

UNIVERSITY OF CALIFORNIA, RIVERSIDE

BOURNS COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

---

**EE 105 Lab 2 Solution**  
MATLAB as an Engineer's Problem Solving Tool

---

**LUIS FERNANDO ENRIQUEZ-CONTRERAS**



## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Pre-Lab</b>	<b>3</b>
<b>3</b>	<b>LSIM and gensig</b>	<b>4</b>
<b>4</b>	<b>Varying Step-Size of <math>1.0 \sin(0.1)</math></b>	<b>7</b>
<b>5</b>	<b>Varying <math>\omega</math> in <math>1.0 \sin(\omega t)</math></b>	<b>14</b>
<b>6</b>	<b>Conclusion</b>	<b>19</b>

## List of Figures

1	lsim Simulation of the State Space Model with a domain of 0-100 seconds . . . . .	6
2	lsim Simulation of the State Space Model with a domain of 0-4 seconds . . . . .	7
3	Euler Plot for $h = 1$ . . . . .	10
4	Euler Plot for $h = 0.1$ . . . . .	11
5	Euler Plot for $h = 0.05$ . . . . .	12
6	Euler Plot for $h = 0.01$ . . . . .	13
7	Euler Plot for $h = 0.001$ . . . . .	14
8	Ode Plot for $\omega = 1$ . . . . .	17
9	Ode Plot for $\omega = 0.09$ . . . . .	18
10	Ode Plot for $\omega = 0.001$ . . . . .	19

## Listings

1	lsim Simulation of the State Space Model with domain of 0-100 seconds . . . . .	4
2	lsim Simulation of the State Space Model with domain of 0-4 seconds . . . . .	5
3	Function for the State Space of the representation of the system . . . . .	7
4	Function to run Euler recursion . . . . .	8
5	Run the Euler function for different step sizes $h$ . . . . .	9
6	Function for the State Space of the representation of the system for ODE23 . . . . .	14
7	Run ODE23 for $\sin(0.1t)$ . . . . .	15
8	Run ODE23 for various $\sin(\omega t)$ . . . . .	15

# 1 Introduction

## 2 Pre-Lab

Given the transfer function:

$$H(s) = \frac{36}{s^2 + 3s + 36}$$

We can determine the following values:

$$\omega_n = 6$$

$$G = 1$$

$$\zeta = \frac{3}{12} = \frac{1}{4}$$

$$\sigma = \zeta \omega_n = \frac{6}{4} = 1.5$$

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \approx 5.2$$

The steady state response is given by:

$$\begin{aligned} H(s) &= \frac{36}{s^2 + 3s + 36} \\ H(s)|_{s=j\omega} &= \frac{36}{36 - \omega^2 + 3j\omega} \\ |H(j\omega)|^2 &= H(j\omega)H(j\omega)^* \\ &= \left( \frac{36}{36 - \omega^2 + 3j\omega} \right) \left( \frac{36}{36 - \omega^2 - 3j\omega} \right) \\ |H(j\omega)| &= \frac{36}{\sqrt{(36 - \omega^2)^2 + 9\omega^2}} \\ \angle H(j\omega) &= \left( \frac{36}{36 - \omega^2 + 3j\omega} \right) \left( \frac{36 - \omega^2 - 3j\omega}{36 - \omega^2 - 3j\omega} \right) \\ &= \frac{36(36 - \omega^2 - 3j\omega)}{(36 - \omega^2)^2 - 9\omega^2} \end{aligned}$$

The steady state response is given by the following:

$$y(t) = 0.999 \sin(0.1t - 0.9163^\circ)$$

Given that  $x = \begin{bmatrix} y & \dot{y} \end{bmatrix}^T$

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \ddot{y}(t) \\
\frac{Y(s)}{U(s)} &= \frac{36}{s^2 + 3s + 36} \\
Y(s)(s^2 + 3s + 36) &= 36U(s) \\
s^2 Y(s) + 3s Y(s) + 36 Y(s) &= 36U(s) \\
\mathcal{L}^{-1}[s^2 Y(s) + 3s Y(s) + 36 Y(s)] &= \mathcal{L}^{-1}[36U(s)] \\
\ddot{y}(t) + 3\dot{y}(t) + 36y(t) &= 36u(t) \\
\ddot{y}(t) &= 36u(t) - 3\dot{y}(t) - 36y(t) \\
\ddot{y}(t) &= 36u(t) - 3x_2 - 36x_1 \\
\dot{x}_2 &= 36u(t) - 3x_2 - 36x_1 \\
\dot{x} &= [x_2; 36u(t) - 3x_2 - 36x_1] \\
y &= [x_1]
\end{aligned}$$

The system is linear, therefore we can solve for  $A, B, C, D$

$$\begin{aligned}
A &= \begin{bmatrix} 0 & 1 \\ -36 & -3 \end{bmatrix} & B &= \begin{bmatrix} 0 \\ 36 \end{bmatrix} \\
C &= \begin{bmatrix} 1 & 0 \end{bmatrix} & D &= [0]
\end{aligned}$$

### 3 LSIM and gensig

**Listing 1:** lsim Simulation of the State Space Model with domain of 0-100 seconds

```

1 clear all;
2 clc;
3 % Define the transfer function of the system
4 H = tf(36,[1 3 36]);
5 % Define A,B,C,D
6 A = [0,1; -36,-3];
7 B = [0;36];
8 C = [1 0];
9 D = 0;
10 % Define state-space model
11 sys = ss(A,B,C,D);
12 % Define initial state

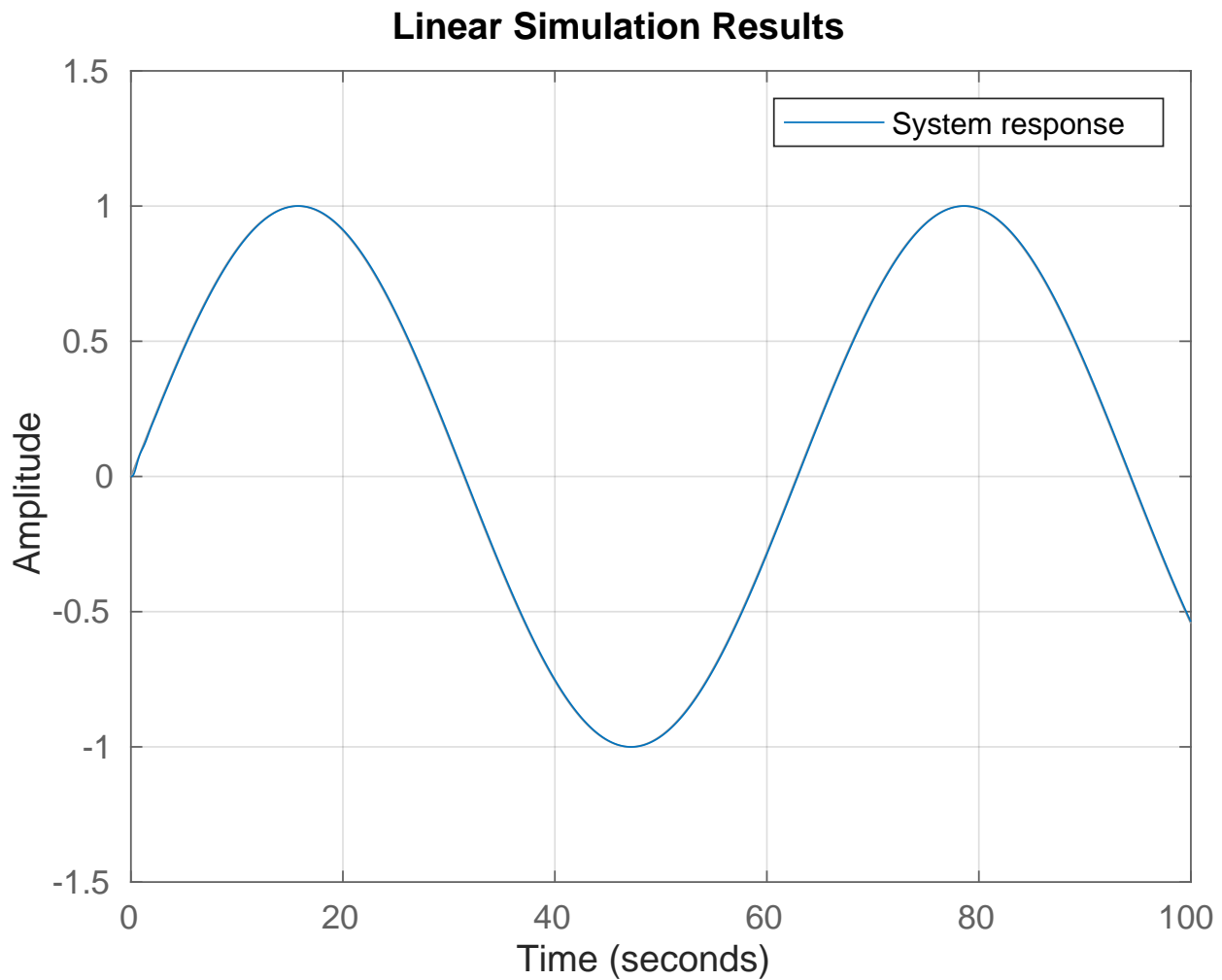
```

```
13 x0 = [0,0];
14 % Define the input u(t) = 1.0sin(0.1t) using sinusoidal signal
15 tau = 2*pi/0.1; % Period = 2*pi/0.1
16 % 0:Ts:Tf
17 Ts = 0.01; % Time step
18 Tf = 100; % Duration
19 [u,t] = gensig('sin',tau,Tf,Ts);
20 % Simulate the system
21 lsim(sys,u,t,x0);
22 grid on;
23 legend(System response);
24 fig = gcf; % Obtains current graphic in matlab
25 exportgraphics(fig, 'Fig/lsim_run_100s.pdf', 'ContentType','vector');
```

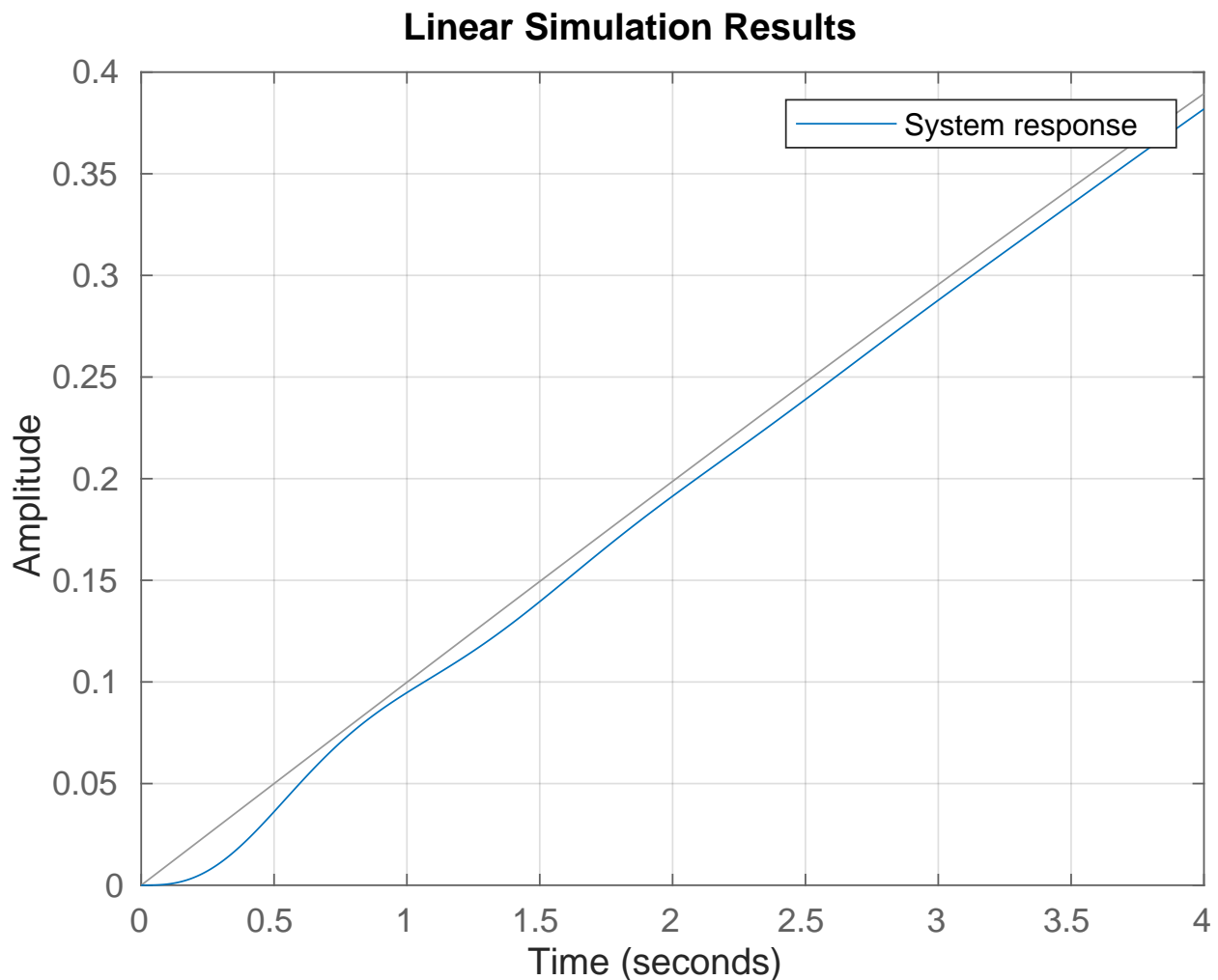
**Listing 2: lsim Simulation of the State Space Model with domain of 0-4 seconds**

```
1 clear all;
2 clc;
3 % Define the transfer function of the system
4 H = tf(36,[1 3 36]);
5 % Define A,B,C,D
6 A = [0,1; -36,-3];
7 B = [0;36];
8 C = [1 0];
9 D = 0;
10 % Define state-space model
11 sys = ss(A,B,C,D);
12 % Define initial state
13 x0 = [0,0];
14 % Define the input u(t) = 1.0sin(0.1t) using sinusoidal signal
15 tau = 2*pi/0.1; % Period = 2*pi/0.1
16 % 0:Ts:Tf
17 Ts = 0.01; % Time step
18 Tf = 4; % Duration
19 [u,t] = gensig('sin',tau,Tf,Ts);
20 % Simulate the system
21 lsim(sys,u,t,x0);
22 grid on;
23 legend(System response);
```

```
24 fig = gcf; % Obtains current graphic in matlab
25 exportgraphics(fig, 'Fig/lsim_run_4s.pdf', 'ContentType','vector');
```



**Figure 1:** lsim Simulation of the State Space Model with a domain of 0-100 seconds



**Figure 2:** lsim Simulation of the State Space Model with a domain of 0-4 seconds

## 4 Varying Step-Size of $1.0 \sin(0.1)$

**Listing 3:** Function for the State Space of the representation of the system

```

1 function [dx] = f(x,u)
2 % x — [2xn] column vector
3 % u — [1xn] vector
4 A = [0,1; -36,-3];
5 B = [0;36];

```



```

6 dx = A*x + B*u;
7 end

```

#### Listing 4: Function to run Euler recursion

```

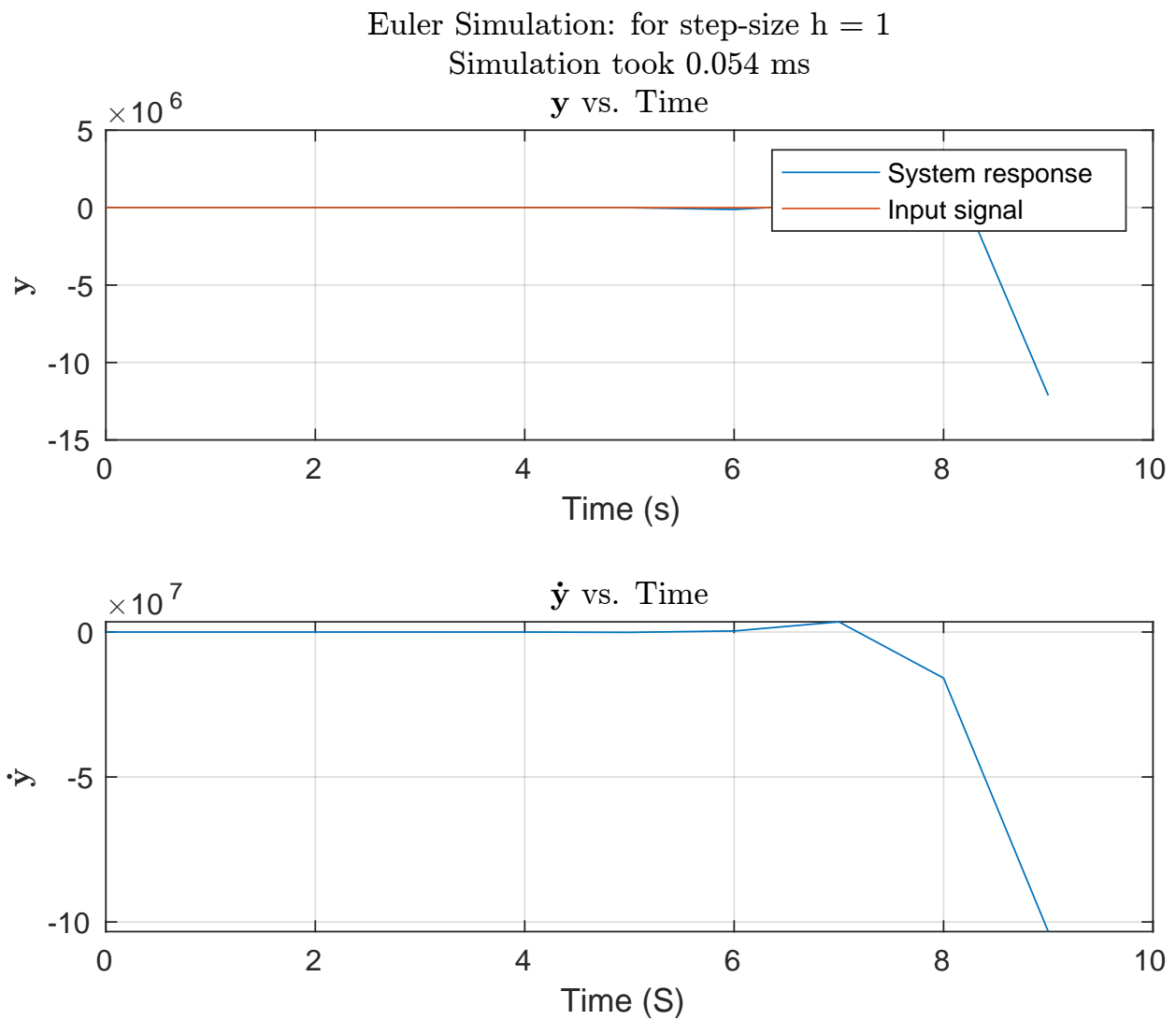
1 function sim_t = Euler(t,x0,h,u, filename)
2 % For the given initial condition x0 and step size
3 % h this function uses Euler integration to
4 % numerically solve the differential equation
5 % of the transfer function.
6 % Function output: sim_t, Euler Simulation time cost
7 tic; % start the clock
8 N = length(u); % The iteration steps based on the length of input signal
9 % Initialize x
10 x = zeros(length(x0),N); %The dimension of x in terms of dimension of x0
11 x(:,1) = x0; % IC
12 for i=1:N
13     [dx] = f(x(:,i),u(i));
14     x(:,i+1) = x(:,i) + dx*h;
15 end
16 sim_t = toc; % end the clock
17 sim_t = sim_t*1000;
18 display(append('Euler Simulation Took = ', string(sim_t) , 'ms'));
19 figure
20 subplot(2, 1, 1);
21 plot(t,x(1,1:i));
22 hold on;
23 plot(t,u);
24 legend('System response', 'Input signal');
25 plot_title = {[append('Euler Simulation: for step-size h = ', string(h))]} [
    append('Simulation took ', string(sim_t), ' ms')] ['$\bf y$ vs. Time']};
26 grid on;
27 title(plot_title, 'Interpreter', 'latex');
28 xlabel('Time (s)');
29 ylabel('$ \bf y$', 'Interpreter', 'latex');
30 subplot(2, 1, 2);
31 plot (t,x(2,1:i));
32 title ('y prime vs. Time', 'Interpreter', 'latex')
33 xlabel('Time (S)');
34 ylabel('y prime', 'Interpreter', 'latex');
35 grid on;

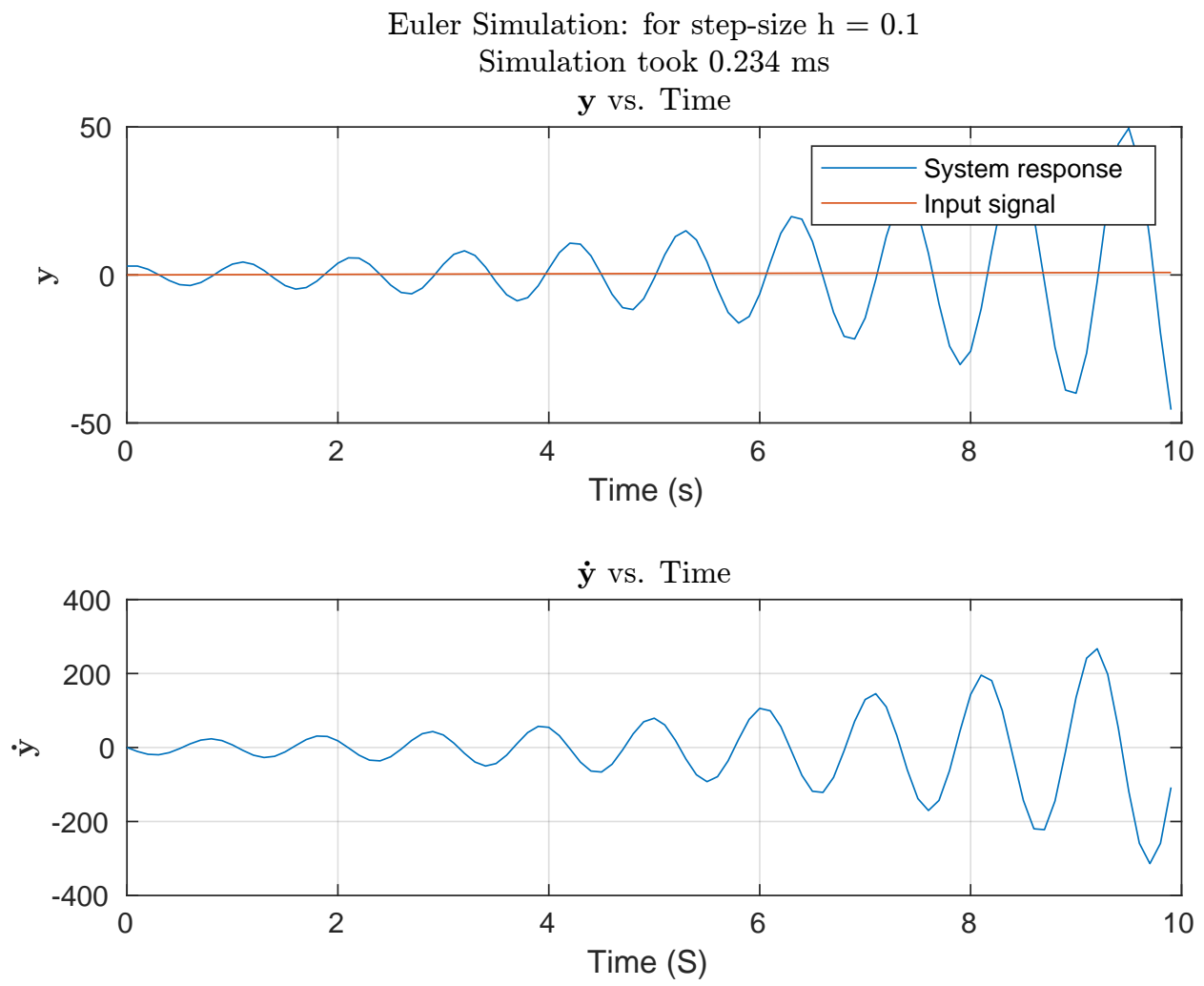
```

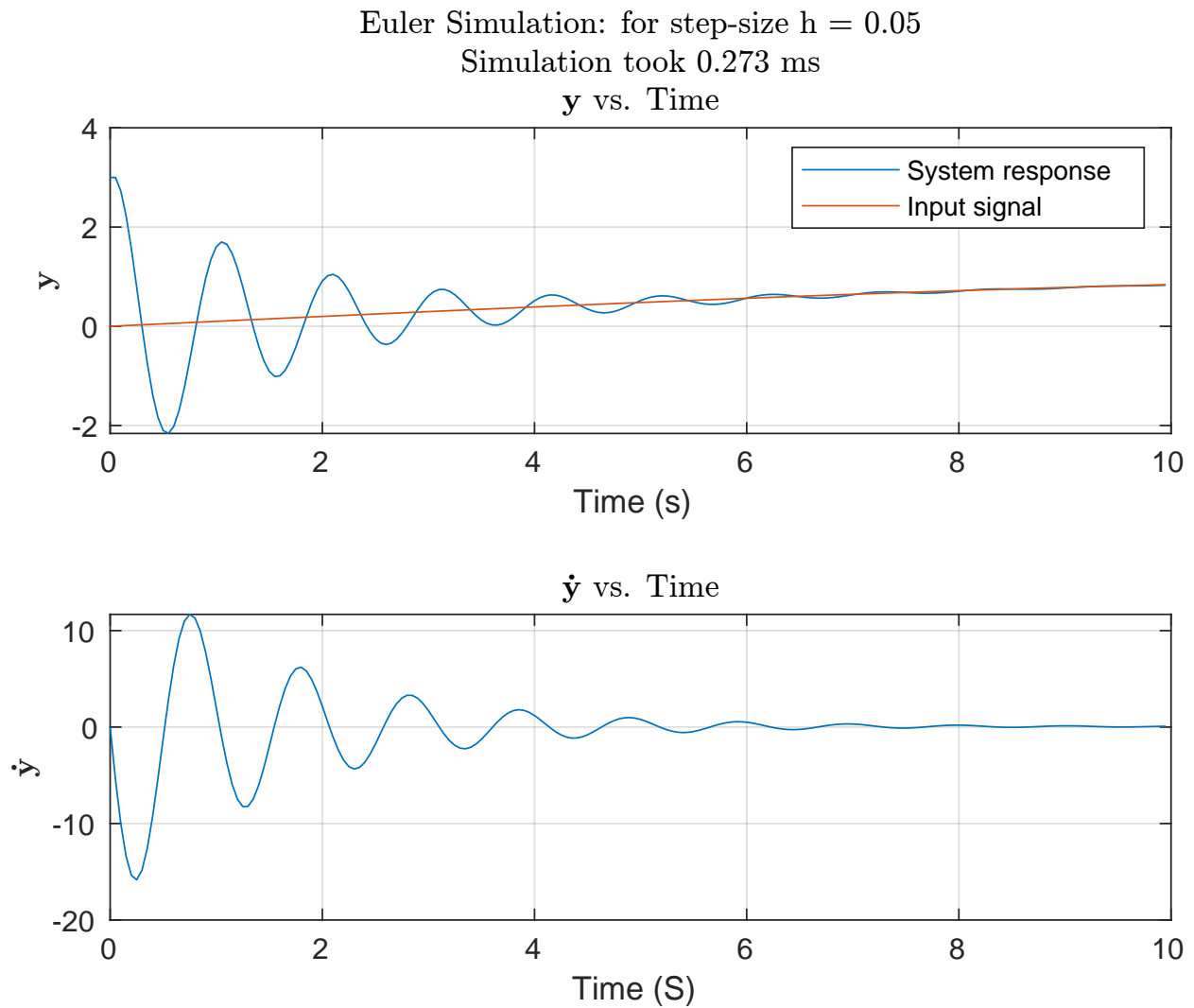
```
36 fig = gcf; % Obtains current graphic in matlab
37 exportgraphics(fig, filename, 'ContentType', 'vector');
38 end
```

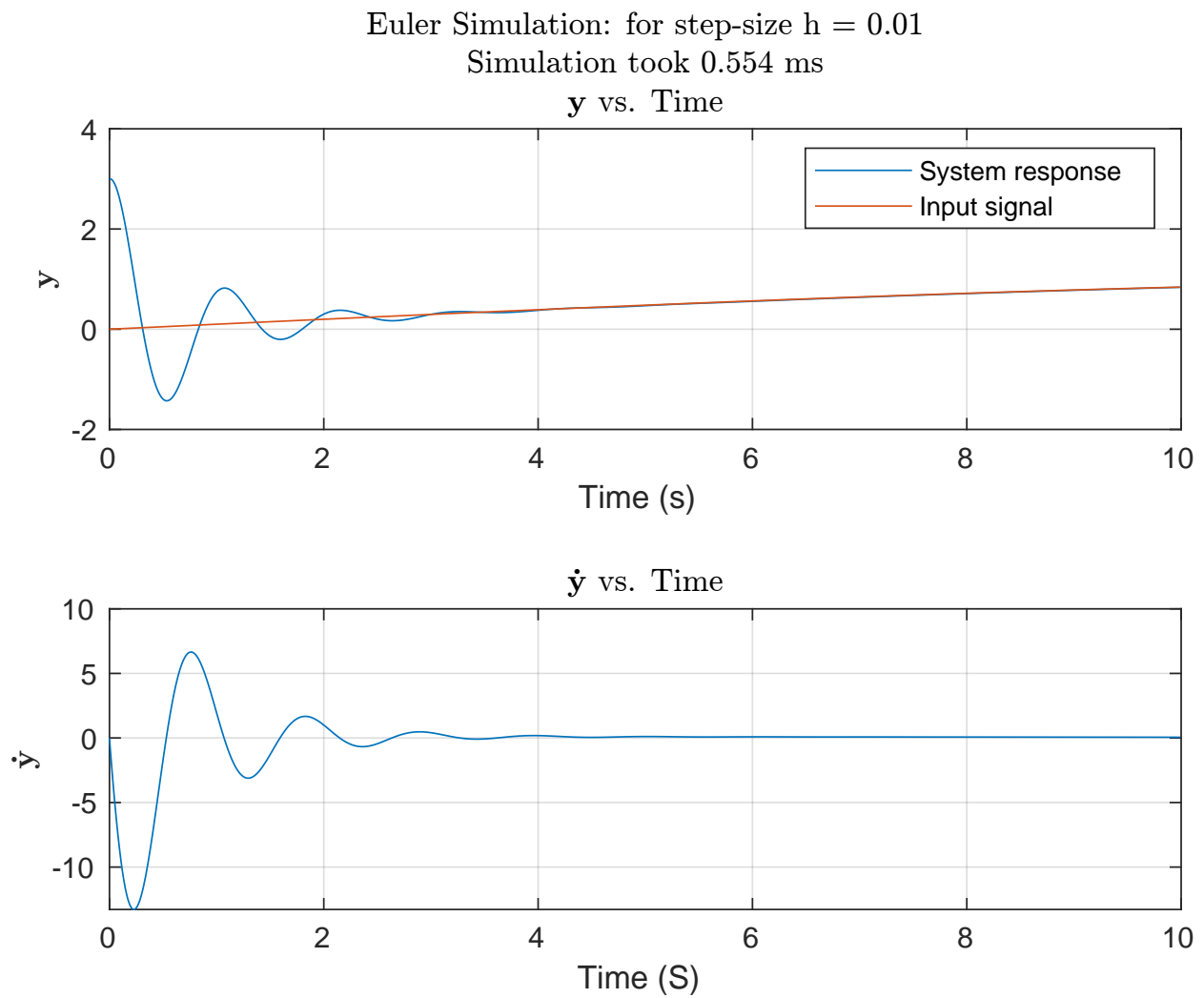
**Listing 5: Run the Euler function for different step sizes  $h$**

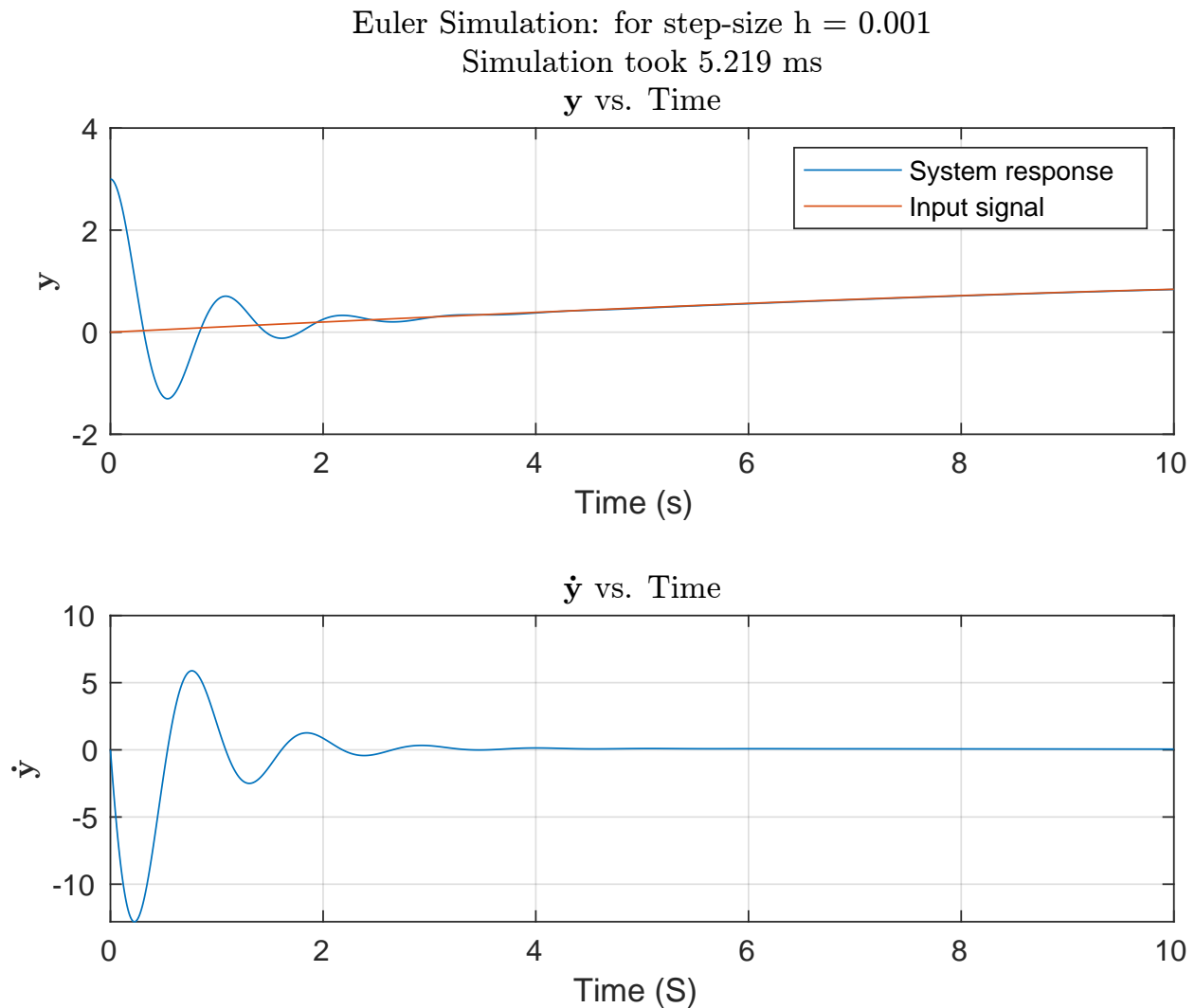
```
1 % Define the input u(t) = 1.0sin(0.1t) using sinusoidal signal
2 tau = 2*pi/0.1; % Period = 2*pi/0.1
3 % 0:Ts:Tf
4 Ts = 1; % Time step
5 Tf = 10; % Duration
6 % Define initial state
7 x0 = [3,0];
8 % Define step-size
9 % count = 1;
10 % h_str = ['1', '01', '005', '001', '0001'];
11 for h = [1, 0.1, 0.05, 0.01, 0.001]
12     % [u,t] = gensig('sin',tau,Tf,h);
13     N = (Tf/h);
14     t = h*(0:N-1);
15     u = sin(0.1*t);
16     filename = append('Fig/Euler_plot_h_', string(h), '.pdf');
17     sim_t = Euler(t,x0,h,u, filename);
18     count = count + 1;
19 end
```

**Figure 3:** Euler Plot for  $h = 1$

**Figure 4:** Euler Plot for  $h = 0.1$

**Figure 5:** Euler Plot for  $h = 0.05$

**Figure 6:** Euler Plot for  $h = 0.01$

Figure 7: Euler Plot for  $h = 0.001$ 

## 5 Varying $\omega$ in $1.0 \sin(\omega t)$

**Listing 6:** Function for the State Space of the representation of the system for ODE23

```

1 function [dx, u] = f(t,x,omega)
2 % x — [2xn] column vector
3 % u — [1xn] vector
4 A = [0,1; -36,-3];

```

```

5 B = [0;36];
6 u = sin(omega*t);
7 dx = A*x + B*u;
8 end

```

### Listing 7: Run ODE23 for $\sin(0.1t)$

```

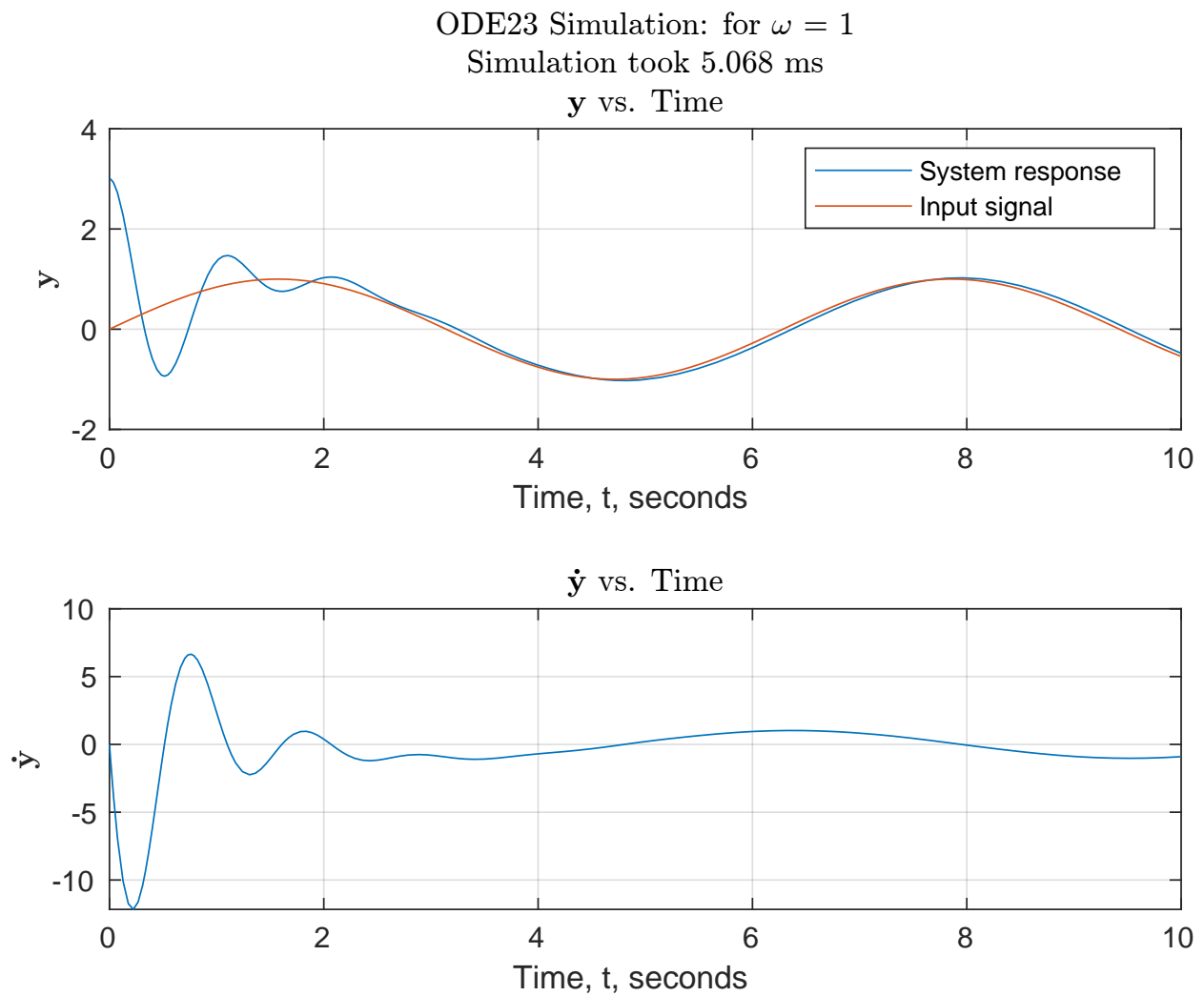
1 tspan = [0 10]; % Interval of integration
2 x0 = [3 0]; % Initial condition
3 omega = 0.1;
4 tic
5 [t_out,y] = ode23(@(t,x) f_ode(t,x,omega), tspan, x0);
6 tm = toc;
7 tm = tm*1000;
8 h = length(t_out) \ (tspan(2) - tspan(1));
9 t = tspan(1):h:tspan(2);
10 u = sin(omega*t);
11 display(append('ODE23 Simulation Took ', string(tm), 'ms'));
12 figure;
13 subplot(2, 1, 1);
14 plot(t_out, y(:,1));
15 hold on;
16 plot(t,u);
17 legend('System response', 'Input signal');
18 plot_title = {[append('ODE23 Simulation: for omega = ', string(omega))] [
    append('Simulation took ', string(sim_t), ' ms')] ['$ \bf y$ vs. Time']};
19 title(plot_title, 'Interpreter', 'latex');
20 xlabel('Time, t, seconds');
21 ylabel('$ \bf y$', 'Interpreter', 'latex');
22 grid on;
23 subplot(2, 1, 2);
24 plot(t_out, y(:,2));
25 title('y prime vs. Time', 'Interpreter', 'latex');
26 xlabel('Time, t, seconds');
27 ylabel('y prime', 'Interpreter', 'latex');
28 grid on;
29 fig = gcf; % Obtains current graphic in matlab
30 exportgraphics(fig, 'Fig/ode_sin_input_01.pdf', 'ContentType', 'vector');

```

### Listing 8: Run ODE23 for various $\sin(\omega t)$

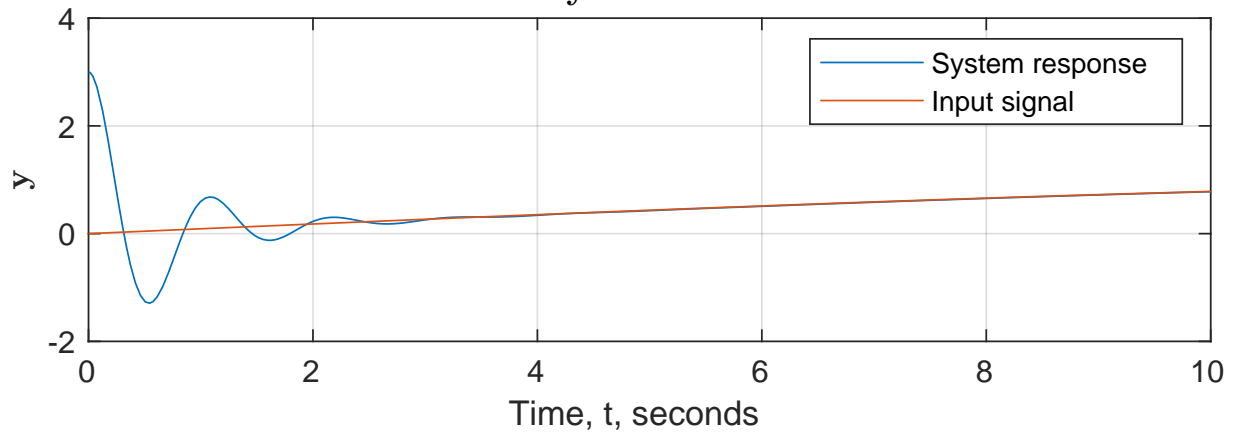
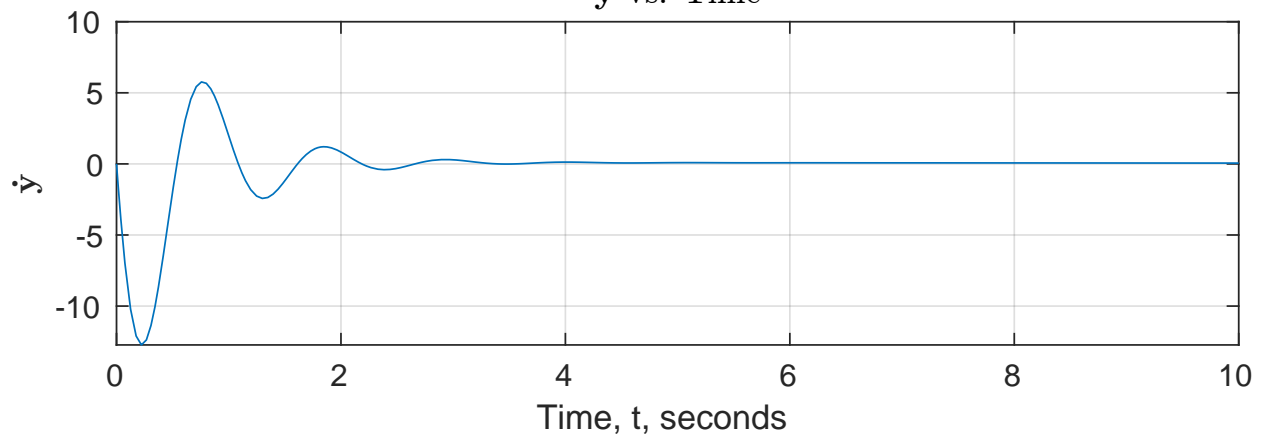


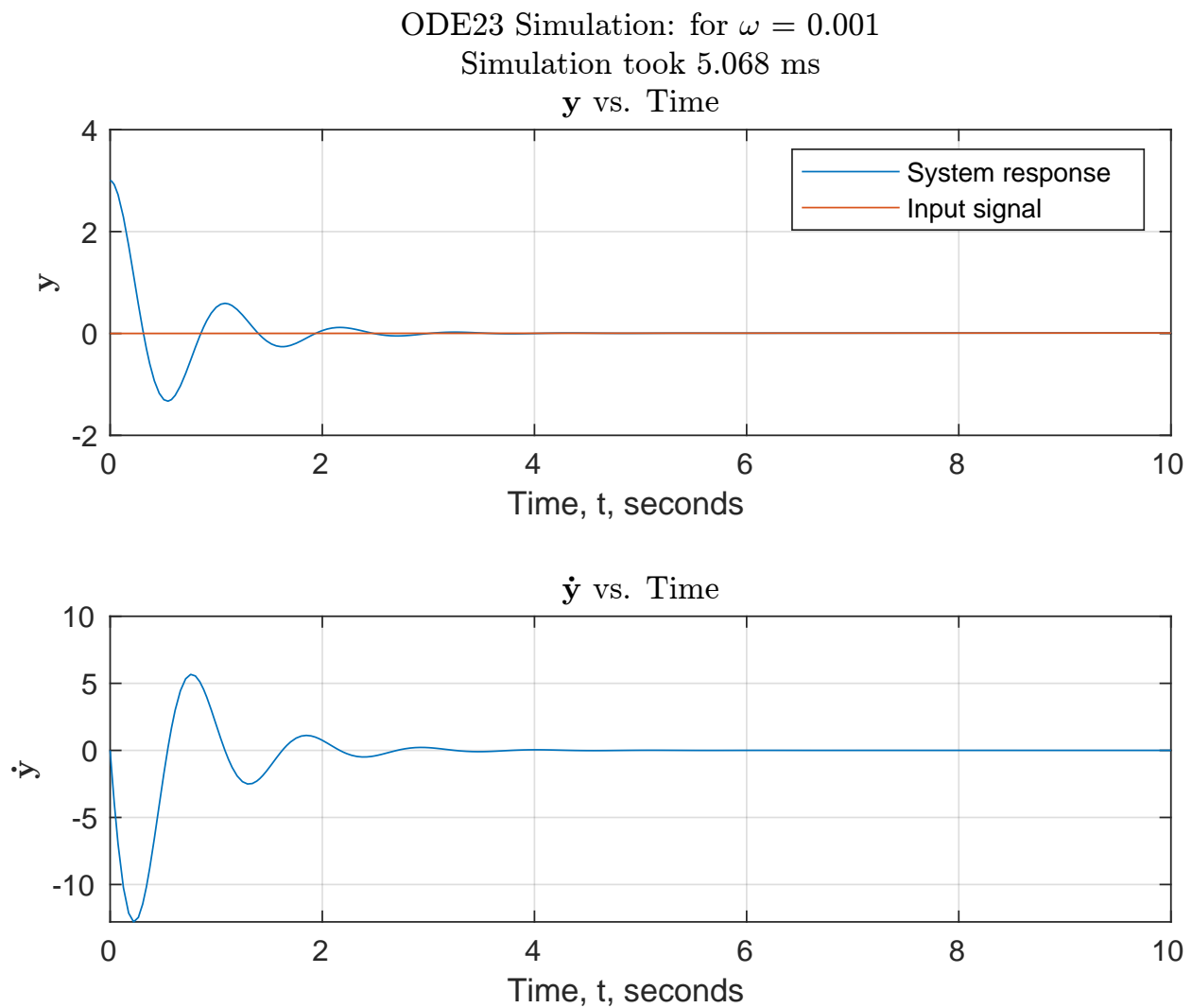
```
1 for i = [0.001, 0.09, 1]
2     tspan = [0 10]; % Interval of integration
3     x0 = [3 0]; % Initial condition
4     omega = i;
5     tic
6     [t_out,y] = ode23(@(t,x) f_ode(t,x,omega), tspan, x0);
7     tm = toc;
8     tm = tm*1000;
9     h = length(t_out) \ (tspan(2) - tspan(1));
10    t = tspan(1):h:tspan(2);
11    u = sin(omega*t);
12    display(append('ODE23 Simulation Took ', string(tm), 'ms'));
13    figure;
14    subplot(2, 1, 1);
15    plot(t_out, y(:,1));
16    hold on;
17    plot(t,u);
18    legend('System response', 'Input signal');
19    plot_title = {[append('ODE23 Simulation: for omega = ', string(omega))]
20                  [append('Simulation took ', string(sim_t), ' ms')] ['$ \bf y$ vs.
21                      Time']];
22    title(plot_title, 'Interpreter', 'latex');
23    xlabel('Time, t, seconds');
24    ylabel('$ \bf y$', 'Interpreter', 'latex');
25    grid on;
26    subplot(2, 1, 2)
27    plot(t_out, y(:,2))
28    title ('y prime vs. Time', 'Interpreter', 'latex');
29    xlabel('Time, t, seconds');
30    ylabel('y prime', 'Interpreter', 'latex');
31    grid on;
32    filename = append('Fig/ode_sin_input_', string(omega), '.pdf');
33    fig = gcf; % Obtains current graphic in matlab
34    exportgraphics(fig, filename,'ContentType','vector');
35 end
```

**Figure 8:** Ode Plot for  $\omega = 1$

ODE23 Simulation: for  $\omega = 0.09$ 

Simulation took 5.068 ms

 $y$  vs. Time $\dot{y}$  vs. Time**Figure 9:** Ode Plot for  $\omega = 0.09$

**Figure 10:** Ode Plot for  $\omega = 0.001$ 

## 6 Conclusion