.74134667	-78.05208056
02A	GRAGG-WADE FIELD	AL	0	32.85048333	-86.61143611
02C	CAPITOL	WI	0	43.08751	-88.17786917
02G	COLUMBIANA COUNTY	OH	0	40.67331278	-80.64140639
04Y	HAWLEY MUNI	MN	0	46.88382083	-96.35025972
05C	GRIFFITH-MERRILLVILLE	IN	0	41.51984083	-87.39950778

Task 1

Create a Matlab script and use the Matlab function xlsread to read the data file into a Matal Cell Array. Rename the variables with names that represent the data. The airport name is a 3-letter code used by the Federal Aviation Administration to name airports (column 1).

Task 2

Add to the script created in Task 1 Matlab code to estimate the number of airports in California, Montana and Washington State. Use the string comparison function in Matlab (strcmp) as needed.

Task 3

Add to the script in Task 2 to plot the location of all the airports in the U.S. Use the usamap.mat file provided. Label your map as needed.

Task 4

Find the total number of passengers boarding planes at airports in the state of California.

Solution

The solution to this problem is similar to that posted for Assignment 7 in 2011. See a7_cee3804_2011_sol.pdf under 2011 homework solutions.

Problem 3

An engineer collects data during the certification of the new high-speed train to be introduced in the Northeast Corridor in the United States. The data collected records train acceleration (a) vs. velocity (V). The data is presented in Table 3. The trains are shown in Figure 4.

Table 3. Train Acceleration and Speed Data.

Train Velocity (m/s)	Maximum Train Acceleration (m/s^2)
0.00	2.1
20	1.56
30	1.30
40	1.06
50	0.76
60	0.51
80	0.00

Figure 4. Acela High-Speed Trains.

Task 1

Use the Matlab script created in HW 7 to find the best second-order polynomial that fits the acceleration vs. train speed data (i.e., use the "polyfit" command). The resulting polynomial will be of the form:

$$\frac{dV}{dt} = A + BV + CV^2 \qquad (1)$$

where A, B and C are the polynomial coefficients found and V is the train speed.

```
% Script to estimate best curve fit for two vectors
clear
clc

% T. Trani

% Task 1

% Define two vectors for velocity and acceleration

velocity      = [0 20 30 40 50 60 80] ;                    % velocity in m/s
acceleration  = [2.1 1.56 1.30 1.06 0.76 0.51 0.00];   % accceleration (m/s-s)

% Do a  basic polynomial fit

coefficients = polyfit(velocity,acceleration,2);            % fits a second order polynomial

% Evaluate the polynomial found for the range of velocities of the train in
% the table

velNew = min(velocity):1:max(velocity);      % define a new velocity vector to evaluate the polynomial
accelerationFromPolyFit = polyval(coefficients,velNew);   % evaluate the polynomial using the coefficients found

% Make a plot and compare

% Create a label for the plot with the values of coefficidents found

labelPlot = horzcat('Acceleration = ', num2str(coefficients(1)), ' * Velocity^2 + ', ...
    num2str(coefficients(2)), ' * Velocity + ', num2str(coefficients(3)));

figure
plot(velocity,acceleration,'or',velNew,accelerationFromPolyFit,'b--')
xlabel('Train Velocity (m/s)','fontsize',20)
ylabel('Train Acceleration (m/s-s)','fontsize',20)
title(labelPlot)
grid
```

Figure 1. Matlab script used in Problem 3 of Assignment 7.

```
% Previously obtained coefficients are:

A = 0.0;         % coefficient of acceleration function (2nd power)
B = -0.0268;    % coefficient of acceleration function (1st power)
C = 2.0997;     % coefficient of acceleration function (constant)
```

Task 2

Using the Matlab Ordinary Differential Equation solver ODE45, to solve numerically the differential equation (1) as a function of time. This problem is similar to the Water Cooling problem discussed in class except that the differential equation is a little more complex. Use as initial conditions zero for the train speed and solve numerically the speed of the train for 200 seconds. Plot the velocity profile of the high-speed train as a function of time. How fast is the train going after 200 seconds?

Task 3

Add code to the script and function containing the differential equation created in Task 2 to calculate the distance traveled by the train. Recall that distance (S) can be obtained from the first order differential equation:

$$\frac{dS}{dt} = V \qquad (2)$$

The solution to this problem requires solving two first order equations (1-2). Refer to the mass-spring damper system discussed in class to help you setup these equations. You can see how these two equations are coupled as follows:

Let x_1 be the speed of the train, x_2 be the position of the train and $\dot{x}_1$ and $\dot{x}_2$ be the derivatives of speed and position . The set of first-order differential equations describing the motion of system are:

$$\dot{x}_1 = \frac{dV}{dt} = A + Bx_1 + Cx_1^2$$

$$\dot{x}_2 = x_1 = \frac{dS}{dt}$$

In solving these equations, plot the profiles for velocity and distance as a function of time. Label as needed. Using the results find the distance traveled by the train after 100 seconds.

Solution

I combined Tasks 2 and 3 in the same Matlab code. The main script is shown in Figure 2.

```
% Main file to solve two differential equations of motion
% Solution to a set of dynamic equations of the form:
%
% define the following state equations
%
% x(1) = speed (m/s)
% x(2) = position (m)

% xdot(1) = A + B *x(1) + C * x(1) .^2;   % acceleration of train
% xdot(2) = x(1);                          % velocity of train
%
% subject to initial conditions:
%
% x (t=0) = xo
%
% where:

global A B C

% Define Initial Conditions of the Problem

xo = [0 0];                 % xo are the initial velocity and distance traveled
to = 0.0;                   % to is the initial time to solve this equation
tf = 200;                   % tf is the final time

% define the coefficients of the acceleration function (A, B, and C)

% Previously obtained as: coefficients = 0.0000   -0.0268    2.0997
% from highest order to lowest (A associated with x(1) ^2, B with x(1) and
% C if the constant term)

A = 0.0;          % coefficient of acceleration function (2nd power)
B = -0.0268;      % coefficient of acceleration function (1st power)
C = 2.0997;       % coefficient of acceleration function (constant)

tspan =[to tf];

[t,x] = ode45('trainDynamics',tspan,xo);    % call ODE solver
```

Figure 2. Main Matlab script to solve the high-speed train equations.

```matlab
% Plot the results of the numerical integration procedure

figure
plot(t,x(:,1))
xlabel('Time (seconds)','fontsize',20)
ylabel('Velocity Profile of the Train (m/s)','fontsize',20)
grid


figure
plot(t,x(:,2))
xlabel('Time (seconds)','fontsize',20)
ylabel('Distance Traveled (m)','fontsize',20)
grid
```

Figure 3. Continuation of main Matlab script to solve the high-speed train equations.

```matlab
% Two first-order DEQ to solve two equations of motion for the train

function xdot = trainDynamics(t,x)

global A B C

% define the rate equations
%
% x(1) = speed (m/s)
% x(2) = position (m)

xdot(1) = A *  x(1) .^2 + B *x(1) + C ;
xdot(2) = x(1);

xdot = xdot';
```

Figure 4. Function containing rate equations for high-speed train equations.

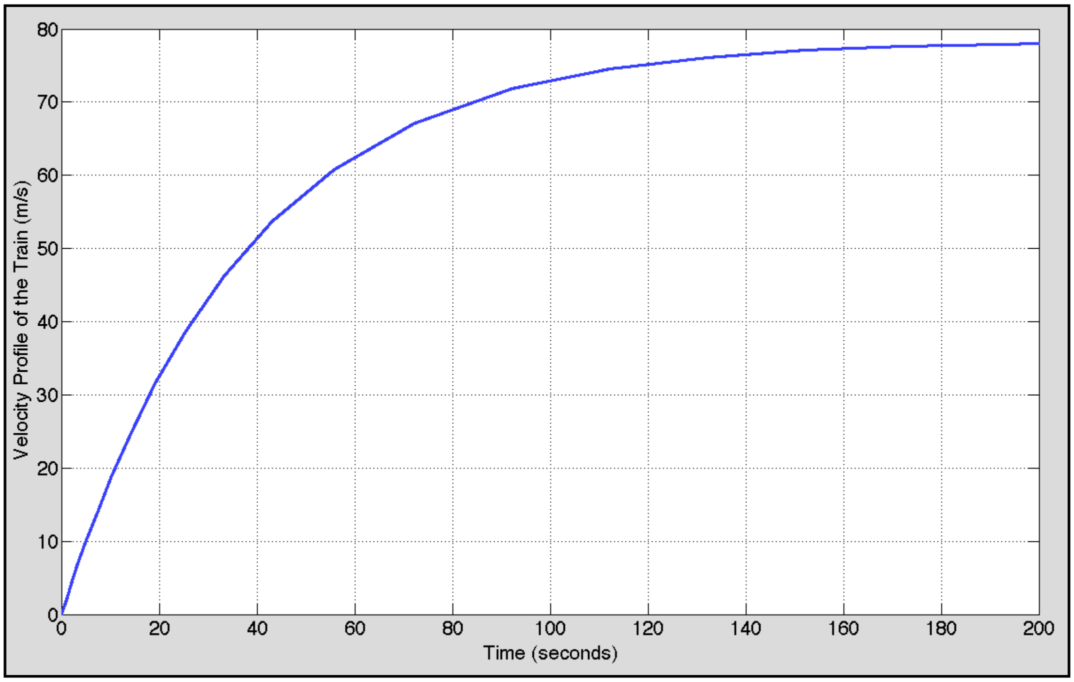

Figure 5. Velocity profile of the High-Speed train in 200 seconds.

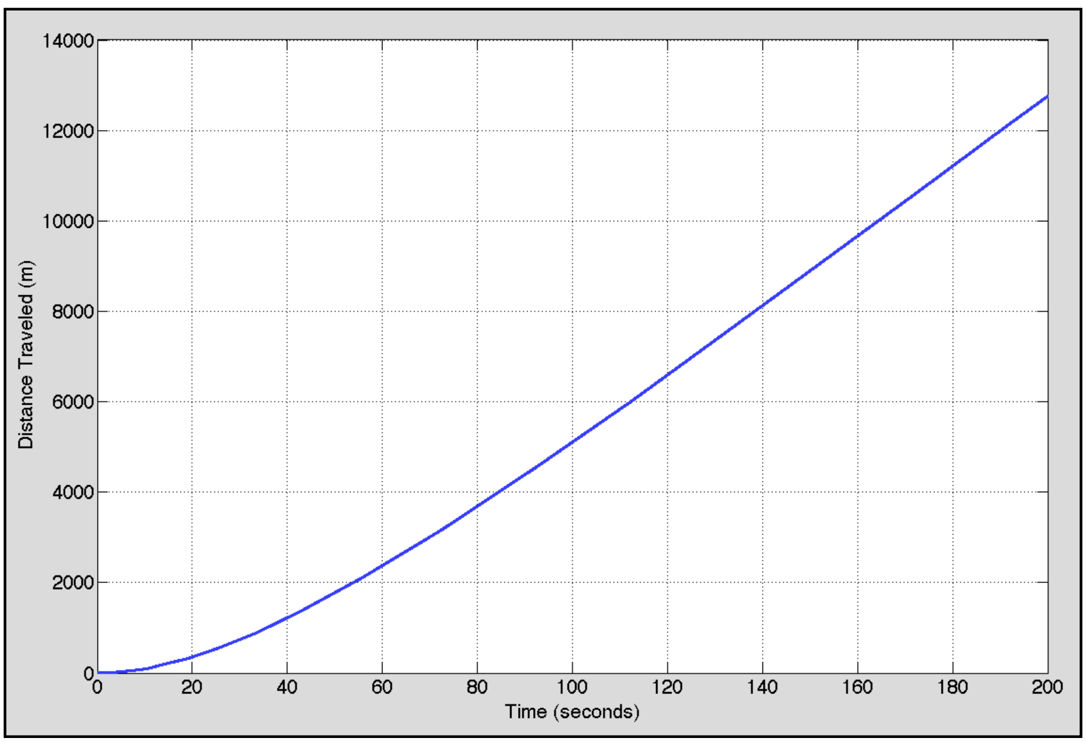

Figure 6 Distance traveled by High-Speed train in 200 seconds.

The train travels at 78 ms after 200 seconds.