

Problem Set 6

ENGR 12, Spring 2026.

Due Date	Thu, Mar 05, 2026
Turn in link	Gradescope
URL	emadmasroor.github.io/E12-S26/Homework/HW5

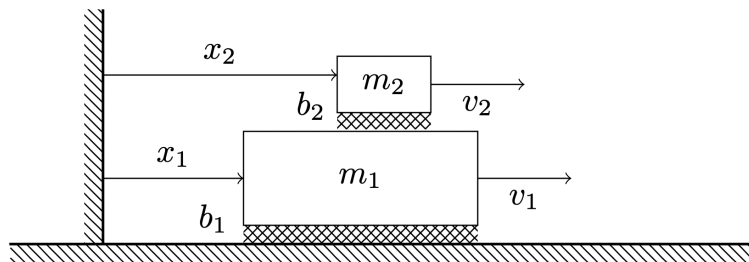
Points Distribution

Please note that each of the following grade items is a **single rubric item**. Each rubric item is scored on a four-level scale of 3-2-1-0. You may wish to take this into account when deciding how to allocate your efforts to each problem.

Problem	Part	% Weightage
Problem 1		20
Problem 2	2.1	20
	2.2	20
	2.3	20
Problem 3		20

1 A coupled first-order system

In [HW 1](#), we looked at the system of equations governing the (somewhat idealized) motion of one box on top of another box as shown in the diagram below.



The governing equations for the system shown schematically above are

$$\begin{aligned} m_1 \dot{v}_1 + b_1 v_1 &= 0 \\ m_2 \dot{v}_2 + b_2 (v_2 - v_1) &= 0 \end{aligned} \quad (1)$$

1.1 Deriving the equations

Express Equation 1 in the form of Equation 3, using an appropriate set of state variables.

1.2 Solving in MATLAB

Solve these equations using `ode45`, starting from the [skeleton code](#) given below. Use parameter values $m_1 = 30$, $m_2 = 3$, $b_1 = b_2 = 5$, and initial conditions $v_1 = 5$ and $v_2 = 0$ at $t = 0$. Continue your plot up to time $t = 10$.

You need not use exactly the same state variables that you chose in the previous part; use whatever set of state variables seems convenient and intuitive.

Your plot should look like [the one that you saw in HW 1](#).

Warning

The given code has a serious mistake, but it should be pretty straightforward to fix.

```
% Define the system of ODEs
% y(1) = v1, y(2) = v2
function dvdt = odefun(t, y)
    % Parameters
    m1 = 30;
    m2 = 3;
    b1 = 5;
    b2 = 5;
```

```

% Unpack
v1 = y(1);
v2 = y(2);

% Define rhs of differential equations
% m1*v1' + b1*v1 = 0 => v1' = -b1*v1 / m1
% m2*v2' + b2*(v2 - v1) = 0 => v2' = -b2*(v2 - v1) / m2
dvdt = [(-b1 * v1) / m1;
        0.5];
end

% Solve
% Initial conditions: v1(0) = 5, v2(0) = 0
[t, sol] = ode45(@odefun, [0, 10], [5; 0]);

% Extract results
v1 = sol(:,1);
v2 = sol(:,2);

% Create the plot
figure(1); clf;
plot(t, v1, t, v2); grid on; grid minor;
legend('v_1', 'v_2');
xlabel('Time t');
ylabel('Velocity');

```

2 State-variable form and MATLAB

For each of the given systems of equations, complete the following tasks:

1. Identify the n state variables
2. Write the equations in the form

$$\frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f_1(y_1, y_2, \dots, y_n) \\ f_2(y_1, y_2, \dots, y_n) \\ \vdots \\ f_n(y_1, y_2, \dots, y_n) \end{bmatrix} + \begin{bmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_n(t) \end{bmatrix} \quad (2)$$

3. If possible, re-write the equations in the form

$$\frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} + \begin{bmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_n(t) \end{bmatrix} \quad (3)$$

4. Write down the MATLAB (or Python) code that will successfully solve the initial value problem.

2.1

$$\begin{aligned}\dot{x}_1 x_2 + 5x_2 &= -3x_1 x_2 - 2x_2 t \\ \dot{x}_2 + 3x_1 &= 7x_2\end{aligned}$$

With initial conditions $x_1(0) = 1$ and $x_2(0) = -3$. We are interested in the behavior of this system up to time $t = 5$.

2.2

$$\begin{aligned}3\dot{x} + 5y &= 2x - \dot{y} \\ 2\dot{y} - 7x &= 4y + \dot{x}\end{aligned}$$

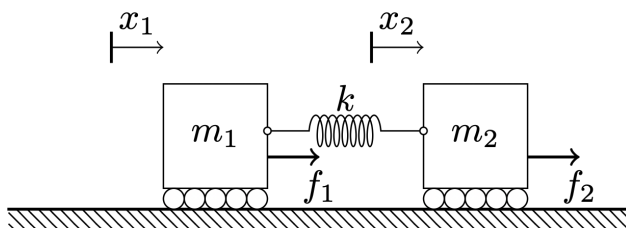
With initial conditions $x(0) = 1$ and $y(0) = 1$. We are interested in the behavior of this system up to time $t = 5$.

2.3

$$\begin{aligned}\ddot{x} &= -2\ddot{x} - 3\dot{x} + 5x - t \\ x(0) &= 1 \\ \dot{x}(0) &= -1 \\ \ddot{x}(0) &= 0\end{aligned}$$

3 Choosing the right state variables

In class, we discussed the system shown below.



Warning

In this problem, please **ignore** f_1 and f_2 . They are both equal to zero for now. Later, we will make use of these ‘input functions’ as well, but you’re off the hook for now!

3.1 Developing the equations

Write down the governing differential equations for this system in three forms:

1. As two second-order differential equations for $x_1(t)$ and $x_2(t)$
2. As four first-order differential equations in state-variable form, arranged in the format of Equation 2. Name your four state variables y_1 , y_2 , y_3 and y_4 .
3. As **three** first-order differential equations in the state-variable form of Equation 2, using the following state variables:

$$z_1 = \dot{x}_1, \quad z_2 = \dot{x}_2, \quad z_3 = x_2 - x_1$$

💡 Tip

Assume that the ‘rest length’ is already accounted for in the definition of x_1 and x_2 . Therefore, the rest length should not explicitly appear in your equations.

3.2 Solving the equations

By modifying the **MATLAB code provided on the website**, write a program that simulates what happens to this system and plot your result. Your plot should show three quantities on the same set of axes, and should go from $t = 0$ to $t = 15$. The three quantities are z_1 , z_2 and z_3 as defined above. Use the following values for the parameters: $m_1 = m_2 = k = 1$. The initial conditions are as follows:

- Mass 1 is initially moving at a velocity of 0.5 to the right
- Mass 2 is initially at rest
- The distance between Mass 2 and Mass 1, i.e., $x_2 - x_1$, is initially equal to 1.0

You can ignore units for the purpose of this problem. Note that the position of the two masses with parameters and initial conditions as given above is given by the following graph.

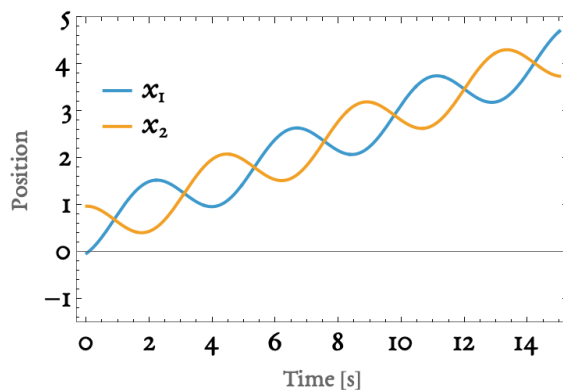


Figure 1

The figure above can be produced using the following code.

3.3 MATLAB

```
% Define the system of ODEs
% y(1) = x1, y(2) = dx1/dt, y(3) = x2, y(4) = dx2/dt
function dydt = odefun(t,y)
    % Parameters
    m1 = 1; m2 = 1; k = 1;

    % Unpack
    x1    = y(1);
    x1dot = y(2);
    x2    = y(3);
    x2dot = y(4);

    % Define rhs of diff. eq.
    dydt = [x1dot;
            k*(x2-x1)/m1;
            x2dot;
            -k*(x2-x1)/m2];
end

% Solve for t = 0 to t= 15, with initial conditions
% on x1 = 0, x2 = 1, x1' = 0.5, x2' = 0
[t, sol] = ode45(@odefun, [0,15], [0; 0.5; 1; 0]);

% Extract results for plotting
x1 = sol(:,1);
x2 = sol(:,3);

% Create the plot
figure(1); clf;
hold on;
plot(t, x1, 'r-', 'LineWidth', 0.5);    % x1
plot(t, x2, 'b-', 'LineWidth', 0.5);    % x2

% Formatting
xlabel('Time [s]', 'FontSize', 16);
ylabel('Position', 'FontSize', 16);
grid on; grid minor;
set(gca, 'FontSize', 14);
ylim([-1.5, 5]);

legend({'x_1', 'x_2'}, ...
      'Location', 'northwest', 'FontSize', 12);
```

3.4 Mathematica

```
Module[{sol}, With[{m1 = 1, m2 = 1, k = 1, x01 = 0, x02 = 0},
  sol = NDSolve[
    {
      m1 x1''[t] == k (x2[t] - x1[t] - x01),
      m2 x2''[t] == -k (x2[t] - x1[t] - x02),
      x1[0] == 0, x2[0] == 1, x1'[0] == 0.5, x2'[0] == 0
    }, {x1, x2, x1', x2'}, {t, 0, 100}]
];
sol1 = sol[[1]];
]

Plot[Evaluate[{x1[t], x2[t]} /. sol1], {t, 0, 15},
  FrameLabel -> (Style[#, FontFamily -> "EB Garamond",
    FontSize -> 16] & /@ {"Time [s]", "Position"}),
  PlotLegends ->
    Placed[Style[#, FontFamily -> "EB Garamond",
      FontSize -> 20] & /@ {"x1",
        "x2"}, {0.12, 0.7}],
  Frame -> True,
  FrameTicksStyle ->
    Directive[Black, "Label", 20, FontFamily -> "EB Garamond"],
  PlotLabel -> Style["", FontSize -> 16], PlotRange -> {All, {-1.5, 5}}]
```

What to turn in

Turn in a single plot, generated by MATLAB, according to the specifications above. Please also provide your code embedded into your PDF submission in a readable format. You should either use a syntax-highlighting tool, or take a screenshot. The code that you provide should correspond exactly to your plots. (We won't type in your code on our computers, but we will read over it). Finally, please provide a short explanation of the relationship between the graph that you have made (which shows z_1 , z_2 and z_3) and the graph provided above (which shows x_1 and x_2).

To summarize, you need:

1. A figure
2. Code
3. A couple of sentences-long explanation of what your figure shows.