Multibody Models Generated from Natural Language-Based Text

Johannes Gerstmayr, Peter Manzl and Michael Pieber

May 24, 2024

Multibody Models Generated from Natural Language-based Text

Johannes Gerstmayr, Peter Manzl and Michael Pieber

Department of Mechatronics,
University of Innsbruck
Technikerstr. 13, 6020, Innsbruck, Austria
[johannes.gerstmayr, peter.manzl, michael.pieber]@uibk.ac.at

Abstract
Large Language Models (LLMs) are currently experiencing a high level of attention due to their extraordinary skills. They are experiencing strong growth and new applications with high benefits for industry and business are added every day. In this work, we investigate the current capabilities of LLMs, specifically their application in generating multibody dynamics simulation models from natural language inputs. Specifically, we investigate LLMs that have been trained on our specialized multibody code, Exudyn[1], and which are able to capture and translate the complexities of kinematics and dynamics into functional programming interfaces.

Natural Language Processing (NLP), a crucial aspect of artificial intelligence, involves understanding, interpreting, and generating human language, as well as exploring syntax and semantics. The introduction of LLMs like GPT, which are based on the transformer architecture [1], has significantly expanded the capabilities of NLP in areas such as code generation. These LLMs have been growing in size, as demonstrated by models like Google's LaMDA and PaLM, with PaLM having 540 billion parameters. The development of LLMs faces challenges in dataset curation, with models like GPT-3 and PaLM being trained on terabytes of data from diverse sources. Platforms like HuggingFace [2] play a pivotal role by providing extensive datasets for training these models. Specialized LLMs exist for example for science (e.g. Galactica) or for software engineering.

In the investigations shown in the full paper [3], smaller LLMs were not capable of performing complicated modeling and code generation tasks. Therefore, more advanced GPT (Generative Pre-trained Transformer) models are used and evaluated. In order to understand the functionality of the approach, the modeling language in the Python package Exudyn is briefly mentioned as well. We note that creating multibody simulation models requires LLMs to have foundational knowledge of mechanical engineering concepts, in particular of kinematics, as well. For our multibody models which are generated from natural language-based text, so far we mainly looked at proprietary LLMs, such as GPT-3.5 and GPT-4 of OpenAI, using the ChatGPT interface, PaLM-2 integrated into Google's Bard and LLaMA-2 of Meta-AI. In the present work, we evaluate leading open source LLMs of HuggingFace, which run locally on advanced GPU-hardware. These models are between one and two magnitudes smaller than GPT-3, however showing performance for creating simulation code comparable to ChatGPT-4, see e.g. WizardCoder.

GPTs are based on the transformer architecture [1], which allows sequence-to-sequence tasks and which is key to current LLMs. The transformer includes the attention mechanism and continuously processes query, key, and value vectors within many attention heads. These attention heads contain large neural networks, which make up a large number of parameters in these LLMs, such as 70 billion parameters in the LLaMA-2 model. Instead of processing text directly, text is translated into so-called tokens, which represent short words and most frequent combinations of letters or other characters. It should be mentioned that data preparation is a huge task, not capable by regular research groups, and the full training for the smaller LLaMA-2 70B model took 1.7 million GPU hours, which requires more than 1000 GPUs to train a single model within 2 months of computation.

For the above reasons, the goal of this work is to improve the performance of existing LLMs to generate Python code for the Exudyn package, to provide a baseline for capabilities of future LLMs and to show how they could be improved. Models in Exudyn are created using Python commands to initialize

[1] https://github.com/jgerstmayr/EXUDYN

Table 1: Number of modeling (*mod*) and syntax (*syn*) errors of LLMs for a double pendulum.

	GPT-3.5		GPT-4		Bard	
Example 5	e_{mod}	e_{syn}	e_{mod}	e_{syn}	e_{mod}	e_{syn}
trial 1	2	3	0	0	2	1
trial 2	2	6	0	0	4	2
trial 3	0	2	0	0	2	3

a system, create nodes, bodies, connectors, and joints. Finally, the system is assembled, simulated for a given time span, and visualized. For evaluation of performance, a set of text-based instructions is evaluated with several LLMs. Such an instruction could read similar to textbook descriptions:

> *Consider a double pendulum using two mass points and distance constraints in Exudyn, using the previous update info with mbs.Create functions. The mass points shall have 1 kg and the length of the first link is 2 m and the second link 1 m. The initial configuration is an L-shape, where the first link points along the x-axis and the second link points up. Gravity acts in a negative y-direction. Draw the nodes not as points, mass points have a size of 0.1 m. Please write the code with no comments and in one block.*

While the direct request of this instruction to LLMs gives almost perfect code, there are usually some small errors due to the many dependencies of nodes, markers, bodies, and joints in the system. In order to improve the quality of the output, we observed that additional modeling information and examples that are provided to the LLM in the context, reduced errors a lot. As a result, error-free (zero-shot) results are obtained in particular by GPT-4, compare Table 1, which shows one of our experiments, a double pendulum with mass points with three trials. Further experiments consider rigid bodies and joints, see Figure 1 for the visualization of a triple pendulum and a slider crank. In addition to previous work [3], we will also investigate the fine tuning of open source LLMs, such as StarCoder. To achieve this, we create a larger set of multibody examples together with modeling information and use HuggingFace's Trainer to train next token prediction as well as question answering. Finally, LLM results are exposed to human evaluation.

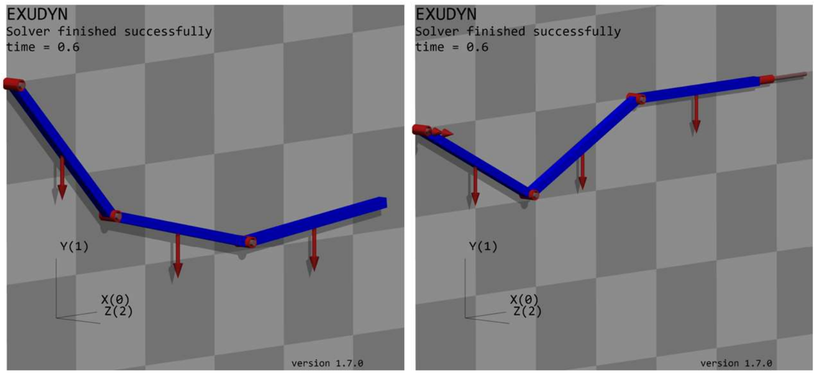

Figure 1: Visualization of a triple pendulum with rigid bodies and a slider-crank mechanism using the Python scripts created by GPT-4.

Conclusions and Outlook

In conclusion, this work intends to offer a comprehensive insight into the current capabilities, potential improvements and future potential of LLMs in the field of multibody dynamics simulation. In addition to the textual description and data, figures and sketches are essential for understanding a multibody system, which will be enabled by future ChatGPT-4V (vision). Furthermore, offline models could be trained in a loop, and thus, interaction with multibody simulation software could be significantly alleviated in the future.

References

[1] Vaswani, A.; et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, 30, 2017.

[2] Wolf, T.; et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771, 2019.

[3] Gerstmayr, J.; Manzl, P.; Pieber, M.: Multibody Models Generated from Natural Language. Multibody System Dynamics, 2024. https://doi.org/10.1007/s11044-023-09962-0