

Title: Designing an Effective Prompt for Biomechanics Research using ChatGPT and Open-Source Models: A Human-in-the-Loop Approach

Author(s):

- Hossein Mokhtarzadeh, Sr. Fellow at the University of Melbourne
 - Github: [hmok \(Hossein Mokhtarzadeh\) · GitHub](#)
 - Email: [Hossein Mokhtarzadeh](#)

Access Our Code: [hmok \(Hossein Mokhtarzadeh\) · GitHub](#) & [Colab Notebook](#)

Contents:

Contents:	1
Abstract:	2
Introduction:	3
Methods:	3
Results and Discussion:	4
Conclusion:	7
References	8
Acknowledgments:	8

Abstract:

This project aims to develop a protocol for generating accurate prompts in ChatGPT to facilitate the development and analysis of biomechanical models. The goal is to leverage free software and packages, such as OpenSim and Google Colab, along with ChatGPT, to generate Python code based on the API of the biomechanical models. While this framework focuses on ChatGPT, OpenSim, and Colab, it can be extended to other Language Models (LMs), biomechanical models, and Integrated Development (ID) environments. The framework begins by iteratively refining the prompt through a series of interactions with ChatGPT, leveraging techniques like few-shot prompting and fact-checking. The end result is a well-crafted prompt that generates detailed Python code, easily executable in Colab, to obtain specific biomechanical outputs, such as the center of mass. By sharing these models with the community, this research aims to enhance our understanding of human and animal biomechanics, prevent injuries, and improve overall performance.

Keywords: Biomechanics research, ChatGPT, Prompt engineering, Python code generation, Biomechanical simulations, OpenSim

Introduction:

The utilization of large language models, like ChatGPT, in engineering and computational simulations is rapidly expanding. While ChatGPT is an AI language model, it lacks the capability to directly perform computational modeling tasks that require specific software packages and simulation tools, such as Python environments and biomechanical tools [1-4]. However, through prompt engineering, we can leverage ChatGPT's potential to interactively assist in developing engineering models, such as biomechanical simulations. This research aims to enhance this process by improving the prompt and utilizing it to generate relevant Python code that can be easily executed in a streamlined environment, such as Google Colab [2-4]. By leveraging this framework, researchers can obtain detailed Python code for executing biomechanical simulations and calculations, including outputs like the center of mass, all without the need to install any software locally on their computers.

Methods:

We used iterative method where the user starts with a conceptual and biomechanical question like how to calculate center of mass a biomechanical model? The such a question or concept will be input as an input to ChatGPT (<https://chat.openai.com/>) and a code will be generated by it./ Following this step, the code will be copied and run in Colab and debugged within the colab. If the outcomes is not achieved (e.g. with the current literature data or the users' own procedure established elsewhere), there will be a loop to go through to optimize the process. The biomechanical concept will be checked so to whether this is a concept that can be achieved and plausible (some biomechanical knowledge is required). Then using ChatGPT the same question plus the errors in the Colab will be input. ChatGPT will study and debug its response and learn from it and then provide another python code. This can be run in the Colab again and the process will continue until the desirable result will be achieved. Note that this process is a human-in-the-loop ChatGPT and Colab and biomechanical model (for simplicity OpenSim [1] here) integration this required some basic understanding of these tools and the author assume these knowledge is there for the user. Next we will provide the prompts that we get after several steps of interaction and its final and desirable code generated by ChatGPT. However, it should be noted as the ChatGPT is a predictive tool, the results might change from time to time or from model to model e.g. (GPT4 vs. GPT3, etc).

We employed an iterative method for designing effective prompts in biomechanics research. The process begins with the user posing a conceptual or biomechanical question, such as calculating the center of mass in a biomechanical model. This question is then input into ChatGPT (<https://chat.openai.com/>), which generates code as a response. Subsequently, the generated code is copied and executed in Colab, where any encountered errors are debugged. If the desired outcome is not achieved, whether due to the available literature or the user's own established procedures, an optimization loop is initiated by the user (i.e. A Human-in-the-Loop Approach). The feasibility and plausibility of the biomechanical concept are assessed (requiring some biomechanical knowledge). The question, along with the

errors encountered in Colab, is input into ChatGPT for further study and debugging, enabling it to learn and generate improved Python code (Figure 1).

This cycle continues until the desired result is obtained. It's important to note that this process involves the integration of human-in-the-loop ChatGPT, Colab, and a biomechanical model (specifically OpenSim for simplicity), assuming a basic understanding of these tools by the user. We will present the prompts generated after several iterations of interaction, along with the final desirable code generated by ChatGPT. However, it should be acknowledged that as ChatGPT is a predictive tool, the results may vary over time or across different models (e.g., GPT4 vs. GPT3, etc.).

Note: OpenSim installation is required to run the provided codes. For complete instructions, please refer to our GitHub repository: [Access Our Code]([hmok \(Hossein Mokhtarzadeh\) · GitHub](#)).

If you need assistance with installing OpenSim on Colab, you can ask ChatGPT for help. However, here is a simplified approach that you can follow [1-3]:

To install OpenSim on Colab, you can use the following simplified steps:

1. Start by installing the `condacolab` package, which allows you to use `conda` in Colab. In a Code cell in your Colab notebook, run the following commands:

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

This will set up the necessary packages for `conda` in Colab.

2. Next, you can install the OpenSim package using the official `conda` package from the `opensim-org` channel. Execute the following command in a Code cell:

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

Following these steps will enable you to successfully install OpenSim in Colab, allowing you to proceed with your biomechanical modeling tasks.

The examples demonstrate the extraction of force names, body names, and muscle names from an OpenSim model.

Results and Discussion:

This section provides illustrative examples of applying the proposed method to obtain well-crafted code snippets. Three biomechanical concepts/tasks were tested using the optimized prompts generated by the iterative process. The examples highlight the retrieval of force names, body names, and muscle names from an OpenSim model.

Example 1: Retrieving Force Names from an OpenSim Model

Optimized Prompt: "Provide a Python code snippet to load an OpenSim model from a file and print the names of the forces present in the model."

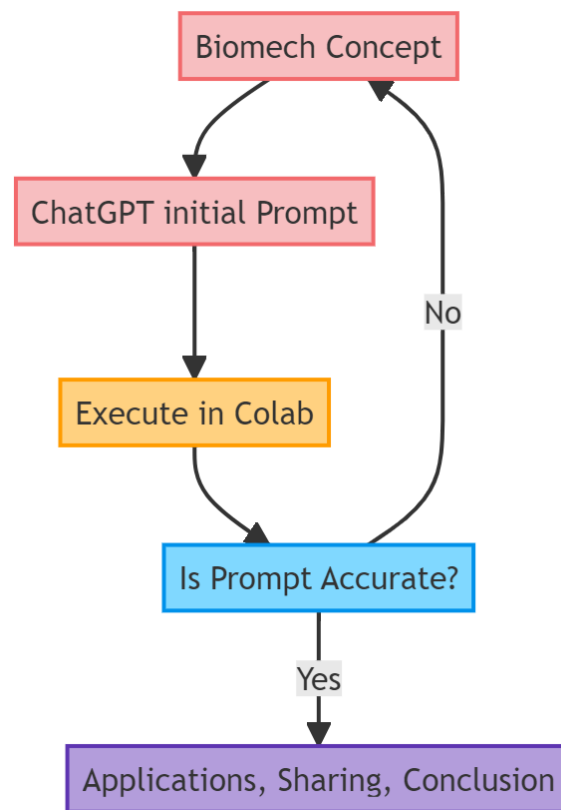


Figure 1. The provided Mermaid code represents a flowchart outlining the process of designing an effective prompt for biomechanics research using ChatGPT where human-in-the-loop is considered for accuracy and validations. The flow starts with understanding the biomechanical concept (represented by box A) and generating an initial prompt using ChatGPT (represented by box B). The flow then proceeds to execute the code in Colab (box C) and evaluates the accuracy of the prompt (box D). If the prompt is accurate, the flow moves to applications, sharing, and conclusion (box E). However, if the prompt is deemed inaccurate, the flow loops back to the initial prompt generation stage, allowing for further refinement and iteration. This flowchart visually represents the iterative process of refining the prompt to ensure accuracy and reliability in biomechanics research.

Results from ChatGPT:

```
```python
import opensim
```

```
Specify the path to the OpenSim model file
model_file = "path/to/your/model.osim"

Load the OpenSim model
model = opensim.Model(model_file)

Get the number of forces in the model
num_forces = model.getForceSet().getSize()

Iterate over the forces and print their names
for i in range(num_forces):
 force = model.getForceSet().get(i)
 print(force.getName())
...

```

### **Example 2:** Retrieving Body Names from an OpenSim Model

**Optimized Prompt:** "Provide a Python code snippet to load an OpenSim model from a file and print the names of the bodies present in the model."

#### **Results from ChatGPT:**

```
```python
import opensim

# Specify the path to the OpenSim model file
model_file = "path/to/your/model.osim"

# Load the OpenSim model
model = opensim.Model(model_file)

# Get the number of bodies in the model
num_bodies = model.getBodySet().getSize()

# Iterate over the bodies and print their names
for i in range(num_bodies):
    body = model.getBodySet().get(i)
    print(body.getName())
...

```

Example 3: Retrieving Muscle Names from an OpenSim Model

Optimized Prompt: "Provide a Python code snippet to load an OpenSim model from a file and print the names of the muscles present in the model."

Results from ChatGPT:

```
```python

```

```

import opensim

Specify the path to the OpenSim model file
model_file = "path/to/your/model.osim"

Load the OpenSim model
model = opensim.Model(model_file)

Get the number of muscles in the model
num_muscles = model.getMuscles().getSize()

Iterate over the muscles and print their names
for i in range(num_muscles):
 muscle = model.getMuscles().get(i)
 print(muscle.getName())
...

```

These examples demonstrate the successful generation of Python code snippets for retrieving specific information from an OpenSim model. The optimized prompts, generated through the iterative process, effectively guide the ChatGPT model to produce accurate and relevant code.

## Conclusion:

Our project showcases a powerful yet user-friendly method to integrate ChatGPT, Colab, and Biomechanical tools such as OpenSim. Leveraging freely available and extensively tested packages, including the free versions of these tools, we have developed a process that empowers users from diverse backgrounds, regardless of programming expertise, LLM knowledge, or biomechanical understanding.

By adopting this framework, various professionals including clinicians, sports scientists, biomechanical engineers, robotics enthusiasts, and even animation and game developers, can significantly improve their understanding of this vital discipline. The outcomes are far-reaching, ranging from more accurate human movement science results to broader applications in fields like robotic engineering and neuromechanics research.

While the framework can be automated via the OpenAI API (with ongoing attempts), we emphasize the importance of human involvement in the loop. This human-guided approach allows for a deeper comprehension of each step and facilitates error detection, ensuring that models align with human instructions. We firmly believe that this framework sets the stage for the development of standardized engineering tools, particularly within the realms of biomechanics, robotics, and neuromechanics research.

Together, we can unlock new possibilities and drive advancements in these fields, ultimately revolutionizing the way we approach biomechanics research and beyond.

## References

- [1]. Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E. and Thelen, D.G., 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*, 54(11), pp.1940-1950.
- [2]. Mokhtarzadeh, H., 2023. Revolutionizing Biomechanics: Integrating AI, Collaborative Coding Environments, and Simulation Software for Enhanced Model Development and Analysis. DOI: <https://doi.org/10.31224/3036>.
- [3]. 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.
- [4]. Mokhtarzadeh, H. and Bagheri, S., 2023. Streamlining C3D File Processing and Visualization: A User-Friendly Approach Using Google Colab and Open-Source Python Packages. DOI: <https://doi.org/10.31224/3088>.

## Acknowledgments:

We express our gratitude to the developers of OpenSim and Colab for their invaluable contributions to Python's open-source packages. Special thanks to the Mermaid team and the creators of its Google Docs extension, as they have facilitated the seamless creation of graphs. Furthermore, we acknowledge the support and assistance provided by OpenAI and ChatGPT in the writing, drafting, and editing processes of this paper.