SIMULATION OF A MASS SPRING DAMPER MODEL IN PHASE VARIABLE

Ejiroghene Kelly Orhorhoro  
Department of Mechanical Engineering,  
Faculty of Engineering, Delta State Polytechnic,  
Otefe-Oghara, Nigeria,  
kelecom@yahoo.com

Monday Erhire Onogbotsere  
Department of Electrical/Electronic Engineering,  
Faculty of Engineering,  
Delta State Polytechnic,  
Otefe-Oghara, Nigeria

Aniekan Essienubong Ikpe  
Department of Mechanical Engineering,  
Faculty of Engineering,  
Coventry University, United Kingdom

ABSTRACT
This paper reported the research work carried on mass spring damper model in phase variable form. This research work applied Newton law of motion, differential equations, MATLAB simulation, and transfer function to model mass-spring-damper model in phase variable form. The scope of state phase variable block representation with Simulink standard was used to obtain a plot of the step response of the state space representation of the system while the SIMOUT block helped in writing the vector sample of the output values and the time response. The Simulink representation for the transfer function was done using a standard Simulink transfer function block in MATLAB. The results show that response in the unit step of state phase variable block representation of MSD system obtained from Simulink has initial value of zero and a final value of 0.33. This implies that the system is overdamped due to the nature of the response curve which starts from zero, reaches a maximum value and then returns to zero where it remains constant again over time.  

Keywords: Newton law of motion, transfer function, modelling, differential equation, simulation

INTRODUCTION
Simulation is an important tool for engineering design, and analysis of complex engineering systems. However, data computation in recent times are complicated and difficult without the application of modelling and simulation tools. For Linear Time Invariant (LTI) systems, problem arises both as model size or model dynamic range. Over the years, LTI has played a major role in the development of engineering and technological frame work. Theoretical and numerical data can be analyzed by means of LTI modeling and simulation tools to obtain the basic functional requirement needed in
a given system [1]. LTI models have found a widespread utilization in theoretical and numerical analyses of linear dynamic systems. Most importantly, the behaviours of dynamic systems can be characterized by LTI system models. The consistency of LTI system analyses with real systems has been proven for numerous applied science and engineering problems. It is obvious that analyses on the base of LTI systems still play a central role in system science. Deepened investigation on behaviours of LTI systems promises further contributions in term of modeling and comprehension of physical and electrical system behaviours.

LTI systems are often expressed as linear time-invariant differential equations such as the ones used in transfer function, stability test, bode plot, etc. It is convenient to represent LTI differential equations as a set of second-order differential equations to determine the poles of a second order system [2]. Transfer functions of second order systems have a standard form that is mathematically express as [3]:

\[ G(s) = \frac{Y(s)}{R(s)} = \frac{k_{ss} \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \]  

(1)

Where,
\( \zeta \) is the damping ratio
\( k_{ss} \) is the steady state gain
\( \omega_n \) is the natural frequency
\( Y(s) \) is output
\( R(s) \) is input

Although, Laplace transform and transfer functions are mainly preferred because of simplification of the system analyses [4]. The characteristic polynomial of the LTI systems provides a valuable tool for the analysis of the character of LTI system. In this research work, transfer function was applied. It involves the application of derivatives and integrals of non-integer orders, and it can be applied in basic engineering design (such as synthesis of linkages), mathematics, mechanics [5]. Mass-spring-damper system contains a mass, a spring with spring constant \( k \ [N=m] \) that serves to restore the mass to a neutral position, and a damping element which opposes the motion of the vibratory response with a force proportional to the velocity of the system, the constant of proportionality being the damping constant \( c \ [Ns=m] \) [6, 7]. An ideal mass spring-damper system is represented in Figure 1. This paper will make use of Newton law of motion, differential equations, MATLAB simulation, and transfer function to model mass-spring-damper model in phase variable form. (Refer Fig. 1)
By isolating the mass \( m \) in Figure 1, four forces will act on the body. The first force is as a result of the acceleration of the body. The second force is a result of the action of the spring, while the third force is a resultant effect from the damper, and the fourth is the external force on the mass. Figure 2 represents the free body diagram of the system. *(Refer Fig. 2)*

**METHODOLOGY**

**APPLICATION OF NEWTON’S LAWS OF MOTION**

Sum of forces in all directions is equal to zero. Therefore,

\[
\sum F_y = 0 \quad (2)
\]

\[
f - m\ddot{x} - kx - d\dot{x} = 0 \quad (3)
\]

\[
f = m\dddot{x} + kx + d\dot{x} \quad (4)
\]

By writing the terms as a function of time,

\[
m\ddot{x}(t) + d\dot{x} + k(x) = f(t) \quad (5)
\]

\[
\ddot{x}(t) + \frac{d}{m}\dot{x}(t) + \frac{k}{m}x(t) = \frac{1}{m}f(t) \quad (6)
\]

Using the phase variable representation,

\[
y_1 = x, \quad (7)
\]

\[
y_2 = \dot{x} = \dot{y}_1, \quad (8)
\]

\[
y_3 = \ddot{x} = \dot{y}_2 \quad (9)
\]

From the equation of the system,

\[
\ddot{x}(t) = \frac{1}{m}f(t) - \frac{d}{m}\dot{x} - \frac{k}{m}x(t) \quad (10)
\]

\[
y_2 = \frac{1}{m}f(t) - \frac{d}{m}y_2 - \frac{k}{m}y_1 \quad (11)
\]

And

\[
y_1 = y_2 \quad (12)
\]

While

\[
x = y_1 + 0y_2 \quad (13)
\]

The state variable can be expressed in matrix form as

\[
\begin{bmatrix}
\dot{y}_2 \\
\dot{y}_1 
\end{bmatrix}
= \begin{bmatrix}
\frac{d}{m} & -\frac{k}{m} \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
y_2 \\
y_1
\end{bmatrix}
+ \begin{bmatrix}
\frac{1}{m} \\
0
\end{bmatrix}f(t) \quad (14)
\]

\[
x = \begin{bmatrix}
0 & 1
\end{bmatrix}
\begin{bmatrix}
y_2 \\
y_1
\end{bmatrix} \quad (15)
\]

**MATLAB SIMULATION**

Modelling parameters
m=2kg  

k=3N/m  

d=5Ns/m  

Therefore:

\[-\frac{d}{m} = -\frac{5}{2} = -2.5\]

\[-\frac{k}{m} = -\frac{3}{2} = -1.5\]

\[1 \div \frac{1}{m} = \frac{1}{2} = 0.5\]

The state variable in matrix form becomes

\[
\begin{bmatrix}
y_2 \\
y_1
\end{bmatrix} = \begin{bmatrix}
-2.5 & -1.5 \\
1 & 0
\end{bmatrix} + \begin{bmatrix}
0.5 \\
0
\end{bmatrix} f(t)
\]

\[x = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} y_2 \\ y_1 \end{bmatrix}\]

A simple representation of the state phase variable is given in (Refer Figure 3)

The representation shown in Figure 3 can be designed with standard Simulink blocks from the MATLAB Simulink library. The Simulink programme allows for a plot of the step function of the system. Transforming the design in Figure 3 with the Simulink standard blocks gives a state phase variable block representation with Simulink standard blocks as show in (Refer Figure 4).

It is however possible to use a standard state phase variable block to represent the system in its simplest form and this is given in (Refer Figure 5).

The scope in Figure 6 can be used to obtain a plot of the step response of the state space representation of the system while the ‘SIMOUT’ block helps to write a vector sample of the output values and the time response. The vector result is written out as ‘simoutstep1’ for the step input response and a command on the command line “figure; plot (tout, simoutstep1); grid” helps to generate a plot of the values available in the workspace.

TRANSFER FUNCTION

In generating the step response or the unit impulse function response of a second order system, the transfer function must be computed.

Recall;

\[
\ddot{x}(t) + \frac{d}{m} \dot{x}(t) + \frac{k}{m} x(t) = \frac{1}{m} f(t)
\]

(16)

Recall also that using Laplace Transforms:

\[
\ddot{x}(t) = s^2 X(s), \dot{x}(t) = sX(s), x = X(s)
\]

(17)
Therefore,

The equation becomes;

\[ s^2 X(s) + \frac{d}{m} s X(s) + \frac{k}{m} X(s) = \frac{1}{m} F(s) \]

(18)

\[ \left( s^2 + \frac{d}{m} s + \frac{k}{m} \right) X(s) = \frac{1}{m} F(s) \]

(19)

\[ \frac{X(s)}{F(s)} = \frac{\frac{1}{m}}{s^2 + \frac{d}{m} s + \frac{k}{m}} \]

(20)

The transfer function can then be taken as the output divided by input i.e.

\[ G(s) = \frac{Y(s)}{R(s)} = \frac{\frac{1}{m}}{s^2 + \frac{d}{m} s + \frac{k}{m}} \]

(21)

But,

\[ \frac{d}{m} = 2.5 \]

\[ \frac{k}{m} = 1.5 \]

\[ \frac{1}{m} = 0.5 \]

Therefore,

\[ G(s) = \frac{Y(s)}{R(s)} = \frac{\frac{1}{m}}{s^2 + \frac{d}{m} s + \frac{k}{m}} = \frac{0.5}{s^2 + 2.5 s + 1.5} \]

The Simulink representation for the transfer function above can be done using a standard Simulink transfer function block in MATLAB as presented in Figure 6. The Simulink programme allows for a plot of the step function or the unit impulse response of the system. To show that the output of the transfer function model is identical to the output (t) of the phase variable model using the MATLAB functions, a transfer function block standard is used to simulate the transfer function and the results compared with those of the state space function. The block diagram representation of this design is given as Figure 6.

RESULTS AND DISCUSSION

The result obtained from the plot of the values from the SIMOUT response is given in (Refer Figure 7).

The impulse response of the system can also be modelled with Simulink. To obtain an impulse response, two step response blocks are employed as in show (Refer Figure 8).

The results shown in Figure 8 and Figure 9 are relatively opposite to each other. The concept of step response is such that the response starts from zero initially and then increases to a maximum value and then
remains the same for the rest of the duration. The response shown in Figure 8 has initial value of zero and a final value of 0.33. The nature of the curve shows that the system is overdamped. Thus, the damping ratio is greater than 1. For the response in Figure 9, it starts from zero, reaches a maximum value and then returns to zero where it remains constant again over time. The combination of the two responses was used to simulate the state space system represented by a state space standard Simulink block (Refer Figure 9).

The ‘SIMOUT’ block writes a vector sample of the output values and the time response to the MATLAB workspace. The vector result is written out as ‘simoutimpulse1’ for the unit impulse response of the system and a command on the command line “figure; plot (t, simoutimpulse1); grid” helps to generate a plot of the values available in the workspace. The result obtained from the plot of the values from the simoutimpulse1 response is given in Figure 10. The result obtained from the scope 2 is similar to that obtained by plotting the SIMOUT values. Figure 11 shows a brief result of the system. The graph in Figure 11 shows an overlap of two graphs which shows that the two methods can necessarily be compared to the Simulink design for the whole design. (Refer Fig. 10)

The simulation can be repeated for a unit impulse response, where the impulse response is simulated using two step responses and other parts of the Simulink representations remains the same. The transfer function is simulated by using a standard Simulink transfer function block in MATLAB as presented in Figure 12. The simulation setup is used to compare the output of the transfer function model with the output (t) of the phase variable model using the mux Simulink block. The result from the scope overlapped and that show that the results are identical. The block diagram representation of this design is given as (Refer Figure 11).

The result obtained from the scope 2 is similar to that obtained by plotting the SIMOUT values. Figure 12 shows a brief result of the system. The graph in Figure 12 shows an overlap of two graphs which shows that the two methods can necessarily be compared to the Simulink design for the whole design. (Refer Fig. 12)
To obtain the velocity profile and the acceleration profile from the Simulink simulation, probes were placed before the integrators as presented in the block representation (Figure 12). The simulation can be repeated for both unit step response while the impulse response is simulated using two step responses and other parts of the Simulink representations remain the same. The block diagram representation of this design for the step response and the unit impulse response are presented in Figure 13 and Figure 14 respectively. (Refer Fig.13)

For the step function system setup in Figure 14, scope 2 and scope 3 can give the $\dot{x}$ and $\ddot{x}$ responses respectively. Taking the results from the probes gives the responses in (Refer Figure 14 and Figure 15).

Comparing the step response from the acceleration feedback and absolute velocity it can be seen that the two responses take an impulse form, rising to a specific value and then returning back to rest. The acceleration response shows a rapid rise with a gradual deceleration while the velocity increased slowly and decreased slowly. (Refer Fig.16)

For the impulse function system setup in Figure 16, scope 2 and scope 3 can give the $\dot{x}$ and $\ddot{x}$ responses respectively. Taking the results from the probe gives the responses in (Refer Figure 17 and Figure 18).

Both the responses acceleration probe and velocity probe shown a rapid response typical of an impulse response. The time for settling of the system was very short in both instances. A ‘SIMOUT’ block function was included in the Simulink design to obtain the values of $\dot{x}(t)$ and $x(t)$, in a bid to plot of $\dot{x}(t)$ against $x(t)$. Figure 19 presents the Simulink block used for generating the various values needed on the workspace. (Refer Fig.19)

To obtain a plot of $\dot{x}(t)$ against$x(t)$, a figure function was used on the command line, the command was written as: figure; plot (displacement, x dot) grid. The result obtained from the command is given in (Refer Figure 20).

Figure 21 gives the transient response from the step function of the system. This describes qualitatively the dynamics of the system. A change in the system will cause its state to trails one of its trajectories which is present on the phase diagram. The variable $\dot{x}(t)$ is a representation of the systems feedback velocity. It is often used in automatic control systems for active
vibration control and isolation by attenuating at the natural frequency of the internal equipment.

**CONCLUSION**

Simulation is an important tool for engineering design, and analysis of complex engineering systems. With simulation, data computation is easy, accurate and faster. This paper reported the research work carried on mass spring damper model in phase variable form. The results shown that the system is overdamped. The graph in unit step response of state phase variable block representation of MSD compared with transfer function representation shows an overlap of two graphs. This implies that the two methods can necessarily be compared to the Simulink design for the whole design. Comparing the step response from the acceleration feedback and absolute velocity it can be seen that the two responses takes an impulse form, rising to a specific value and then returning back to rest. The acceleration response shows a rapid rise with a gradual deceleration while the velocity increased slowly and decreased slowly. Hence, a change in the system will cause its state to trails one of its trajectories which is present on the phase diagram.

**REFERENCES**

http://edu.levitas.net/Tutorials/Matlab/Simulink/examples.html


LIST OF FIGURES:

Figure 1. Mass Spring Damper Representation

Figure 2. Free Body Diagram of Mass Spring Damper

Figure 3. State Phase Variable Block Representation of MSD System
Figure 4. State Phase Variable Block Representation with Simulink Standard Blocks

Figure 5. State Phase Variable Block Representation with Simulink Standard Blocks for Step Response

Figure 6. Unit step response of State Phase Variable Block Representation of MSD System compared with Transfer Function Representation
Figure 7. Unit Step Response of State Phase Variable Block Representation of MSD System obtained from Simulink

Figure 8. State Phase Variable Block Representation with Simulink Standard Blocks for Impulse Response

Figure 9. Unit Impulse Response of State Phase Variable Block Representation of MSD system obtained from Simulink
**Figure 10.** Unit Step Response of State Phase Variable Block Representation of MSD compared with Transfer Function Representation

**Figure 11.** Unit Impulse Response of State Phase Variable Block Representation of MSD system compared with Transfer Function Representation

**Figure 12.** Compared Impulse Response of State Phase Variable and Transfer Function Representation
Figure 13. State Phase Variable Block Representation with Simulink standard blocks with velocity and Acceleration Probes.

Figure 14. Acceleration Step Response of the Transfer Function
Figure 15. Velocity Step Function of the Transfer Function

Figure 16. Unit Impulse State Phase Variable Block Representation with Simulink Standard Blocks with Velocity and Acceleration Probes

Figure 17. Acceleration Impulse Function of the Transfer Function
**Figure 18.** Velocity Impulse Function of the Transfer Function

**Figure 19.** Simulink Block to plot $\dot{x}(t)$ against $x(t)$

**Figure 20.** Graph of $\dot{x}(t)$ against $x(t)$