Title: Revolutionizing Biomechanics: Integrating AI, Collaborative Coding Environments, and Simulation Software for Enhanced Model Development and Analysis

Abstract **1**
Introduction **2**
Methods **2**
  Examples 3
    A. Example 1: Increasing muscle strength in OpenSim models 4
    B. Example 2: Computing the Center of Mass using OpenSim 5
    C. Example 3: Extracting Model Details in OpenSim 6
Discussion **7**
Potential limitations and challenges **8**
Conclusion **8**

Author:
Hossein Mokhtarzadeh
Email: mokhtarzadeh dot hossein at gmail dot com
The notebook will be soon shared on GitHub at hmok (Hossein Mokhtarzadeh) · GitHub.

Abstract

Biomechanics is a crucial field that investigates human movement and performance. This paper presents a comprehensive framework that combines ChatGPT, Colab, and OpenSim to streamline the creation and analysis of biomechanical models. By integrating ChatGPT and Colab, the framework offers a user-friendly and accessible approach to model development, while OpenSim provides a versatile platform for simulations. The framework's advantages include intuitive model development, online accessibility, and simplified setup. However, considerations such as the accuracy of ChatGPT responses and data security in Colab should be addressed. The paper demonstrates the framework's potential through practical examples, including muscle strength enhancement, center of mass determination, and extracting model details, showcasing its efficacy in advancing biomechanical research.

Keywords: Biomechanical modeling, ChatGPT, Colab, OpenSim, Integration

Introduction

Biomechanics plays a crucial role in understanding and optimizing human body mechanics and performance. This paper presents a novel framework that revolutionizes biomechanical modeling by leveraging the combined potential of ChatGPT, Colab, and OpenSim [1,2,3]. Our framework offers a unique approach by harnessing ChatGPT's capabilities to simplify the modeling process, while Colab provides a convenient platform for executing and debugging the generated models. Through seamless integration with OpenSim, a powerful open-source software, researchers gain access to a wide range of biomechanical analyses. This framework offers significant advantages, including user-friendliness, online accessibility, and the elimination of local installations. However, it is important to acknowledge potential limitations, such as reliance on ChatGPT's responses and security considerations when using clinical data in Colab.

Methods

Our proposed framework streamlines the biomechanical modeling process by combining the power of ChatGPT, Colab, and OpenSim. ChatGPT acts as an innovative tool, providing intuitive prompts and guidance for generating biomechanical models. Colab serves as an ideal platform, offering online accessibility without the need for local installations, for executing and debugging these models. Integration of OpenSim, an open-source software, is seamlessly achieved through Anaconda within Colab, enabling researchers to leverage its powerful features for biomechanical analyses [4].

While OpenSim has diverse applications in biomechanics, including gait analyses, rehabilitation, brain injuries, and sport injuries, our focus is on a few practical examples for training purposes. These examples include muscle strength enhancement, center of mass determination, and detailed biomechanical model simulations.

The combination of ChatGPT, Colab, and OpenSim within our framework offers notable advantages, such as user-friendliness and efficient model iteration. However, it is important to acknowledge potential limitations, such as the reliance on ChatGPT's responses and the need to handle clinical data carefully for security reasons (Figure 1).

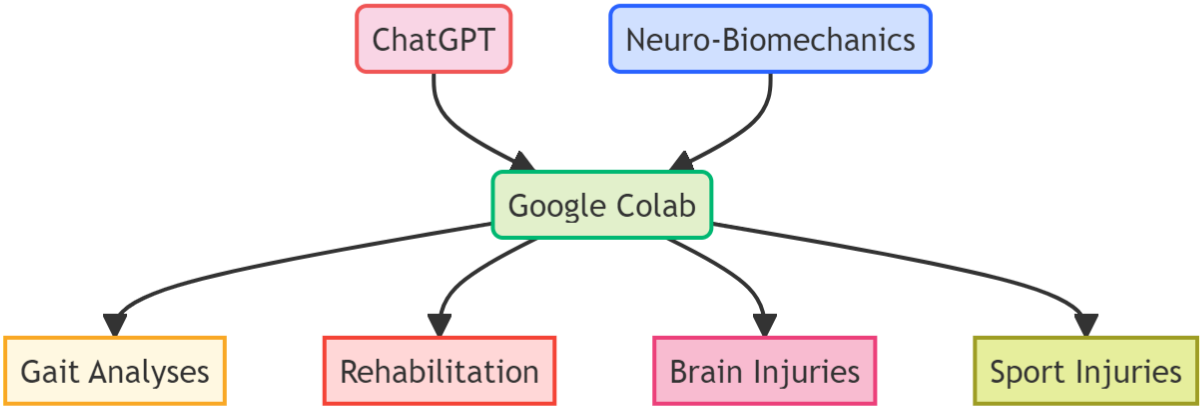

Figure 1 illustrates the integration of ChatGPT and Google Colab in neuro-biomechanics research. ChatGPT (pink node) facilitates model generation through prompts and guidance, while Google Colab (green node) provides a platform for executing and debugging the models. The framework enables various biomechanical analyses, such as gait analyses, rehabilitation, brain injuries, and sport injuries (connected nodes). The different colors and styles represent distinct domains within neuro-biomechanics, highlighting its multidisciplinary nature. OpenSim, a free software, was used for neuro-biomechanic modeling. Thanks to mermaid extension in Google docs by Renan LE CARO. For more information, please refer to the appendix.

Examples

While you can ask ChatGPT for assistance with OpenSim installation on Colab, here's a simplified approach:

First, install the condacolab package to enable conda usage in Colab. Run the following commands in https://colab.research.google.com/ in a Code cell:

```
!pip install -q condacolab
import condacolab
condacolab.install()
```

This will install the necessary packages for conda in Colab.

Next, you can install the OpenSim package using the official conda package from the opensim-org channel. Execute the following command:

```
!conda install -c opensim-org opensim
```

By following these steps, you'll be able to install OpenSim in Colab and proceed with your biomechanical modeling tasks.

A. Example 1: Increasing muscle strength in OpenSim models

To enhance and increase muscle strength in biomechanical models, the proposed framework incorporates OpenSim, which allows for the design of interventions. Researchers can utilize OpenSim to explore the impact of different training protocols on muscle performance.

To obtain a Python code snippet specifically for increasing muscle strength in OpenSim models using the OpenSim API, you can use the following prompt:

"Could you please provide a Python code snippet that demonstrates how to increase muscle strength in OpenSim using the OpenSim API?"

By using this prompt, you can request the ChatGPT model to generate the desired Python code for increasing muscle strength in OpenSim models.

Python Code Snippet:
Here is an example Python code that demonstrates how to increase muscle strength in OpenSim using the OpenSim API:

```
import opensim as osim

# Load an OpenSim model
model = osim.Model("path/to/your/model.osim")

# Access the muscles in the model
muscles = model.getMuscles()

# Define the scaling factor for increasing muscle strength
scaling_factor = 1.2

# Increase the maximum isometric force of each muscle
for muscle in muscles:
    muscle.setMaxIsometricForce(muscle.getMaxIsometricForce() * scaling_factor)

# Save the modified model
model.printToXML("path/to/save/modified_model.osim")
```

The provided code snippet demonstrates how to access the muscles in an OpenSim model, define a scaling factor to increase muscle strength, and subsequently modify the maximum isometric force of each muscle accordingly. The modified model can then be saved for further analysis.

By leveraging this example within the framework, researchers can explore the impact of various muscle strength enhancements on biomechanical simulations and gain insights into training effectiveness.

B. Example 2: Computing the Center of Mass using OpenSim

To accurately determine the center of mass of the human body and utilize OpenSim within the framework for biomechanical analyses such as balance assessment and gait analysis, you can use the following prompt:

"Could you please provide a Python code snippet that demonstrates how to compute the center of mass for an OpenSim model using the OpenSim API?"

By using this prompt, you can request the ChatGPT model to generate the desired Python code for computing the center of mass in OpenSim models.

Python Code Snippet:

The following Python code demonstrates how to compute the center of mass for an OpenSim model using the OpenSim API:

```
import opensim as osim

# Load an OpenSim model
model = osim.Model("path/to/your/model.osim")

# Initialize the OpenSim model's state
state = model.initSystem()

# Compute the center of mass
com = model.calcMassCenterPosition(state)

# Print the center of mass coordinates
print("Center of Mass (X, Y, Z):", com[0], com[1], com[2])
```

The provided code snippet showcases how to load an OpenSim model, initialize its state, and compute the center of mass. The resulting center of mass coordinates are then printed for analysis.

By utilizing this example within the framework, researchers can accurately determine the center of mass and utilize this information for a wide range of biomechanical analyses, including balance assessment and gait analysis.

C. Example 3: Extracting Model Details in OpenSim

OpenSim's capabilities enable researchers to simulate human movements and gain insights into the dynamic behavior of the musculoskeletal system, thereby informing injury prevention and rehabilitation strategies.

Here's a prompt you can use to get a Python script that demonstrates how to load an OpenSim model and extract information about bodies, joint sets, marker sets, muscle groups, as well as the number of muscles and coordinates:

"Could you please provide a Python script that demonstrates how to load an OpenSim model and extract information such as bodies, joint sets, marker sets, muscle groups, and the number of muscles and coordinates?"

By using this prompt, you can request the ChatGPT model to generate a Python script that showcases the process of loading an OpenSim model and extracting various information about its components and properties.

Python Script:

The provided Python script demonstrates how to load an OpenSim model and extract information about bodies, joint sets, marker sets, muscle groups, as well as the number of muscles and coordinates:

```
import opensim as osim

# Load an OpenSim model
model = osim.Model("path/to/your/model.osim")

# Print information about bodies
print("Bodies:")
for i in range(model.getBodySet().getSize()):
    body = model.getBodySet().get(i)
    print("- Name:", body.getName())

# Print information about joint sets
print("Joint Sets:")
for i in range(model.getJointSet().getSize()):
    joint = model.getJointSet().get(i)
    print("- Name:", joint.getName())

# Print information about marker sets
print("Marker Sets:")
```

```
for i in range(model.getMarkerSet().getSize()):
    marker = model.getMarkerSet().get(i)
    print("- Name:", marker.getName())

# Print information about muscle groups
print("Muscle Groups:")
for i in range(model.getMuscles().getSize()):
    muscleGroup = model.getMuscles().get(i)
    print("- Name:", muscleGroup.getName())

# Print the number of muscles and coordinates
num_muscles = model.getMuscles().getSize()
num_coordinates = model.getNumCoordinates()
print("Number of Muscles:", num_muscles)
print("Number of Coordinates:", num_coordinates)
```

The script loads an OpenSim model and provides details about bodies, joint sets, marker sets, and muscle groups. Additionally, it outputs the number of muscles and coordinates present in the model.

By utilizing this example within the framework, researchers can extract essential model details, enabling in-depth analysis of biomechanical characteristics and informing their research on injury prevention and rehabilitation strategies.

Discussion

The presented framework, combining ChatGPT, Colab, and OpenSim for biomechanical modeling, offers several notable advantages. The integration of ChatGPT provides a user-friendly approach, allowing researchers to interact with the framework using natural language prompts. Colab, as an online coding environment, eliminates the need for local installations and offers seamless accessibility for running generated models. OpenSim, being an open-source software, facilitates the creation and simulation of musculoskeletal models. This framework empowers researchers and practitioners to efficiently generate models, run simulations, and iteratively debug them using ChatGPT.

However, it is important to acknowledge potential challenges and limitations. One such challenge is the accuracy of ChatGPT responses, which may require careful validation and verification. Additionally, using clinical data in Colab raises security concerns that need to be addressed to ensure data privacy and compliance.

Despite these limitations, the proposed framework paves the way for leveraging AI-driven interfaces to enhance and streamline biomechanical modeling workflows. Future research can focus on refining ChatGPT's accuracy, exploring alternative data sources, and addressing security considerations to further enhance the framework's potential in advancing biomechanics research and applications.

Potential limitations and challenges

One potential limitation of the framework is the accuracy of ChatGPT responses. While ChatGPT is a powerful tool, it may occasionally generate inaccurate or misleading suggestions. Users should exercise caution and critically evaluate the generated models and recommendations to ensure their validity and reliability.

Another challenge relates to the security of clinical data when using Colab. As an online platform, there are potential security and privacy concerns when working with sensitive clinical data. It is crucial to implement proper measures to protect patient information and adhere to relevant regulations and ethical considerations.

The availability and quality of OpenSim models can also impact the framework's effectiveness. Limited access to accurate and validated models may restrict the applicability and reliability of the generated results. Efforts should be made to expand the availability of high-quality models to enhance the utility of the framework.

Biomechanical modeling itself is a complex domain that requires expertise in both biomechanics and computational modeling. Users must possess a solid understanding of the underlying principles to effectively utilize the framework and interpret the results accurately.

Furthermore, running complex simulations in OpenSim may demand significant computational resources. Limitations in hardware capabilities or cloud service quotas could potentially restrict the execution of large-scale simulations. Adequate computational resources should be available to ensure the smooth operation of the framework and the efficient execution of biomechanical analyses.

By acknowledging these potential limitations and challenges, researchers and practitioners can plan accordingly, taking appropriate measures to mitigate risks and optimize the use of the framework in their biomechanical modeling endeavors.

Conclusion

In conclusion, the integration of ChatGPT, Colab, and OpenSim presents a promising framework for advancing biomechanics research and modeling. The interactive nature of ChatGPT streamlines the generation of models and simulations through intuitive natural language prompts. Colab provides a convenient and accessible platform for executing these models without the need for local installations. OpenSim, as an open-source software, enhances the versatility and flexibility of the framework by facilitating the creation and simulation of musculoskeletal models [4].

While the framework offers significant advantages, it is important to acknowledge potential limitations and challenges. These include the accuracy of ChatGPT responses, security concerns with clinical data, model availability and quality, complexity of biomechanical modeling, and computational resource requirements. Addressing these challenges will further enhance the framework's utility and reliability.

Looking ahead, the ongoing advancements in AI technologies and collaborations between the biomechanics and AI communities hold tremendous potential for further innovations in the field. By leveraging the power of AI-driven interfaces and cloud computing resources, biomechanical modeling becomes more accessible, efficient, and impactful.

By addressing the identified limitations and challenges, researchers can harness the full potential of this framework to drive innovation and gain deeper insights into human movement and performance. The integration of ChatGPT, Colab, and OpenSim opens up new possibilities for the biomechanics field and paves the way for exciting advancements in the future.

Acknowledgment

We gratefully acknowledge the invaluable contribution of ChatGPT in assisting us during the writing process of this paper. ChatGPT's remarkable language generation capabilities played a vital role in refining and improving the manuscript's structure and content. The prompt-based suggestions and text generation capabilities provided by ChatGPT enabled us to iterate and enhance our ideas with remarkable efficiency. We express our gratitude for the advancements in AI technologies that have made tools like ChatGPT accessible, empowering researchers and writers in diverse domains.

References

1. OpenAI. (2023). ChatGPT [Software]. OpenAI. Retrieved from OpenAI
2. Google. (2023). Google Colaboratory [Software]. Google. Retrieved from Colab
3. Delp, Scott L., et al. "OpenSim: open-source software to create and analyze dynamic simulations of movement." IEEE transactions on biomedical engineering 54.11 (2007): 1940-1950.
4. Mokhtarzadeh, H., Jiang, F., Zhao, S. and Malekipour, F., 2022. OpenColab project: OpenSim in Google colaboratory to explore biomechanics on the web. Computer Methods in Biomechanics and Biomedical Engineering, pp.1-9.

Appendix

We would like to express our gratitude to the creators of Mermaid and its extension for Google Docs, which allowed us to create this graph seamlessly. The initial version of the graph was generated using a prompt in ChatGPT, and it was then recreated using the Mermaid Live editor (https://mermaid.live/). The combination of these tools enabled us to visualize the integration of ChatGPT and Google Colab in an intuitive and informative manner.

```
graph TD;
  A(ChatGPT) -- " " --> B(Google Colab);
  C(Neuro-Biomechanics) -- " " --> B;
  B -- " " --> D[Gait Analyses];
  B -- " " --> E[Rehabilitation];
  B -- " " --> F[Brain Injuries];
  B -- " " --> G[Sport Injuries];
  style A fill:#F9D5E5, stroke:#F05555, stroke-width:2px;
  style B fill:#E2F0CB, stroke:#00B872, stroke-width:2px;
  style C fill:#D0E0FF, stroke:#2962FF, stroke-width:2px;
  style D fill:#FFF8E1, stroke:#F9A825, stroke-width:2px;
  style E fill:#FFD7D7, stroke:#F44336, stroke-width:2px;
  style F fill:#F8BBD0, stroke:#EC407A, stroke-width:2px;
  style G fill:#E6EE9C, stroke:#9E9D24, stroke-width:2px;
```