Paper ID #45347

Introducing AI into an undergraduate Kinematics of Machines course

Dr. Heather Louise Lai, State University of New York at New Paltz

Heather Lai is an Associate Professor of Mechanical Engineering at SUNY New Paltz, NY where she teaches courses in dynamics, system dynamics, finite element analysis and computer simulation. Her professional background and research interests include automotive vibration (Motorola Inc.), musculoskeletal biomechanics (BME, Wayne State University), room acoustics, wind farm acoustics and the dynamic behavior of 3D printed multi-materials. Over the past 8 years, she has collaboratively developed a number of new and revised courses, including a new System Dynamics Lab. She has also worked with a number of SUNY students to investigate different aspects of 3D printed multi-material structures.

Introducing AI into an undergraduate Kinematics of Machines course

The integration of emerging technologies, such as Machine Learning and Artificial Intelligence (ML/AI), into traditional mechanical engineering courses has the potential to transform how foundational concepts are taught and applied. This paper discusses the initial implementation of ML/AI-related computational techniques in an undergraduate Mechanical Engineering course on Kinematics of Machines. Specifically, this first attempt at updating the course includes three main modifications.

The first is the early introduction and extensive use of MATLAB to enhance students' understanding of analytical and computational methods as applied to forward kinematics problems. The second builds on the first through the introduction of optimization and machine learning tools to assist in students' synthesis of four-bar mechanisms. The third develops the concepts further through a student-led research component where students explore and evaluate the use of advanced AI tools, such as deep learning-based generative models, in kinematic synthesis. These modifications aim to deepen students' grasp of both traditional and modern approaches to kinematic analysis and design.

The primary focus of this paper is the scaffolding required to support undergraduate students, with little computer programming training, as they learn how to implement these advanced technologies. The course modifications aim to provide students with basic, pre-written code and introduce ML/AI techniques gradually. This approach allows students to gain exposure to computational tools while simultaneously reinforcing their understanding of core engineering principles. This paper also addresses the instructor's experiences, both the benefits and challenges, of incorporating AI into a traditionally structured mechanical engineering course. It provides insights into how these tools can enhance the learning process without introducing unnecessary complexity, aiming to establish a framework for integrating ML/AI into other undergraduate engineering courses. The discussion emphasizes the need to balance educational enrichment with practicality, especially in classes with students of varying levels of programming experience.

The use of ML/AI in kinematic machine design

Machine learning and artificial intelligence are increasingly being applied in the synthesis and optimization of kinematic four-bar mechanisms, enhancing both the design process and the performance of these systems. In this modified course students will be exposed to both optimization machine learning techniques and deep learning-based generative model approaches.

The design of a four-bar mechanism often involves multiple objectives and constraints, such as minimizing mechanical stress while maximizing motion efficiency or achieving a specific motion trajectory. ML algorithms, particularly optimization techniques like Genetic Algorithms (GA), along with more advanced AI methods such as deep learning, can automate and improve this process by efficiently searching through a large space of design possibilities. [1, 2, 3] GAs mimic natural selection processes, evolving better designs through iterations. In four-bar mechanism synthesis, GAs can optimize the estimation of parameters related to link lengths and joint positions to achieve desired motion profiles (e.g., coupler curve shape or motion path) without manually solving complex equations. Deep learning methods can be trained on large datasets of four-bar designs and their corresponding motion characteristics. Once trained, these models can quickly predict optimal configurations based on desired outputs, significantly speeding up the design process.

Rational for integrating ML/AL in Kinematics of Machines

Kinematics of Machines, with its focus on analyzing the motion of mechanical systems—such as position, velocity, and acceleration—serves as an effective platform for introducing ML/AI into the undergraduate mechanical engineering program. It involves theoretical mechanical principles that are applied in very practical ways to mechanism design for a wide range of relevant applications which appeal to students. While many of the techniques used were established well before the widespread use of computers, the effective utilization of modern optimization tools can significantly enhance these processes. The computational nature of kinematic analysis aligns well with the capabilities of ML/AI tools. Kinematic systems, especially in multi-link mechanisms like four-bar linkages, often involve complex relationships between variables that can be difficult and time-consuming to solve using traditional methods. ML/AI techniques can streamline these processes by recognizing patterns, integrating additional constraints, and enabling more effective mechanism synthesis. ML/AI techniques, particularly optimization algorithms, can directly improve the accuracy and speed of these designs. Introducing these tools within this course exposes students to modern computational methods that are becoming increasingly relevant in real-world mechanical engineering applications.

Additionally, Kinematics of Machines, though fundamental, provides a well-defined problem space with clear inputs and outputs. This structured environment allows for the gradual introduction of ML/AI, providing students with digestible and understandable applications of advanced techniques without overwhelming them. MATLAB, which is commonly used in the course, serves as an accessible platform for integrating these tools, given its extensive support for both numerical analysis and machine learning toolkits.

Modification 1: Position Analysis using computer based analytical and computational methods

The study of Kinematics of Machines begins with position analysis. These problems start with a kinematic model, labelling all rigid body links, all joints (kinematic pairs), joint coordinates, calculating the mobility, identifying independent variables (keeping in mind that the number of input variables is the same as the mobility of the mechanism) and dependent variables. Generating equations and solving for the dependent variables can be completed using graphical methods using analytical methods based on Geometrically developed equations or Vector loop equations. Traditionally, mechanism analysis has been completed using graphical methods, which make use of scaled drawings, such as velocity and polygon vectors, or methods such as instantaneous centers utilizing very carefully crafted scaled drawings, in which the solution relies on geometric values measured from the drawing. Not only is this very difficult to realize for any but the most basic problems, but it also is impractical and rarely used in any current engineering practices. Analytical methods can be developed for solving for the dependent position variables directly. But, once again, these methods can only be applied to a small class of mechanisms, granting that in some situations, can be applied like building blocks to more complex structures.

In leu of graphical and analytical methods, computational methods that utilize numerical root finding techniques, such as the Newton-Raphson method (which is included in the adopted text) provide a good bridge to more complex optimization algorithms. For example, in performing position analysis, in addition to utilizing code provided in the text to implement the application of the Newton-Raphson, method, students in the course were taught to use the MATLAB built-in function, **fsolve** to numerically estimate link positions for a specified input angle.

The **fsolve** function attempts to solve a system of equations by minimizing the sum of squares of the components by using one of three algorithms: Trust-region, Trust-region-dogleg or Levenberg-Marquardt. Through the use of more advanced algorithms students begin to be exposed to the use of complex algorithms for more effectively solving kinematic problems. While this can provide a streamlined way to estimate the link positions, because there are multiple solutions to the position equations (relating to open and closed configurations of the mechanisms), computational methods may convert on the wrong root. Helping students to see both the benefits and the risks of the use of computational methods is key to developing engineers able to appropriately apply developing computational technologies.

Modification 2: Mechanism synthesis optimization using ML algorithms

For the initial implementation of the course redesign, the primary use of ML/AI is focused on optimization of mechanism synthesis. One of the most common applications of ML in four-bar mechanism synthesis is in optimizing dimensional synthesis. Traditionally, the synthesis of four-bar mechanisms involves solving complex equations and iteratively adjusting link lengths and angles to meet specific design requirements (e.g., achieving certain path or motion characteristics). In an undergraduate course, this process has often been taught using a "guess and check" method, which limits the ability to incorporate complex constraints into potential solutions. By introducing optimization techniques, students are able to avoid suboptimal designs and apply more meaningful constraints during synthesis. Not only will they learn about various types of mechanism synthesis, but also how optimization, along with ML/AI, can enhance the overall design process and outcomes.

Fourbar mechanism synthesis can take several different forms, including function generation, position guidance and path generation. Optimization of function generation has been integrated into the modified course content. Function generation involves designing a fourbar mechanism in which the relationship between the input and output angles follows a predefined function. The final design will include both the link length dimensions and the input crank initial and final angles as well as the output link initial angle. In textbook problems, the selection of the link angles is usually specified ahead of time, and only if the mechanism synthesis cannot be completed, or returns an impractical design, will they be modified, resulting in a new set of synthesized link lengths. In reality, these angles are design parameters that can have a large effect on the synthesized link lengths and should be carefully selected by the designer such that the end design can meet a number of potentially completing objectives. This is where design optimization comes in. Using MATLAB, code can be developed to optimize the input and output angles to synthesize a mechanism which meets specified criteria, while resulting in the smallest structural error.

In preparation for covering synthesis optimization in this course, a MATLAB Grader assignment has been developed which performs optimized linear function generation through the use of Freudenstein's equations with the use of a straight-line function via Chebyshev spacing. The optimization is related to the limit angles of the design. The code incorporates a user-defined function which passes in the input crank's initial and final angles and calculates the structural error associated with the synthesized link lengths. Using several different optimization algorithms, including interior-point, pattern search and genetic algorithm, available through the Optimization and Global Optimization Toolboxes, comparisons can be made about the efficacy of and efficiency of the different optimization algorithms.

Modification 3: Addition of deep learning to mechanism synthesis project

Additionally, although more advanced AI deep learning was not directly applied to the student's work, due to time constraints and limited computational resources, students are assigned a project that required them to explore and learn about the application of advanced AI methods to mechanism synthesis. In order to better understand the basics of AI and specifically its application to mechanism synthesis, students will begin by reviewing several provided journal articles [2,3], and from these, identify other useful resources. They will then be asked to describe from a high level how these methods could be implemented in their designs, to compare the expected results with those from the ML optimization techniques used and to make recommendations for how AI could be utilized in future designs.

At the time of writing, the first offering of this course is still underway, MATLAB Grader assignments have been developed and implemented related to mechanism analysis, and students are currently using these to aid in completion of structured (homework) assignments. Additional MATLAB Grader assignments have been developed to perform function generation synthesis using the built-in MATLAB optimization functions described earlier and will be deployed in order to aid students in the completion of a mechanism synthesis project.

Barriers to the implementation of AI to undergraduate engineering courses

There are a number of barriers to implementation of AI related solutions in an undergraduate mechanical engineering course. Some of the most significant identified so far include:

Lack of programming experience. While the majority of the students in the course have had some computer programming experience, as a programming course is not a prerequisite for the course, there were several students who did not have familiarity with even basic programming skills.

Instructor's lack of specific knowledge related to AI implementation. I have some basic experience using ML/AI to perform clustering and characterization tasks on extracted features in large data sets and from image data. However, my fundamental knowledge of the topic is also limited, which hampers my ability to develop projects that are properly scoped and provide the structure and explanation required for students' intuitive development of code for performing AI tasks.

Limited computing power. Implementing AI based computer programs requires a large amount of computing power. Some campuses may already have a mechanism in place (such as server banks or cloud-based services) for providing students with access to low-cost compute hours, however, our campus has not yet developed a means to provide that to either students or faculty. In order for students to be able to engage in AI projects of any significance, additional resources will need to be developed at the campus level.

Scaffolding for facilitating the implementation of AI in undergraduate courses

In order to overcome the identified barriers and facilitate learning without undo frustration on the part of the students, computer programming scaffolding was put in place. A description and preliminary evaluation of this includes:

Choice of textbook with focus on computational methods using MATLAB. The textbook, *Introduction to Mechanism Design: With Computer Applications* by Eric Constans [4], was selected because it emphasizes implementation in MATLAB, helping to integrate programming into the course. By presenting both kinematic theory and programming concepts in a well-organized, cohesive way, it has normalized the use of programming for the solution of kinematic problems, making it accessible even to students with no prior programming experience.

Reassurance that this is not a programming class. Because program development is not the focus of the course, and students are given all of the building blocks needed. Students are regularly reminded that they will be responsible for combining prewritten portions code that they have been provided and will be responsible for using this code to perform analysis, rather than write their own code from scratch. Regular reassurance is provided that they are not expected to leave the course as programmers.

MATLAB Grader. Partial code (in addition to the extensive code provided in the text) has been provided to students through MATLAB Grader [5] – a browser-based tool that allows the instructor to develop assignments in which students are able to test portions of code in an automatically graded MATLAB coding environment. This has increased students' confidence with the code that they are writing in class with the positive reinforcement gained through completion of the step-by-step assessments.

MATLAB optimization and ML/AI resources. MATLAB has extensive documentation on each of its built-in functions, along with we developed tutorials relating to ML/AI. Links to this information is regularly include in MATLAB Grader assignments, and it often referred to during class. It will also be provided as part of the required resources for the mechanism synthesis project. [6]

In retrospect, the modification of this course to implement ML/AI has been challenging, and the benefits to the students still need to be evaluated. Changes made based on the lessons learned in this first implantation should lead to improvements which will continue to increase the benefit of incorporating ML/AI into fundamental courses such as Kinematics of Machines.

References

Constans, E. (2018). *Introduction to Mechanism Design: With Computer Applications* (2nd ed.). CRC Press.

[1] S. Deshpande and A. Purwar, "A Machine Learning Approach to Kinematic Synthesis of Defect-Free Planar Four-Bar Linkages," *Journal of Computing and Information Science in Engineering*, vol. 19, no. 2, p. 021004, June 2019. doi: 10.1115/1.4042325

[2] C. M. O'Neill, L. S. Seif, and N. M. Kandasamy, "Deep Generative Models in Engineering Design: A Review," *Journal of Mechanical Design*, vol. 144, no. 7, p. 071704, July 2022. doi: 10.1115/1.4052998

[3] S. Lee, J. Kim, and N. Kang, "Deep Generative Model-based Synthesis of Four-bar Linkage Mechanisms with Target Conditions," *arXiv*, Feb. 2024. [Online]. Available: https://arxiv.org/abs/2402.14882

[4] E. Constans, *Introduction to Mechanism Design: With Computer Applications*, 2nd ed. Boca Raton, FL: CRC Press, 2018.

[5] "MATLAB Grader," MathWorks, [Online]. Available: https://grader.mathworks.com/. [Accessed: 16-Oct-2024].

[6] "Artificial Intelligence," MathWorks, [Online]. Available: https://www.mathworks.com/solutions/artificial-intelligence.html. [Accessed: 16-Oct-2024].