Spring-Mass Walker Model: Exploring Dynamics and Energy Conservation

Assignment 6
Name: Nicolas Helio Cunha Nabrink

Submission Date: 28.05.2024
required time (hh:mm): 10:00

Problem statement

Bipedal walking is a complex biomechanical process that can be effectively studied using the spring-mass walker model. This model simplifies the dynamics of walking by representing legs as massless springs and the body as a point mass, capturing the essential principles of locomotion such as energy efficiency and stability.

The task involves developing a Matlab/Simulink model of the spring-mass walker. This requires deriving the equations of motion (EoMs) and programming the system to simulate various gait phases, including single and double support, as well as take-off (TO) and touch-down (TD) conditions. A key part of this challenge is determining the angle of attack and ensuring the model handles potential failure modes to maintain robust simulations.

Additionally, the assignment includes analyzing the spring forces and their patterns, and reporting the model's forward speed, both maximum instantaneous and average. Given parameters like spring constant (k), mass (m), and natural length of the spring (l0), the task aims to propose a strategy for achieving continuous walking patterns.

This essay will explore the development and implementation of the spring-mass walker model in Matlab/Simulink, focusing on the specific challenges of simulating bipedal gait and providing insights into practical solutions for modeling bipedal locomotion. Failure mode and continuous pattern identification are also covered.

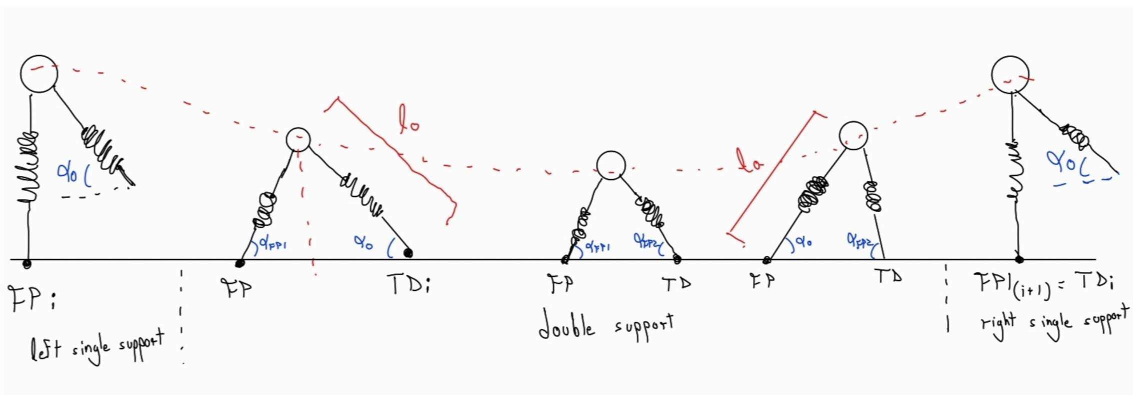

Figure 1: This diagram illustrates the spring-mass walker consisting of two springs (representing legs) attached to a central mass (representing the body). The model transitions between single support, where one leg contacts the ground at the FP (flying phase) point, and double support, where the second leg touches down at the TD (touch down) point.

In the developed spring-mass walker model, the system consists of two springs attached to a central mass, representing the legs and the body, respectively. The model transitions between two primary gait phases: single support and double support. During the single support phase, one leg is in contact with the ground at the point labeled FP (flying phase), while the other leg is in the air. As the walker progresses, the second leg makes contact with the ground at the point TD (touch down), initiating the double support phase. Subsequently, the first leg loses contact with the ground, and the previous TD point becomes the new FP. This cyclical process mimics the alternating support phases seen in natural bipedal walking, ensuring continuous motion and stability in the model.

Definitions

Table 1 gives an overview to the most important symbols, constants and variables
For all following calculations we assume that:

- Drag (resistance of air) is neglected
- The coordinate system is located at the bottom of figure 2 with the y-axis pointing up
- Body-movement is just considered in the 2D plane

Symbol	Property	Value	Unit
g	Gravitational acceleration	9.81	m/s^2
k	Spring stiffness	10	kN/m
l	Spring length	-	m
l_0	Spring resting length	1	m
m	Mass of the body	80	kg
y	Vertical position of the body	-	m
$\dot{y}$	Vertical velocity of the body	-	m/s
x	Horizontal position of the body	-	m
$\dot{x}$	Horizontal velocity of the body	-	m/s
y_0	Simulation initial vertical position	0.97	m
$\dot{y}_0$	Simulation initial vertical velocity	0	m/s
x_0	Simulation initial horizontal position	0	m
$\dot{x}_0$	Simulation initial horizontal velocity	1.05	m/s
e_p	Gravitational potential energy	-	J
e_s	Elastic potential energy	-	J
e_k	Kinetic energy	-	J
F_s	Spring force	-	N
t	Time	-	s
α	Angle of the leg (spring) in respect to the floor	-	°
α_0	Initial angle suspended leg	68	°
$\dot{x}_{max}$	Simulation maximum horizontal velocity	1.1496	m/s
$\dot{x}_{mean}$	Simulation mean horizontal velocity	1.0944	m/s

Table 1: Definitions of Symbols and Properties

1 Approach and Implementation

The spring-mass walker model was implemented entirely using Matlab. The process involved several key steps to ensure accurate simulation and analysis of bipedal walking dynamics.

1. **Model Development and Equations of Motion**:
   - Developed the physical model of the spring-mass walker.
   - Derived the equations of motion (EoMs) while respecting the problem's constraints.
2. **Function Implementation**:
   - Implemented each function in Matlab to handle different aspects of the simulation, such as force calculations and kinematic updates.
3. **Numerical Solution of ODEs**:

- Developed a Runge-Kutta 4 algorithm to solve the ordinary differential equations (ODEs) derived from the model.

4. **Cycle Implementation**:

   - Used the position y of the body to change from the different ODEs for each cycle. If $y > y_0 sin(\alpha_0)$ than the ODE used is the single support, otherwise the double support ODE is used. The "if" condition is defined inside the function where the ODE is stated.

   - Used the cyclic process described earlier to update the FP (flying phase) and TD (touch down) points, ensuring a continuous walking pattern.

5. **Data Visualization**:

   - Plotted the x and y positions of the mass.
   - Analyzed the horizontal and vertical forces exerted by each leg.
   - Calculated and plotted the total energy of the system.
   - Created a phase diagram to illustrate the walker's state transitions.
   - Tracked and plotted the forward velocity to evaluate the model's performance.

Failure Mode

To address the task of identifying failure modes and implementing safeguards for aborting the simulation, a conditional statement within the simulation framework was employed, as shown in figure **??**. Specifically, if the vertical position (y) of the walker exceeded the natural length of the springs (l_0), indicating that both legs had lost contact with the ground, the simulation was promptly halted. This condition, triggered by $y > l_0$, served as a crucial safety measure to prevent the simulation from proceeding in scenarios where the walker's initial conditions or parameters might lead to unrealistic or unstable states. Upon encountering this condition, an error message was displayed, alerting the user to the issue and prompting further investigation or adjustments to the simulation setup. By proactively identifying and addressing potential failure modes, such as excessive initial velocities (dx_0), this approach ensured the robustness and reliability of the simulation framework.

Equation of Motion

In this section, we derive the equations of motion for the spring-mass walker model. The derivation involves applying Newton's second law to the mass and considering the forces exerted by the springs during different gait phases. These equations will form the basis for simulating the walker's dynamics, capturing the transitions between single and double support phases, and ensuring the realistic representation of bipedal locomotion.

First, we need to define some important functions and equations for the implementation, the following auxiliary functions can be used for calculating the values in respect to FP and TD points/attached leg.

$$l(x, y, FP) = \sqrt{(x - FP)^2 + y^2} \tag{1}$$

$$F_s(l) = k(l_0 - l) \tag{2}$$

$$cos(\alpha_{FP}) = \frac{FP - x}{l} \tag{3}$$

$$sin(\alpha_{FP}) = \frac{y}{l} \tag{4}$$

$$\tag{5}$$

It is important to note that the points FP and TD have each a attached leg in this model, therefore the equations can be used both for FP and TD. However, the model expects that at least one leg will be in touch with the ground.

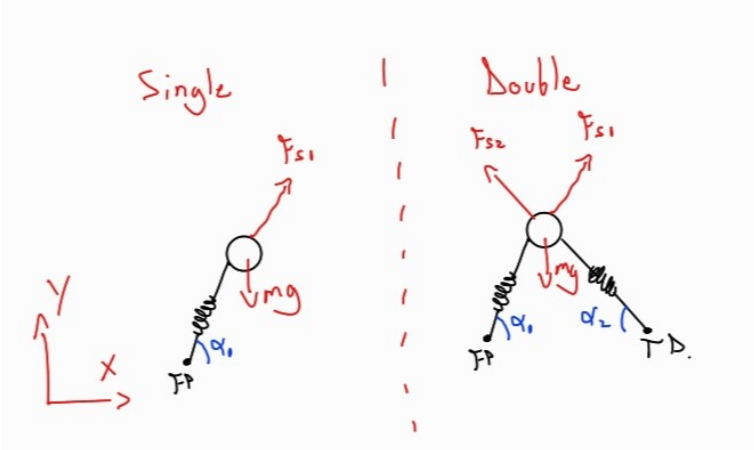

Figure 2: This diagram illustrates the free body diagram for both the single and double support phases of the spring-mass walker model. It depicts the forces acting on the mass during each phase, including the gravitational force and spring forces providing insight into the dynamic behavior of the system during bipedal locomotion.

The EoM can then be derived for the single support cycle and double support cycle together, with the difference that for the double support another term for the second leg is added, in red, in all equations.

$$m\ddot{x} = -F_{s1}cos(\alpha_1) - F_{s2}cos(\alpha_2) \tag{6}$$

$$\ddot{x} = -\frac{1}{m}F_{s1}cos(\alpha_1) - \frac{1}{m}F_{s2}cos(\alpha_2) \tag{7}$$

$$m\ddot{y} = -g + F_{s1}sin(\alpha_1) + F_{s2}cos(\alpha_2) \tag{8}$$

$$\ddot{y} = -g + \frac{1}{m}F_{s1}cos(\alpha_1) + \frac{1}{m}F_{s2}cos(\alpha_2) \tag{9}$$

$$\tag{10}$$

When adding the spring force term to the equations, the signals of the terms are defined in respect to the functions $cos(\alpha)$ and $sin(\alpha)$ previously defined. For example, even if $\alpha 1$ in respect to FP and $\alpha 2$ in respect to TD, have the same value, their cosine will have opposite signals, due to the definition of the angles in figure 2.

Finally, for implementing the cycles we implement the following functions:

$$FP_{i+1} = TD_i \tag{11}$$

$$TD_i = x_i + l_0 cos(\alpha_0) \tag{12}$$

$$\tag{13}$$

The condition for changing between the cycles is, again, if $y > y_0 sin(\alpha_0)$ the phase is single support, otherwise the system is in double support phase.

In the quest for accurate simulation of the spring-mass walker model, we turn to numerical methods like the Runge-Kutta 4 (RK4) algorithm. It was used a implementation of RK4 to solve the model's equations of motion. By employing RK4, we iteratively compute the walker's trajectory and kinematics over discrete time intervals, offering a robust and efficient approach to numerical integration.

The RK4 method calculates the solution y at time $t + h$ based on the solution y at time t, using four intermediate evaluations of the function $f(t, y)$ within the time step h.

$$
\begin{aligned}
k_1 &= h \cdot f(t, y) \\
k_2 &= h \cdot f(t + \frac{h}{2}, y + \frac{k_1}{2}) \\
k_3 &= h \cdot f(t + \frac{h}{2}, y + \frac{k_2}{2}) \\
k_4 &= h \cdot f(t + h, y + k_3) \\
y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

Results

In the culmination of the efforts, the results obtained from simulating the spring-mass walker model are presented. Utilizing Matlab, simulations were conducted over a time horizon of 2.5 seconds, with time steps of 10^{-5} seconds. Valuable insights were gained into the dynamic behavior of the walker across various gait phases and terrain conditions. In this section, the outcomes of the simulations are showcased, offering a comprehensive examination of the model's performance and dynamics.

The xy position plot serves as a foundational visualization in understanding the locomotion dynamics of the spring-mass walker model. Figure 3 plot showcases the trajectory of the walker's center of mass in the horizontal (x) and vertical (y) planes over the simulated time period.

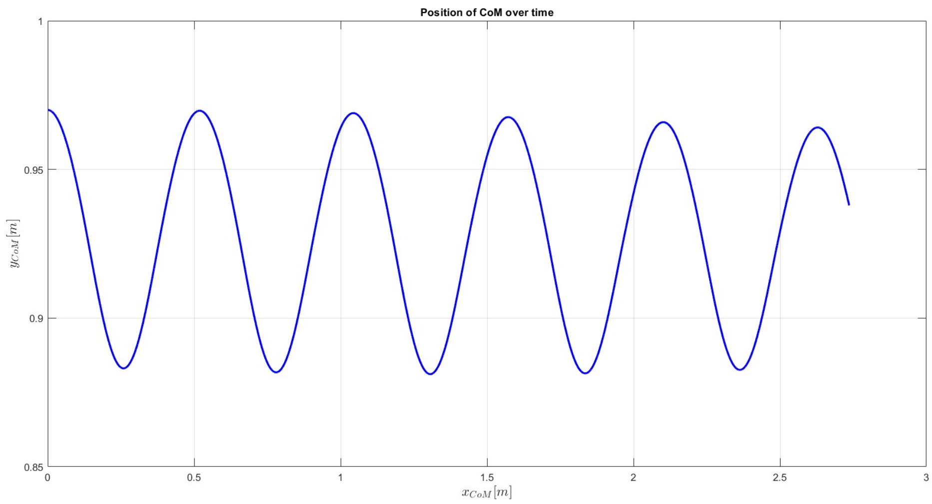

Figure 3: This plot illustrates the trajectory of the spring-mass walker's center of mass in the horizontal (x) and vertical (y) planes over the simulated time period of 2.5 seconds.

A diagram depicting the horizontal and vertical forces exerted by each spring has been included in the analysis, figures 4 and 5. This visualization provides insight into the dynamic forces acting on the spring-mass walker model, aiding in the understanding of its mechanical behavior during locomotion.

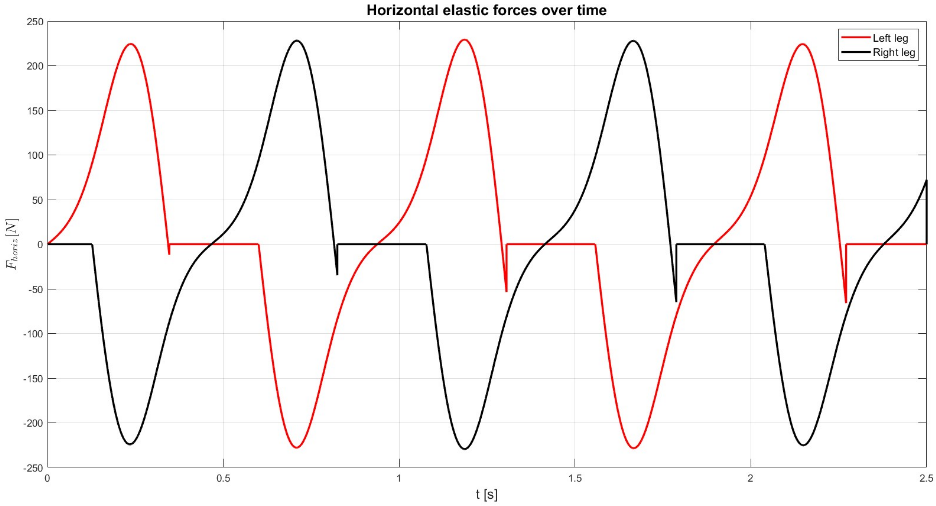

Figure 4: This diagram illustrates the horizontal forces exerted by each spring of the spring-mass walker model. The plot offers a visual representation of the lateral forces acting on the system, aiding in the analysis of its dynamic motion and stability during locomotion.

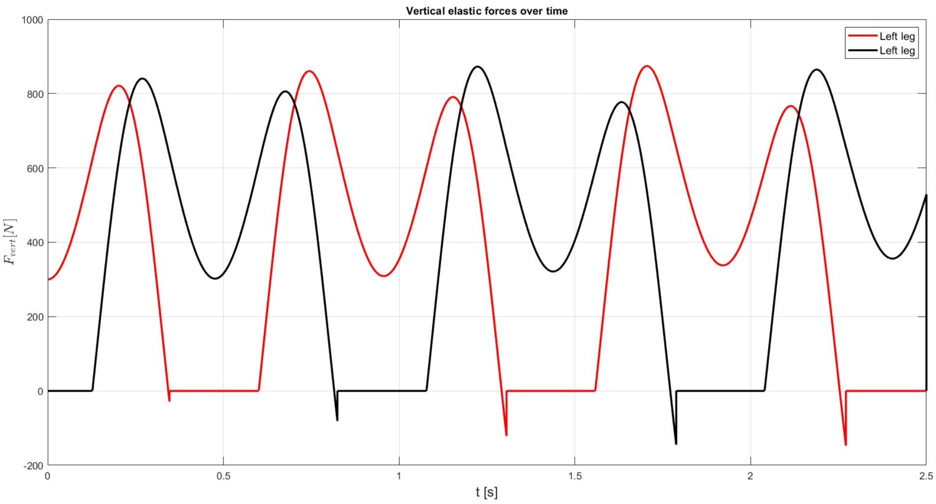

Figure 5: This diagram displays the vertical forces exerted by each spring of the spring-mass walker model. The plot provides insight into the vertical dynamics of the system, highlighting the forces involved in supporting the walker's weight and facilitating its movement.

The phase plot, figure 6, depicting state-space of the dynamics of the spring-mass walker model, offers a comprehensive visualization of the system's behavior over time. By analyzing the phase plot, we can discern the continuous gait of the robot, given initial states y_0.

Continuous pattern identification

The phase plot serves as a crucial tool for analyzing continuous walking patterns. By observing the trajectory of the walker in the phase space (e.g., velocity vs. position), we can discern stable and repetitive patterns indicative of continuous locomotion. Specifically, we can look for closed loops or limit cycles in the phase plot, suggesting that the walker's motion repeats over time. Variations in the shape and size of these loops provide insights into the stability and efficiency of the walking pattern. Therefore, analyzing the phase plot can offer a visual confirmation of continuous walking behaviors, given a set of parameters k,m and l_0, guiding parameter adjustments to achieve desired locomotion dynamics. A rough approach involves systematically varying these parameters within a defined range and checking the solution with the phase plot.

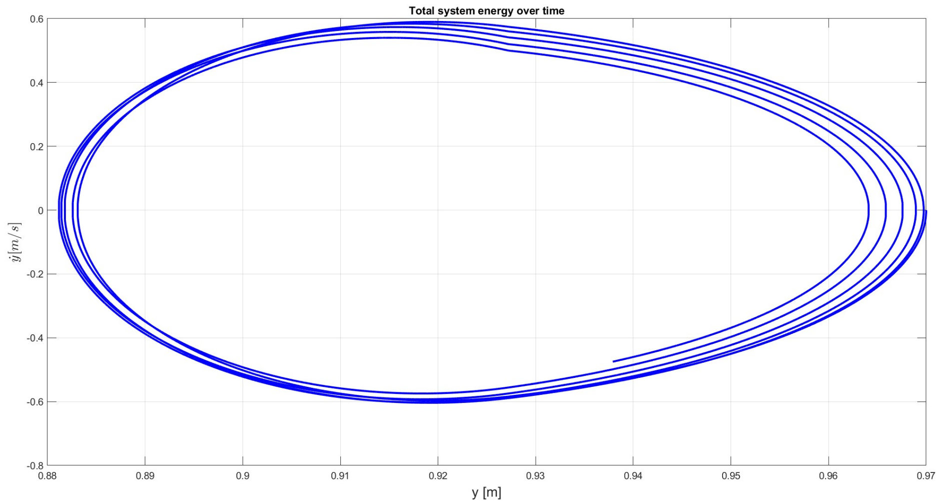

Figure 6: Phase plot or state-space of the body. The spiral form shows the continuous gait path.

It is also shown, in figure 7, the forward velocity of the model and its mean. The maximum instantaneous forward velocity is $1.1496 m/s$ and it's mean is $1.0944 m/s$.

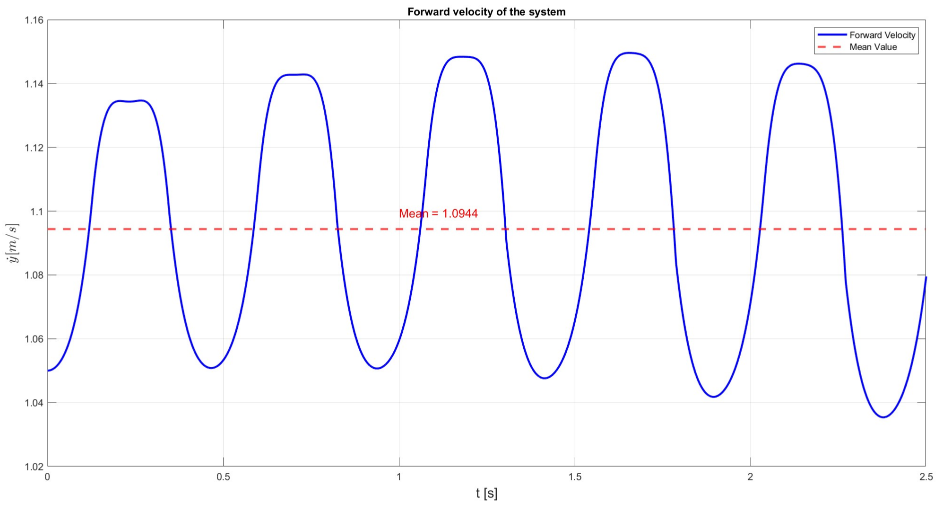

Figure 7: This diagram shows the forward speed of the body and it's mean across time.

Finally, it is analysed the total energy of the system, shown in figure 8. The initial energy of the system is 809.8560 J and the final energy decays to 805.9606 J.

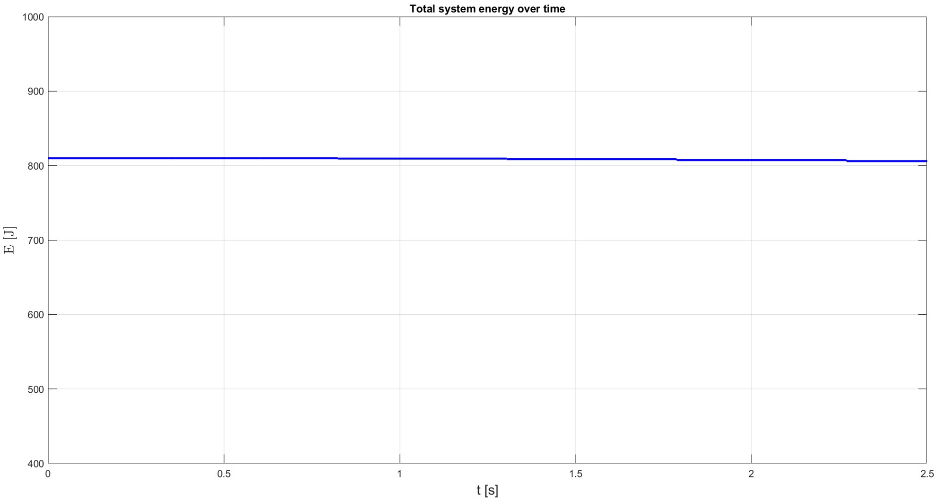

Figure 8: Total energy of the system. Initial value: 809.8560 J. Final value: 805.9606 J.

Discussion

In the discussion of the simulation results, the analysis of forces reveals that the transition between single support and double support phases of the spring-mass walker model is generally accurate. The forces indicate correct transitions between these phases, validating the fidelity of the simulation in capturing fundamental aspects of bipedal locomotion. However, a small error is observed in the first point after the transition, resulting in overlayers. This minor discrepancy may stem from limitations in the algorithm used to detect phase transitions, suggesting areas for improvement in future iterations of the model.

Moreover, the analysis of position data shows that the simulated trajectory of the walker aligns roughly with expected behaviors of the system, particularly within shorter time horizons. However, for larger time horizons, noticeable decay is observed in the maximum vertical position and other variables. This decay may indicate instabilities or inconsistencies in the simulation setup, potentially leading to inaccuracies in long-term predictions of the walker's motion. Further investigation and refinement of the model parameters and numerical methods are warranted to address these discrepancies and improve the accuracy and reliability of long-term simulations.

Additionally, the analysis of energy dynamics highlights significant decay in the system's total energy over the course of the simulation. Starting with an initial energy of 809.8560 J, the system experiences a considerable decay to 805.9606 J. This decay is not expected since it is a conservative system and it can be attributed to inconsistencies in the algorithm used to change phases. Addressing these inconsistencies and refining the phase transition algorithm may help mitigate those problems.

AI usage declaration

tools used

ChatGPT

used prompts

1-I have to write an essay based on the following problem: (problem copied from moodle) The first part of my essay is problem statement and introduction, please write that for me.

2-I implemented the solution following the steps: (...) Write a text defining the approach and implementation for my essay based on those steps

3- My results show the following conclusions: (...) write it in a paragraph form

4-Now I need you to write a discussion for that essay, I want you to note the following points in it: (...)

comments

It was very useful to write paragraphs when given bullet points of what I wanted to include