SPINE INTEGRATION FOR QUADRUPEDAL ROBOTS

Sukhreen Sandhu
Dr. Hao Xu

Department of Electrical and Biomedical Engineering
University of Nevada, Reno

OVERVIEW

Purpose:

- Objective: Investigating integration of a flexible spine into quadrupedal robots for enhanced adaptability and locomotion in challenging terrains.
- Inspiration: Drawing from biological systems to develop a jointed flexible body capable of absorbing external forces during locomotion.
- Benefits: Broadening range of movement while reducing noise on IMU sensors.

Methodology:

- Force and Position Analysis: Conducting analysis to evaluate impact of spine incorporation on force absorption and gallop dynamics.
- Modeling Body: Substituting traditionally rigid body with a compliant joint linking two masses through simulation and assessing motion dynamics on multiple setups.

Advancements:

- Balancing Flexibility and Rigidity: Achieving balance between flexibility and rigidity to promote additional handling without sacrificing control capability.
- Potential Applications: Exploring in space and rescue missions to navigate unstable and dangerous terrain.

INTRODUCTION

- To introduce an ideal force to various model configurations of compliant joint: series, parallel, rigid connection for force analysis.
- To simplify model for analysis of differences and responses between setups through ideal force sensor measurements.
- To simulate a gallop gait by modifying various sinusoidal force input parameters.
- To determine motion characteristics and displacement results as forces are applied through generating equations of motion.
- To determine more advanced configurations of model using preliminary results.

THEORY

Equations of motion

- The behaviors of the masses in the system were determined as the response to external forces and conditions in a spring-mass system.
- Spring/Damper Joint in Series:
  - $m_1 \cdot \ddot{x}_1 = -k(x_1) - c(\dot{x}_1) + F_1 + m_1 \cdot g$
  - $m_2 \cdot \ddot{x}_2 = -k(x_2) - c(\dot{x}_2) + m_2 \cdot g$
- Spring/Damper Joint in Parallel:
  - $m_1 \cdot \ddot{x}_1 = -k(x_1) - c \cdot (\dot{x}_1) + F_1 + m_1 \cdot g + k(x_3)$
  - $m_2 \cdot \ddot{x}_2 = -k(x_2) - c \cdot (\dot{x}_2) - k(x_3) + m_2 \cdot g$
- $k(x_1)$ and $k(x_2)$ are spring forces exerted on their respective masses and are calculated based on the displacement of each mass from its equilibrium position according to Hooke's law. $k(x_3)$ is the force exerted by the spring connecting mass 1 and 2 in parallel.
- x_1 , x_2 represent the displacements of m_1, m_2; k is the spring constant, c is the damping coefficient, F_1 is the external force, g is the acceleration due to gravity.

METHODOLOGY

Using Simscape's Mechanical Translational library, a translational system of the simplified quadruped was built with series, parallel, and no compliant joint configurations, (*See Figures 1-3.*)

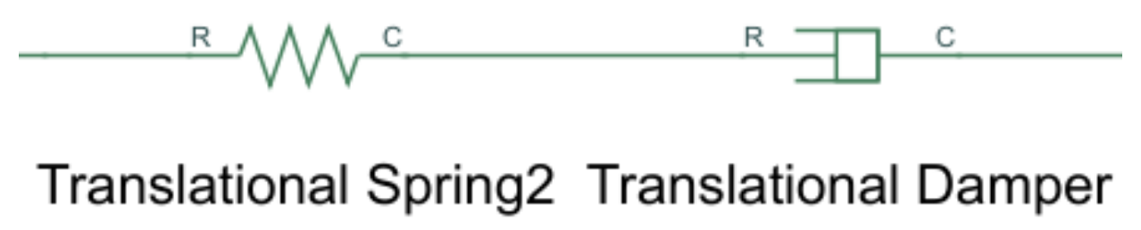


Figure 2. Series Configuration

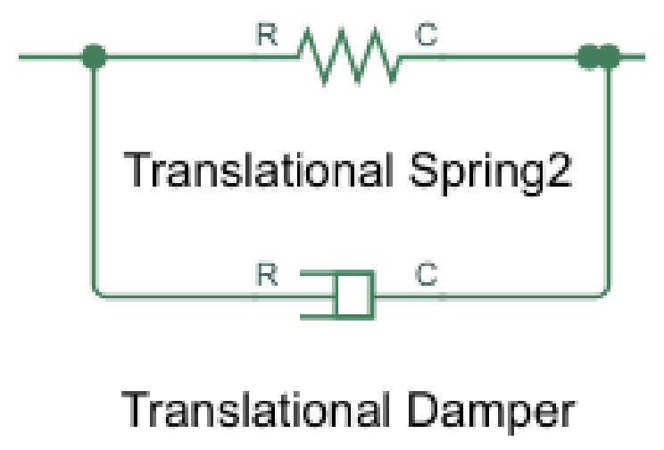


Figure 3. Parallel Configuration

The initial values for m_1, m_2 were set to 6 kg each to represent the total Unitree robot's 12 kg body weight. k and c were kept at MATLAB's default configurations of 1000 N/m and 100 $N \cdot s/m$, respectively, and g was set to 9.81 $\frac{m}{s^2}$.

Each configuration was tested at a base rate of 1 for the amplitude and 1 rad/sec for the frequency values of the sinusoidal wave. This was compared against changing the values to 10 for the amplitude and 31.4 rad/sec for the frequency to mimic a gallop gait force input. The scope measured the force sensor values between the two masses.

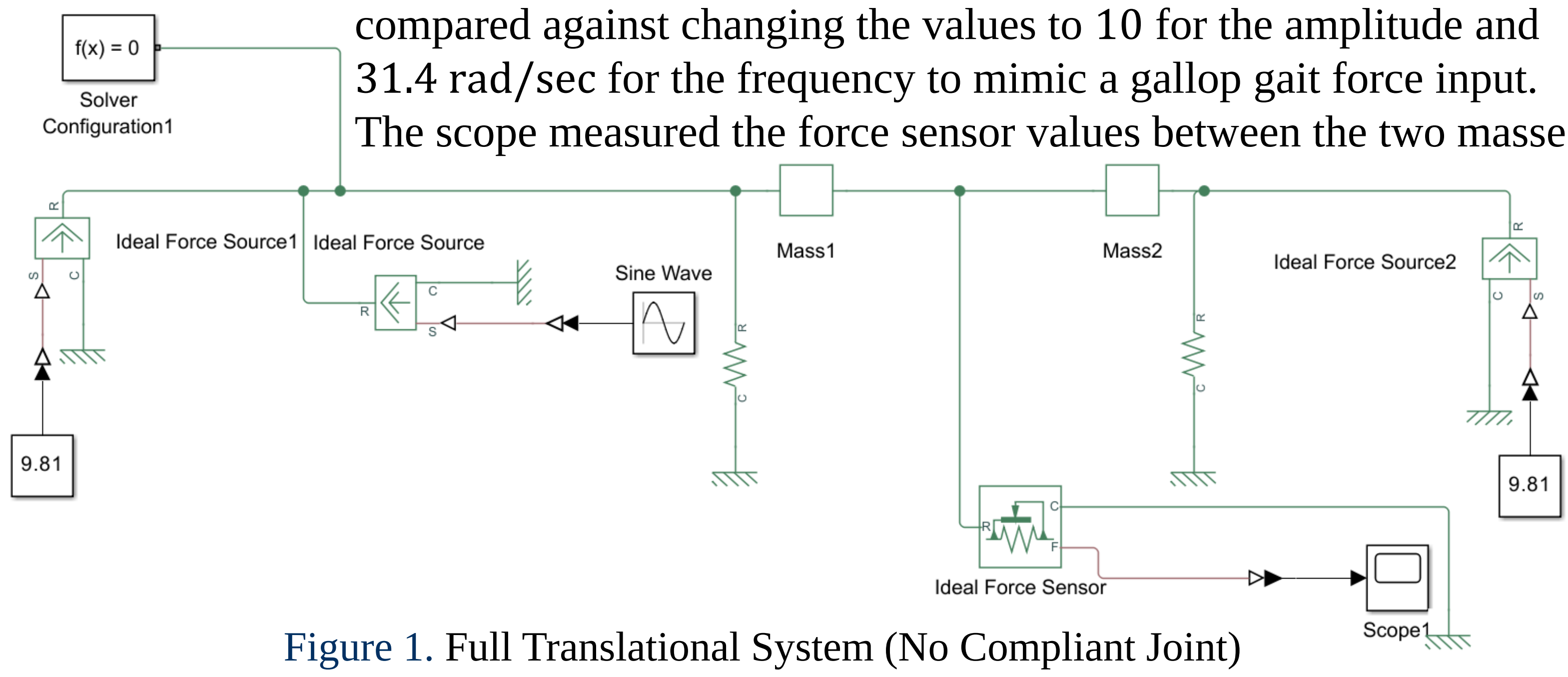
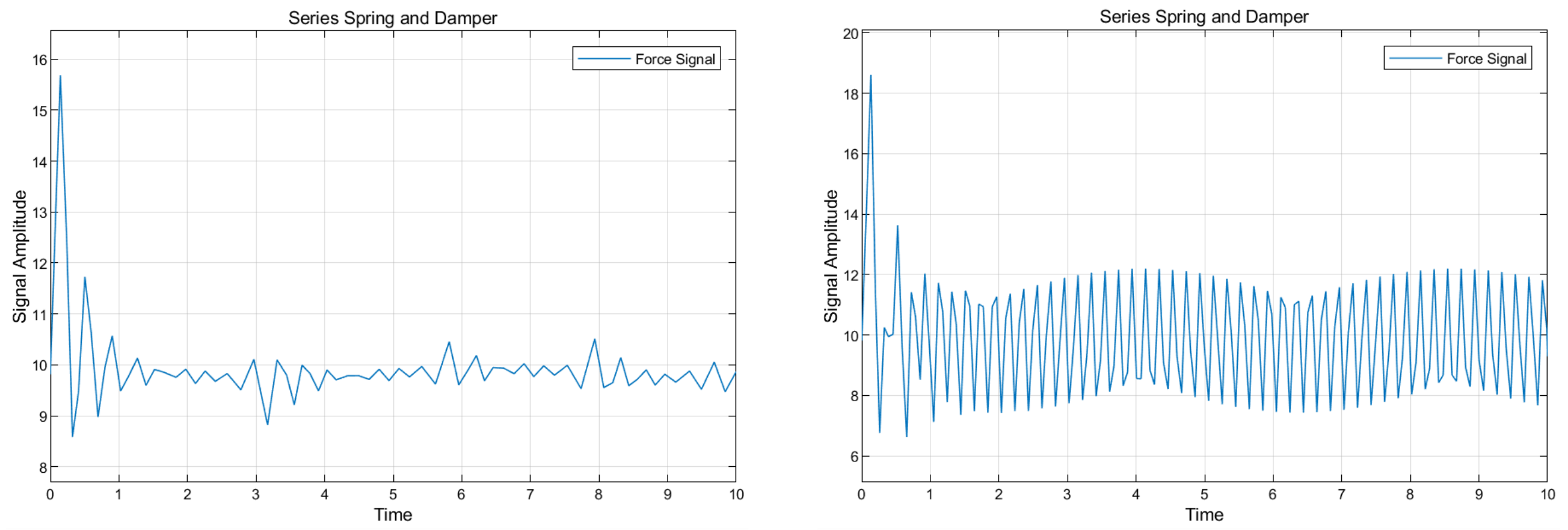
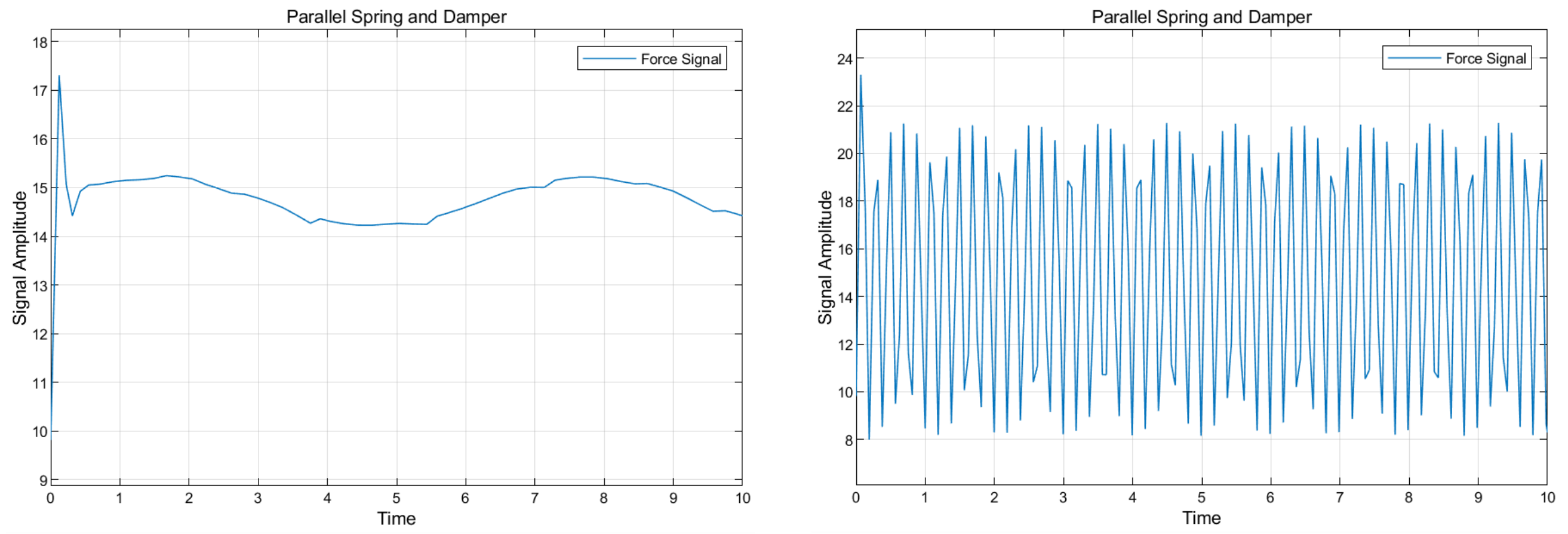


Figure 1. Full Translational System (No Compliant Joint)

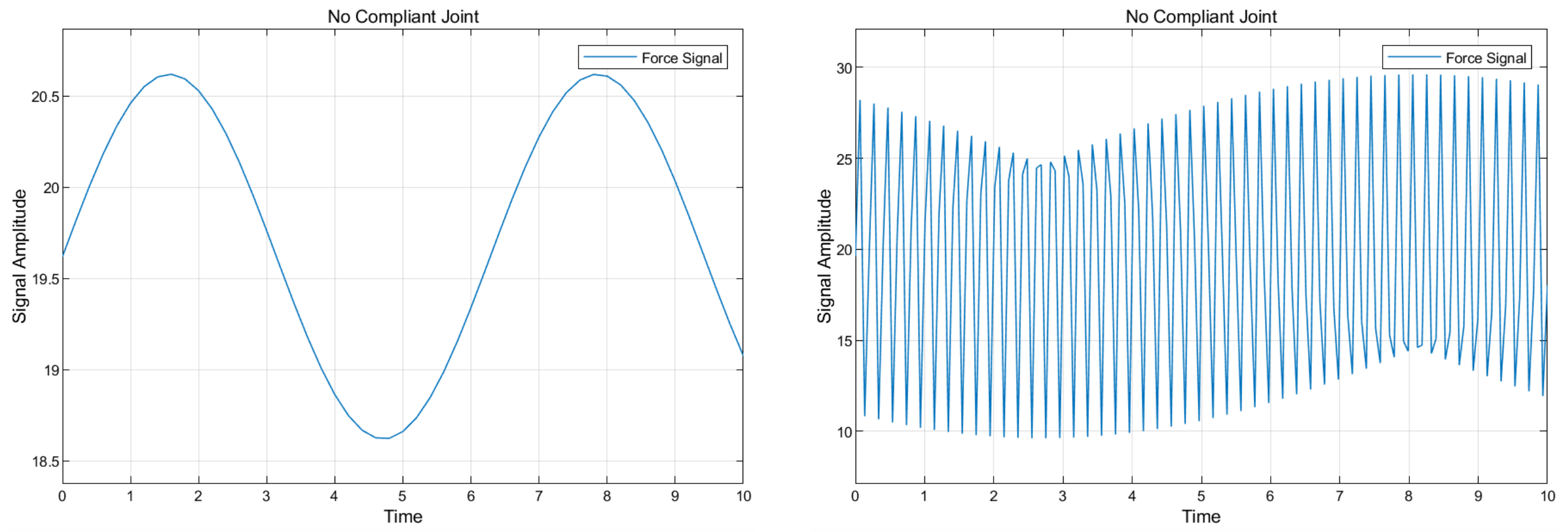
RESULTS



Figures 4-5. Series Spring and Damper: Base Configuration, Gallop Configuration



Figures 6-7. Parallel Spring and Damper: Base Configuration, Gallop Configuration



Figures 8-9. No Compliant Joint: Base Configuration, Gallop Configuration

CONCLUSION

Table 1. Base Sinusoidal Data

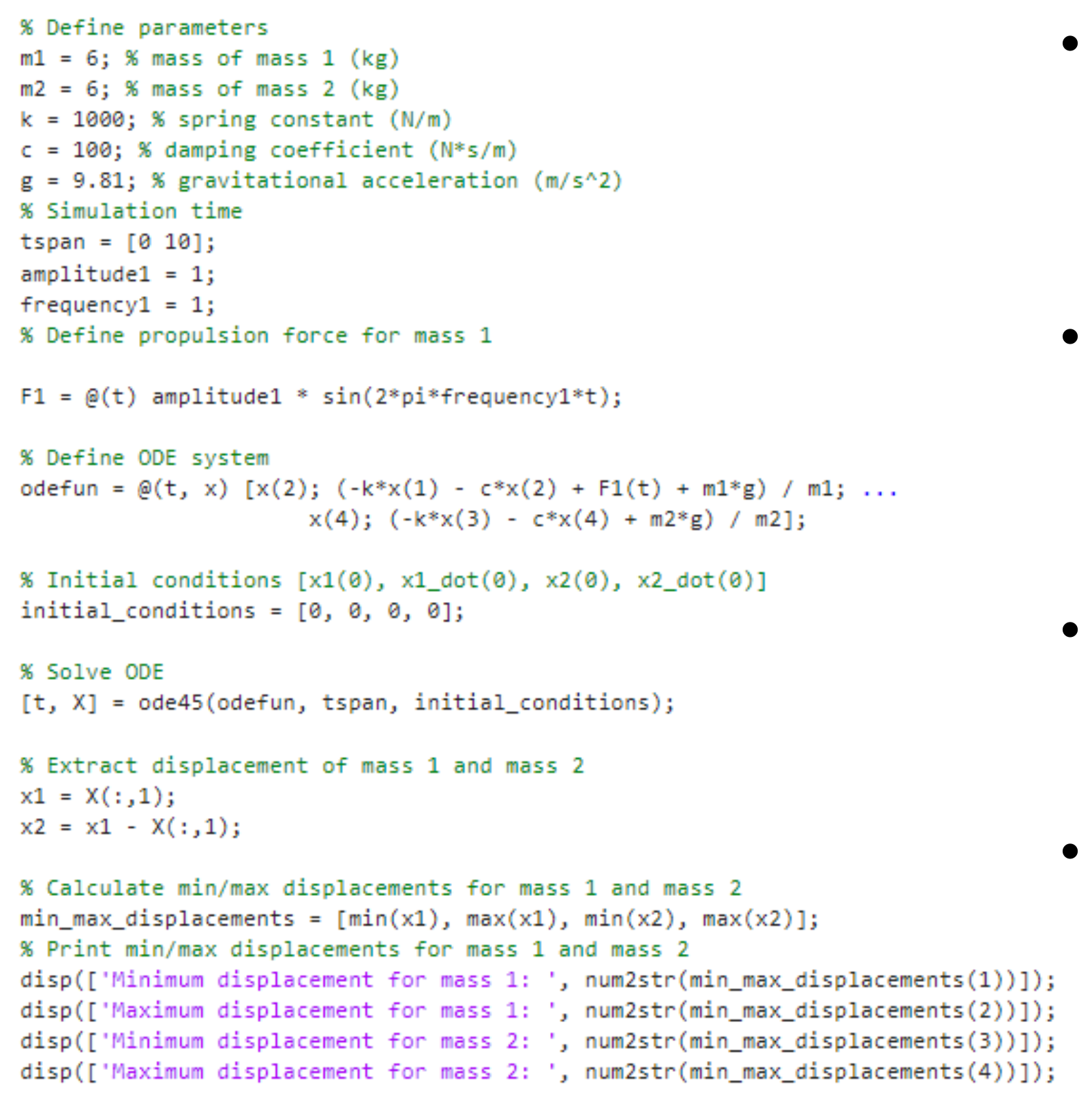
Base Case (Amp = 1, *F* = 1)	**Series Configuration**	**Parallel Configuration**	**No Joint Configuration**
Peak force amplitude	7.104 units	7.496 units	19.96 units
Mean force value	9.976 units	14.74 units	19.79 units
Median force value	9.820 units	14.88 units	19.93 units

Table 2. Gallop Sinusoidal Data

Gallop Gait (Amp = 10, *F* =31.4)	**Series Configuration**	**Parallel Configuration**	**No Joint Configuration**
Peak force amplitude	11.99 units	15.31 units	20 units
Mean force value	9.91 units	14.68 units	19.6 units
Median force value	9.89 units	14.68 units	19.92 units

- Both series and parallel configurations exhibit lower mean/median force values compared to the no-joint configuration (*see Tables 1-2*).
- Lower force values indicate absorption of some force by the compliant joint.
- Peak force values vary across inputs, with smaller values suggesting dissipation of force resulting in smaller amplitudes.
- Larger variations between series/parallel states and no-joint data suggest greater resistance to deformation within the no-joint configurations due to lack of flexibility.

FUTURE WORK



```
% Define parameters
m1 = 6; % mass of mass 1 (kg)
m2 = 6; % mass of mass 2 (kg)
k = 1000; % spring constant (N/m)
c = 100; % damping coefficient (N*s/m)
g = 9.81; % gravitational acceleration (m/s^2)
% Simulation time
tspan = [0 10];
amplitude1 = 1;
frequency1 = 1;
% Define propulsion force for mass 1

F1 = @(t) amplitude1 * sin(2*pi*frequency1*t);

% Define ODE system
odefun = @(t, x) [x(2); (-k*x(1) - c*x(2) + F1(t) + m1*g) / m1; ...
                  x(4); (-k*x(3) - c*x(4) + m2*g) / m2];

% Initial conditions [x1(0), x1_dot(0), x2(0), x2_dot(0)]
initial_conditions = [0, 0, 0, 0];

% Solve ODE
[t, X] = ode45(odefun, tspan, initial_conditions);

% Extract displacement of mass 1 and mass 2
x1 = X(:,1);
x2 = x1 - X(:,1);

% Calculate min/max displacements for mass 1 and mass 2
min_max_displacements = [min(x1), max(x1), min(x2), max(x2)];
% Print min/max displacements for mass 1 and mass 2
disp(['Minimum displacement for mass 1: ', num2str(min_max_displacements(1))]);
disp(['Maximum displacement for mass 1: ', num2str(min_max_displacements(2))]);
disp(['Minimum displacement for mass 2: ', num2str(min_max_displacements(3))]);
disp(['Maximum displacement for mass 2: ', num2str(min_max_displacements(4))]);
```

Figure 10. Displacement Simulation Progress

- Analyze displacements of complex systems for better understanding of forces acting on the system. Figure 10 demonstrates working progress.
- Implement a forced feedback control system using reinforcement learning to adjust joint angles based on the applied forces.
- Enhance gait simulations by considering other parameters such as sinusoidal phase, leg motion, and ground interaction.
- Considering more complex dynamics, the future model will look closer to Figure 11.

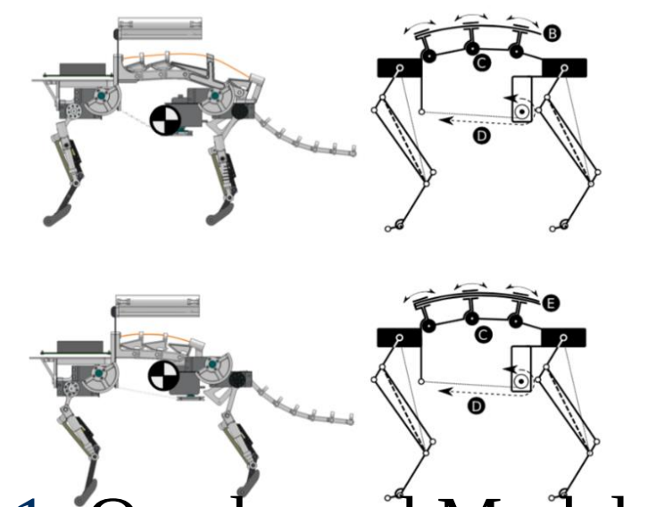


Figure 11. Quadruped Model

ACKNOWLEDGEMENTS

This material is based on work supported by the NVSGC program under Grant No: 80NSSC20M0043. research is supported by National Science Foundation REU grant EEC #1852578. This research was done under Dr. Hao Xu at the University of Nevada, Reno at the Autonomous Systems Laboratory. Figure 11 obtained from: (Eckert, Peter & Badri-Spröwitz, Alexander & Witte, Hartmut & Ijspeert, A.J.. (2014). Comparing the effect of different spine and leg designs for a small, bounding quadruped robot.)