Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #49071

A Web-based Tool for Generating Bond Graphs and Differential Equations for Mechatronic Systems

Prof. Pradeep Radhakrishnan, Worcester Polytechnic Institute

Dr. Pradeep Radhakrishnan is an Associate Professor of Teaching in Mechanical Engineering and Robotics Engineering at Worcester Polytechnic Institute. Dr. Radhakrishnan teaches fundamental courses in mechanics and design at the undergraduate level.

David C Brown, Worcester Polytechnic Institute

David Brown is a Professor Emeritus in the Computer Science Department of Worcester Polytechnic Institute. He specializes in Human Computer Interaction and the uses of AI in Computational Design.

Margaret E Earnest, Worcester Polytechnic Institute

A Web-based Tool for Generating Bond Graphs and Differential Equations for Mechatronic Systems

Abstract

The paper describes a web-based tool for generating bond graphs and differential equations from user-constructed system diagrams. This tool includes features such as differential equation generation, an undo-redo system, exporting URLs (universal resource locators), and an interactive tutorial. The tool is designed to act in a way that mimics the method of teaching bond graphs in lectures and textbooks. Students can use the tool to verify both the bond graphs generated by them and the differential equations.

The web application was assessed in a senior level course, and students had a positive reaction to the application, with many wishing they had been able to use it earlier in the course. We found that over 91% of users were satisfied with the process of creating the system diagram along with saving and opening the same. All other features earned positive reactions from at least 95% of users. URL generation had the highest satisfaction rate, with a favorable reception from 100% of users. Most students' dissatisfaction came with small user interface and backend bugs. Students also requested improvement of a few features, such as labeling system diagram elements with hover text, and asked that some features, such as panning the graph, be added to the tutorial. The paper will describe the various features of the application as well as results from user studies.

Keywords: Bond graph generation; State equations; system modeling

1. Introduction

Mechatronic systems are a class of systems that combine mechanics and electronics [1]. In fact, the four major components of such systems are the mechanism, sensors, control unit and actuators. Mechanisms relate to mechanical translation, mechanical rotation, or thermo-fluids. Sensors are important to collect data, which will be instrumental in determining the control strategies as part of the control unit. The appropriate decisions from the control unit are then transferred to the actuator, which will then power the machine. Mathematical modeling of such mechatronic systems is integral to the development of control strategies as well as to understand the overall performance of the system. There are different methods available to generate models such as those based on Newton's second law and Kirchoff's laws. The bond graph technique, introduced by Henry Paynter, is used to describe dynamical mechatronic systems, and to generate differential equations. This is a graphical technique and provides a symbolic approach to deriving the differential equations [2]. Since the method is domain-independent, students will find its graphic technique engaging as well as easier to follow compared to other techniques.

A node in a bond graph is called an element and an edge a bond. An element can be one of the following: an R-dissipation element; a source of flow or effort; an I-storage element; a C-storage element; a transformer; a gyrator; a 0-junction; or a 1-junction. Bonds are denoted by a line with a half-arrow at the head (see Figure 1). The direction of the half-arrow is called flow direction. The other direction is distinct from flow direction; it is called causality and is denoted by a short perpendicular bar on the bond: a causal stroke. Each bond is labeled with an effort (e) and a flow (f). This effort is always labeled on the left or on top of the bond, and the flow is below or to the right of the bond. The flow direction half-arrow will always be on the same side of the bond as this flow label. Figure 1 shows some examples of bonds. In the mechanical domain, force and torque are the effort and translational or angular velocity are the flow. In the electrical domain, effort would be voltage and flow would be current.

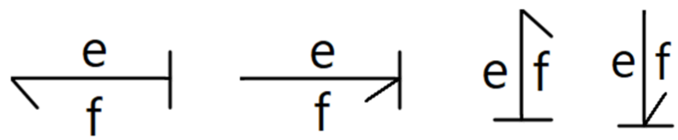

Figure 1: Possible orientations of flow direction (half-arrow) and causal stroke (represented by the horizontal line perpendicular to the bond).

Figure 2 below shows the process of determining the causal bond graph for a two-mass and spring system with an applied load. Detailed process on bond graph generation is available in Karnopp et al. [3]. Step 1 of the process requires assignment of appropriate velocities, V, on each of the masses in the system. This is followed by developing an initial bond graph in step 2. Here, M refers to mass of the object and K indicates the spring's stiffness. Causalities are assigned in Step 3. Causality is an important step before the derivation of equations and helps identify the energy storing states. For this system, there are three states and those are p1', p2' and x', which are derivatives of momentum p of mass 1 and mass 2 and displacement x of spring.

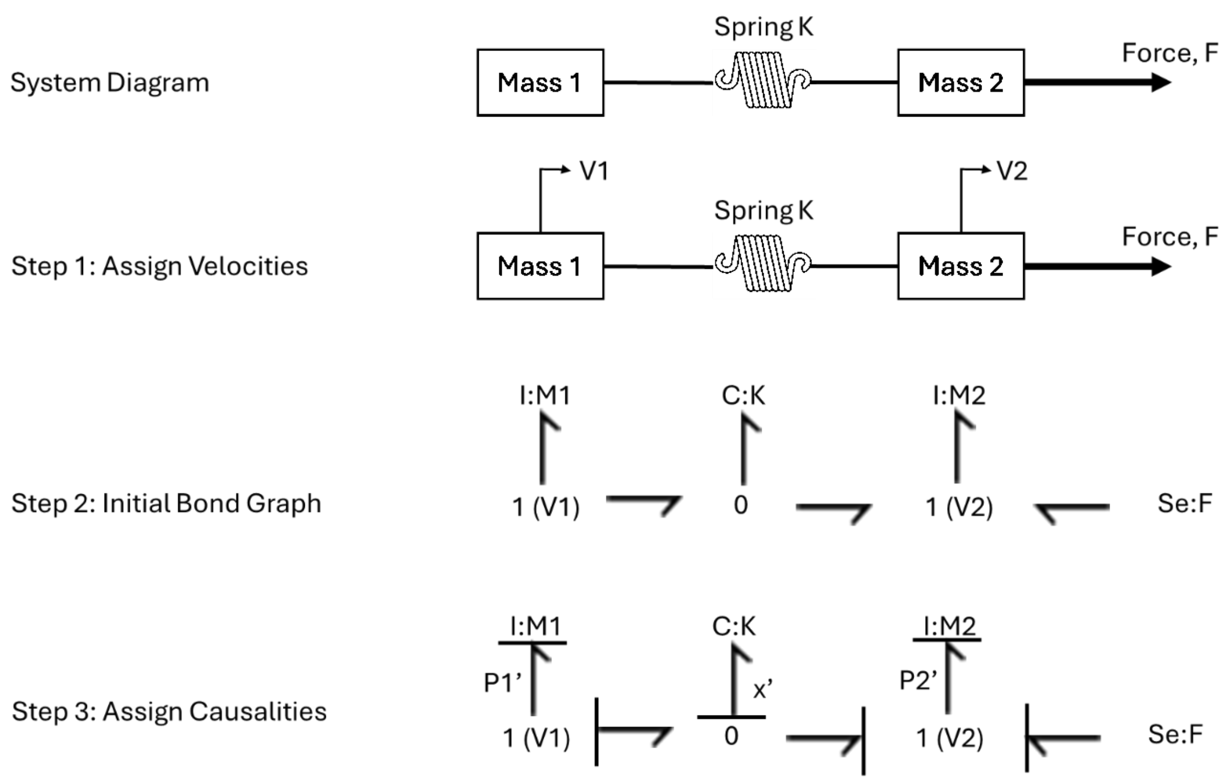

Figure 2: Derivation of causal bond graph from system diagram.

Once the causalities have been assigned, the next step would be to determine the state equations or differential equations (shown in Figure 4). For that, the first step would be to label all the efforts and variables as shown in Figure 3.

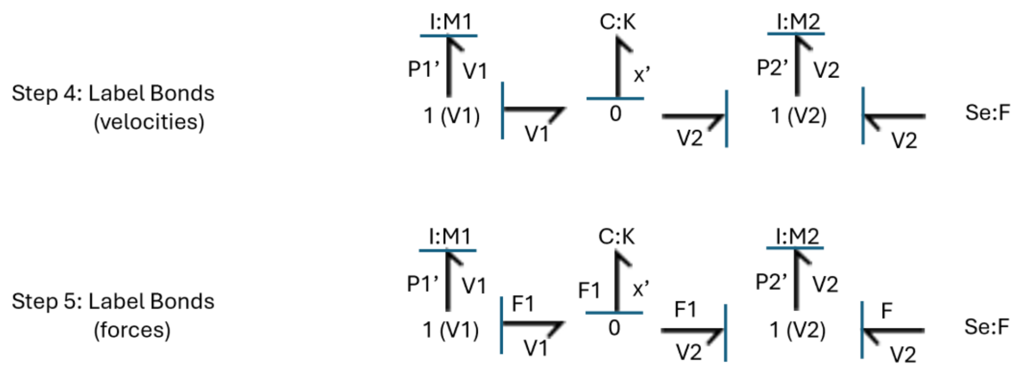

Figure 3: Assignment of effort and flow variables.

The next step would be to derive the equations as shown below in Figure 4.

Step 6: Equation for p1'

$$\dot{p}1 = -F1 = -K * x$$

Step 7: Equation for p2'

$$\dot{p}2 = F1 + F = K * x + F$$

Step 8: Equation for x'

$$\dot{x} = V1 - V2 = \frac{p1}{M1} - \frac{P2}{M2}$$

Figure 4: Derivation of state equations.

Since this is a different approach to generating mathematical models, it is important to complement lectures with appropriate software tools to improve student comprehension. Unfortunately, there are no easy-to-use tools available for this purpose and hence a tool was developed at Worcester Polytechnic Institute (WPI). The software tool in its current version allows users to construct a system diagram and it automatically generates the bond graph and state equations. This will be useful for students to verify their work. This will also be a way for students to be connected to and refresh this topic.

The paper is organized as follows. Section 2 will present information on the related software and past work in this area. This will be followed by a detailed explanation of the various features of the bond graph software in Section 3. Section 4 will describe the results of extensive user testing conducted in related courses followed by concluding remarks in Section 5.

2. Related Work

There are no software programs that can automatically generate bond graphs from system diagrams. Only one program called "20-sim" exists for bond graph. This application can model bond graphs and icon diagrams – their system diagram equivalent – for electrical, mechanical, thermal, and hydraulic systems. But it does not automatically generate bond graphs. It requires users to construct bond graphs manually. Its interface consists of a canvas with an alignment grid, and its primary menu is a panel of component options listed on the left side of the screen. Instead of using text to denote menu options, 20-sim opts for icons with images of the different elements [4]. A 20-sim user must construct bond graphs manually instead of generating them from icon diagrams. The user must then instruct the application to "inspect" the bond graph, at which point it can display initial and final state equations in a separate window. The 20-sim interface is shown in Figure 5.

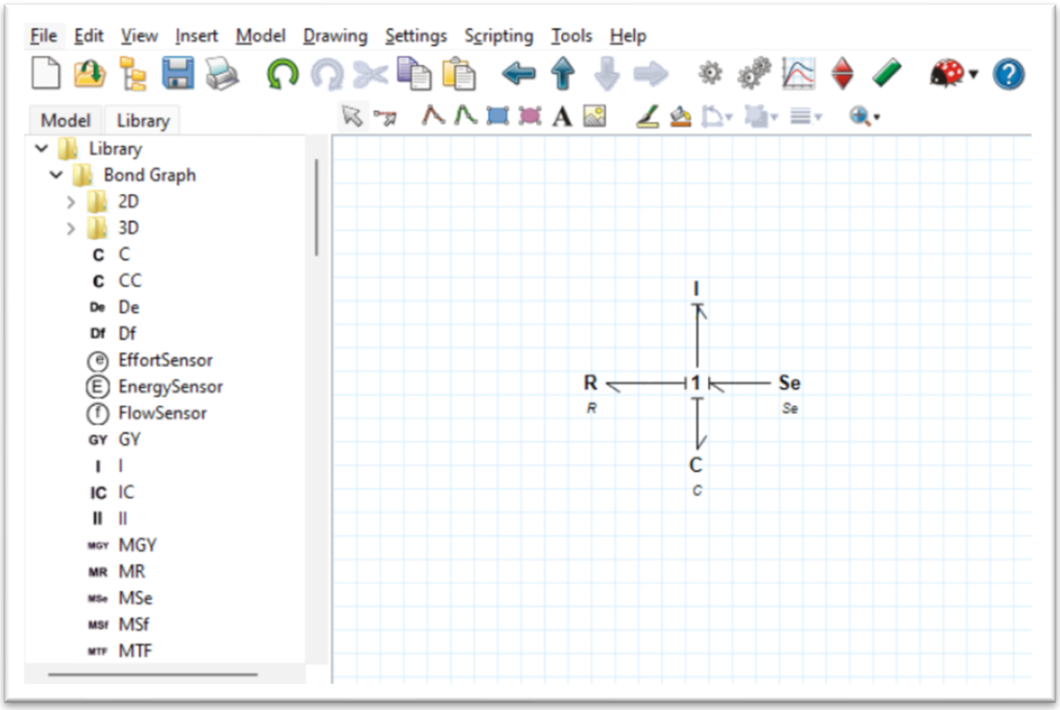

Figure 5: A bond graph as displayed in the 20-sim interface.

To create a tool that can also be useful from an educational standpoint, a graph-grammar based Windows-desktop application for generating bond graphs was developed in 2016 [4,5] by students at WPI. That tool allowed users to construct a system diagram and generate causal bond graphs. This was an initial prototype that had various user interface issues that prevented utilization of the program in coursework. An updated version of the system was developed in 2020 [6] where the user interface was improved, along with updating the backend algorithm for improved performance, including a speed-optimized depth-first search algorithm. But that system was limited to Windows-based computers and did not have the ability to generate state equations. This meant that students, who did not have access to Windows computers, would either need to use a complex solution to get the software running, or not use the software at all. In order to have a universal application, the desktop version of the bond graph software was transitioned into a web-framework. This would mean that all users, with access to a modern web browser, would be able to access the system. This will be a vast improvement over the previous system as it adds support for both Mac and Linux systems. Besides, the new implementation included state equation generation along with automated layout of bond graphs on screen to avoid users rearranging them manually.

3. Software Interface and Features

The web-based software is being developed as free, open-source, and browser-based application for bond graph generation from system diagrams (https://boglweb.github.io/). A screenshot of the application in Figure 6 shows the *Canvas, Element Palette and ToolBar.* The *Canvas* is where the user will create the system diagram. The *Canvas* in Figure 6 shows a mechanical translation system consisting of two masses, two springs and a damper with an applied force. This system is constructed from the *Element Palette* through a drag and drop interface and connecting different elements. The *Element Palette* is organized by system type that emphasizes commonly used components (shown in Figure 7) and uses standard representations for ease of identifying various components. The *Canvas* also includes the *Modifier* panel, *Zoom* options, as well as a *Generate* button to generate causal bond graphs and differential equations.

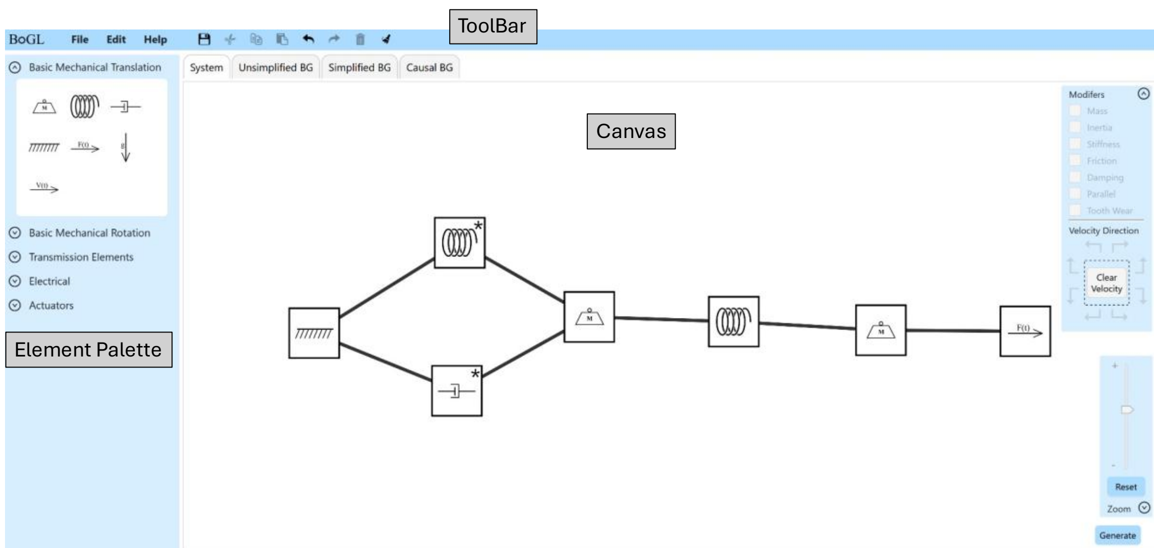

Figure 6: Screenshot of the bond graph software shown with a mechanical system and annotations for key features such as Canvas, Element Palette and ToolBar.

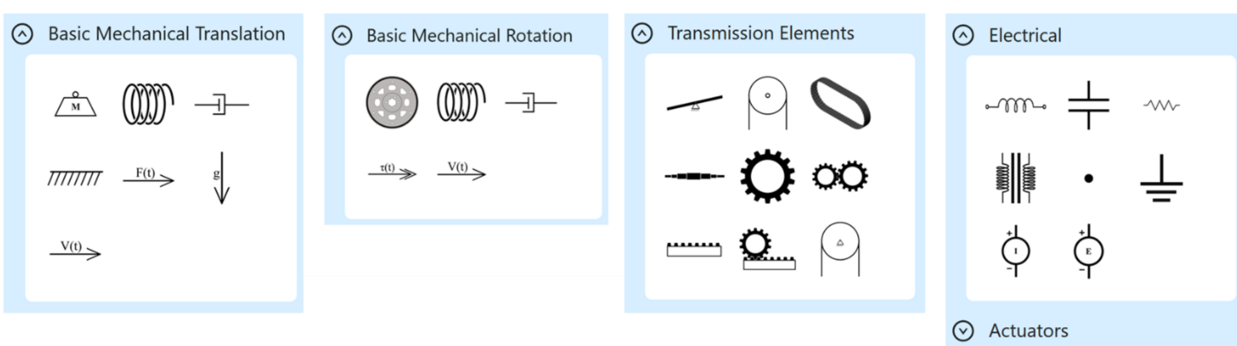

Figure 7: The hierarchical Element Palette.

The *Modifier* panel allows users to specify additional conditions such as friction for any mass, stiffness characteristics for a shaft and so on. This is usually specified by clicking on a particular component and selecting the appropriate option. In Figure 6, you will see that there is an asterix symbol on the spring and damper located between ground and mass, indicating the presence of a modifier. The modifier used in this scenario is to indicate that the spring and damper are in parallel. While that does not necessarily influence the outcome of this bond graph and generated equation, it does indicate to the user that the components have been modified. The *Modifier* panel shown in Figure 6 also includes options for specifying velocity directions on mass. While coordinate axis and velocities do not affect the result in this version, those features fully integrated in the future. Below the *Modifier* panel is the *Zoom* functionality, which allows users to increase the size of *Canvas* elements.

The *ToolBar* consists of options typically found in standard software such as opening and saving user-created files. In this web-version, there are additional features to make the software more useful to students. It is often the case then, when working on a homework assignment, or other project, students and professors need to quickly share information about the system they are working on. This software tool includes a method to encode system diagrams within URLs (Universal Resource Locator). This allows for a much faster way for users to share system diagrams. Since a URL is just a string of text as opposed to a file, it is much easier to share over systems such as email, or other messaging services. The URL approach also makes it easy to submit system diagrams for homework, as a grader would just need to click on the link instead of having to download and open a file. This added efficiency was the main motivation for adding this feature. This software also included a full-stack undo-redo feature allowing for more than one action to be undone by the user.

The *ToolBar* also includes a link to a series of guides that are hosted online (https://bogl.mech.website/). These guides described the various parts of the system as well as providing information on how to perform various tasks within the system. This web-version of the software also includes an interactive tutorial (as shown in Figure 8) to help users learn about the application. In addition, there are some prebuilt examples that allow users to quickly assess the features of the application.

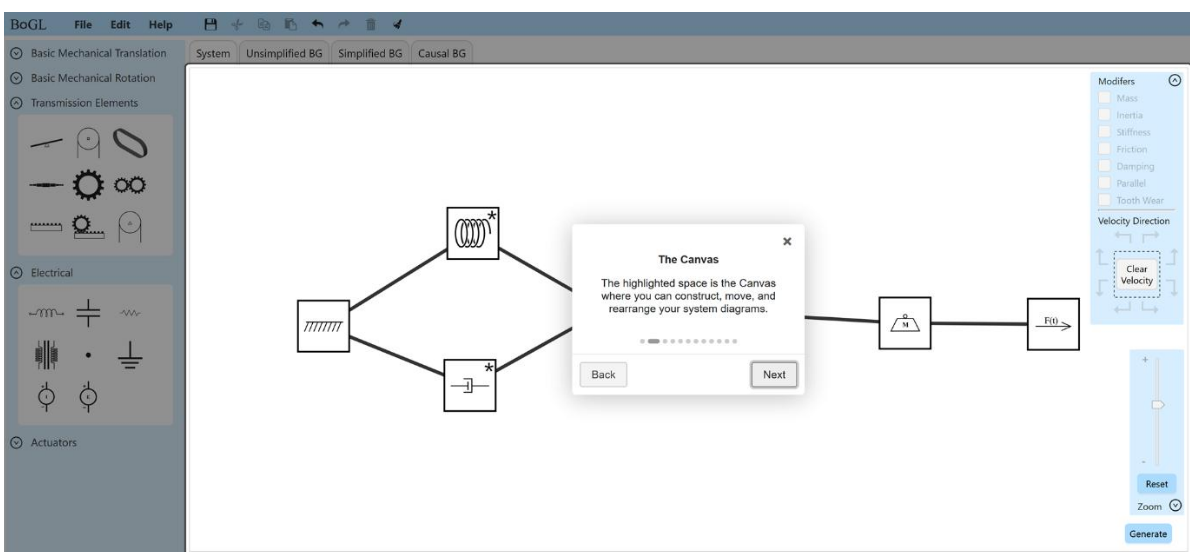

Figure 8: Screenshot of the interactive tutorial within the application.

On the bottom right corner of the *Canvas* (Figure 8), there is a *Generate* button. This allows the system to generate the bond graph of the system displayed on the *Canvas*. This takes a few seconds and various types of bond graphs are generated. Under the *Unsimplified BG* tab, the bond graph shown in Figure 9 will be displayed. This is the first step in the bond graph generation process where the velocity and effort junctions as well as representations for masses, springs and dampers are shown. This is an intermediate step between steps 1 and 2 shown in Figure 2. As you may notice that the graph is not displayed similar to how it is typically done manually. This is because the force-directed algorithm [7] that is implemented requires further adaptation to fit the needs of displaying bond graphs. Upon manual reorganization, the bond will appear as shown in Figure 10. Note that Figures do not show the bond (line with half arrow) and that is why this is termed as an *Unsimplified BG (Bond Graph)*.

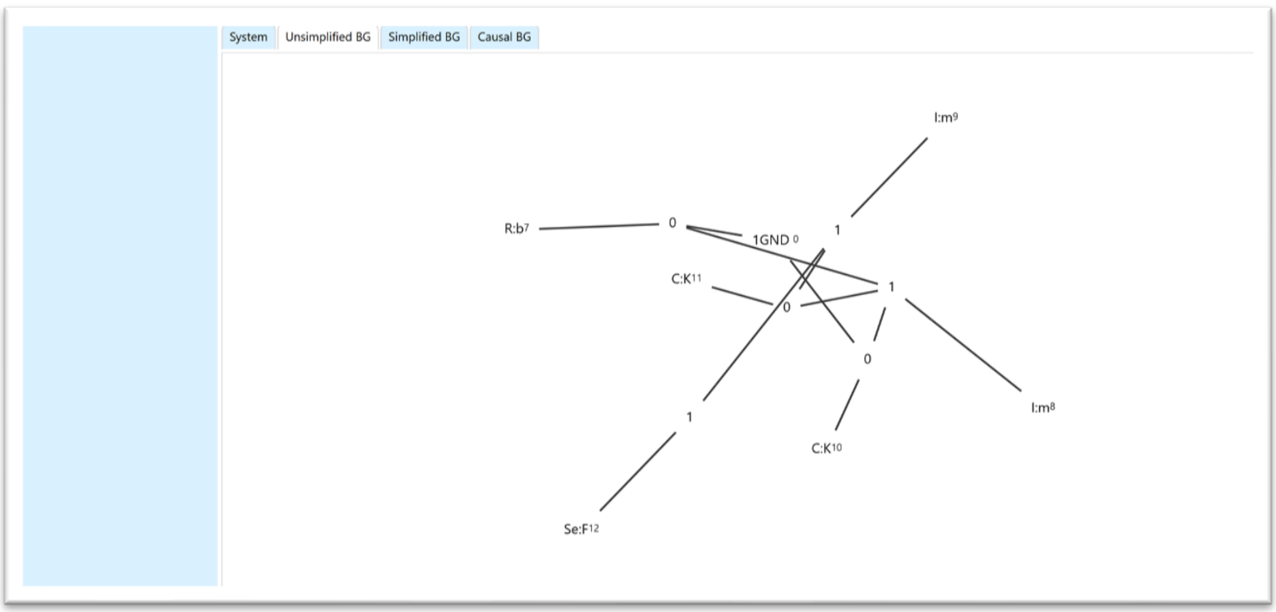

Figure 9: Screenshot of the Unsimplified bond graph generated by the application.

Upon clicking the *Simplified BG* tab, the bond graph will be displayed (requires manual reordering) as shown in Figure 11. This Similar to step 2 shown in Figure 2. If you compare Figures 10 and 11, you will notice that the *Simplified BG* tab showcases the bond direction after simplification. From a student perspective, this will be helpful to check if their process is correct. The *Causal BG* tab will display the bond graph with causalities assigned and state variables displayed on the left as shown in Figure 12. Since the velocity directions have not been implemented in this version of the software, the directions of certain bonds in Figures 11 and 12 are different. That will be resolved in future versions. Figure 12 shows four state variables, p26', p27', x28' and x29'. The states p26' and p27' correspond to the inertia members I:m26 and I:m27 respectively, while x28' and x29' correspond to the stiffness members C:K28 and C:K29. Here, p refers to momentum and p' refers to derivative of momentum (force) while x refers to the displacement and x' refers to the velocity (derivative of displacement). This will help students verify their causality assignment and state variables.

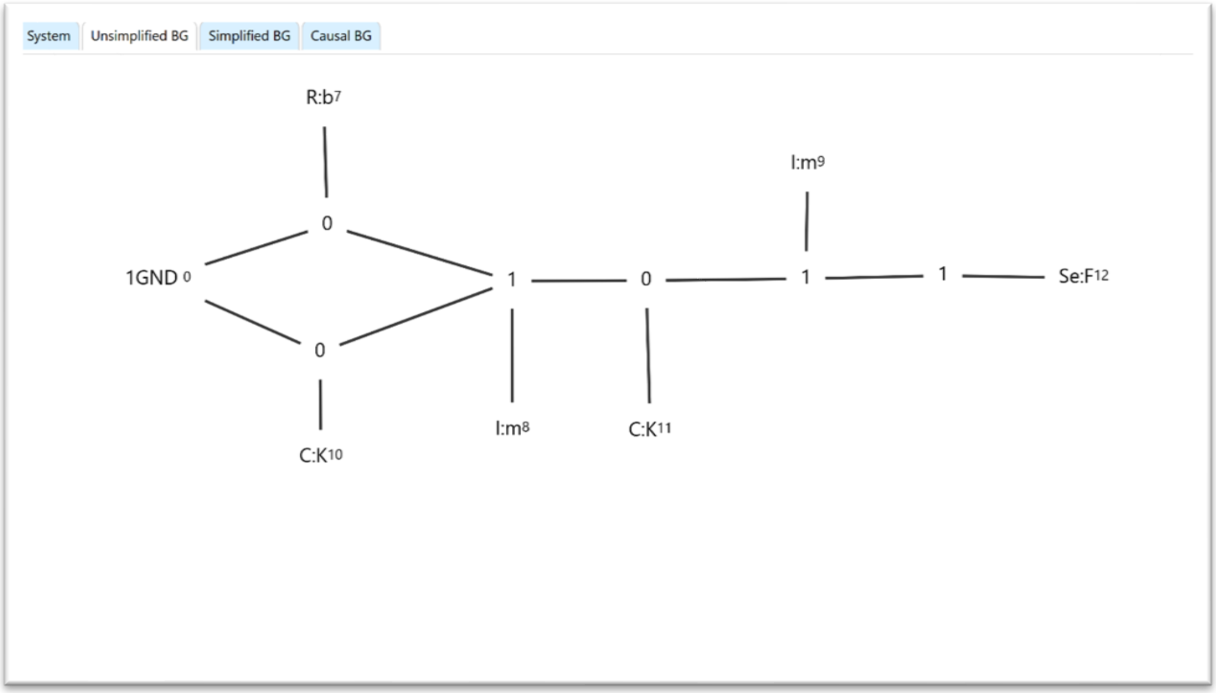

Figure 10: Screenshot of the Unsimplified bond graph that has been manually reorganized.

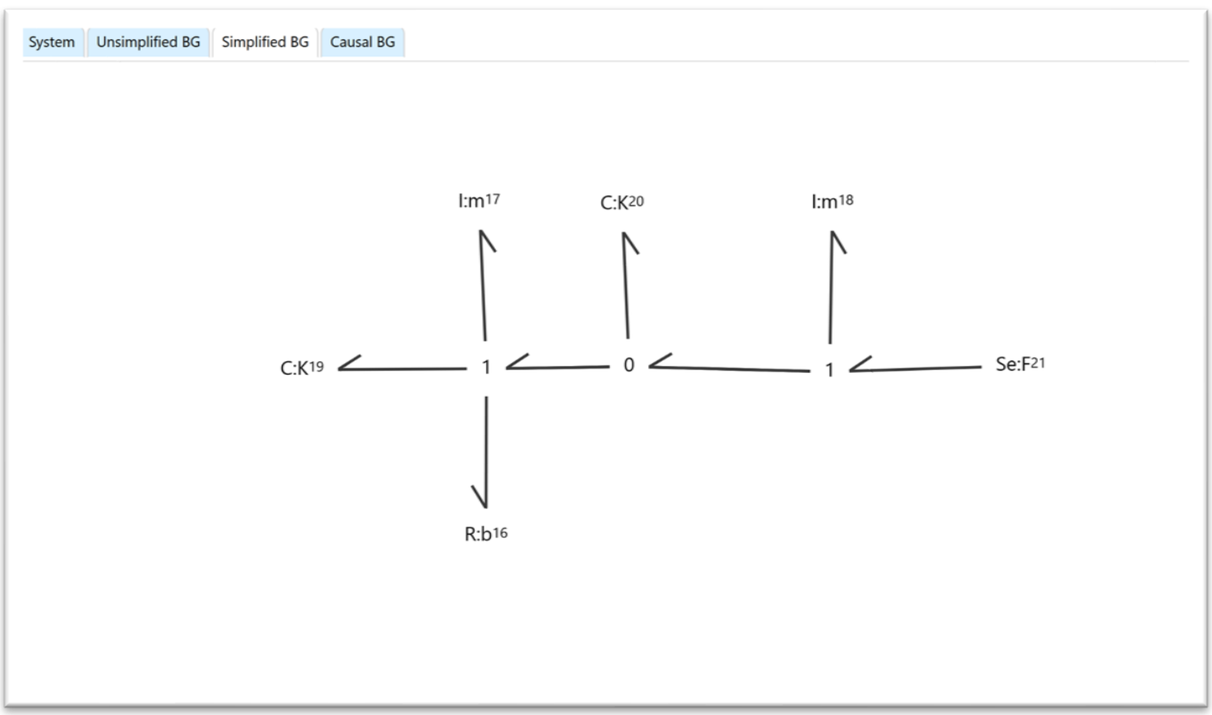

Figure 11: Screenshot of the Simplified bond graph.

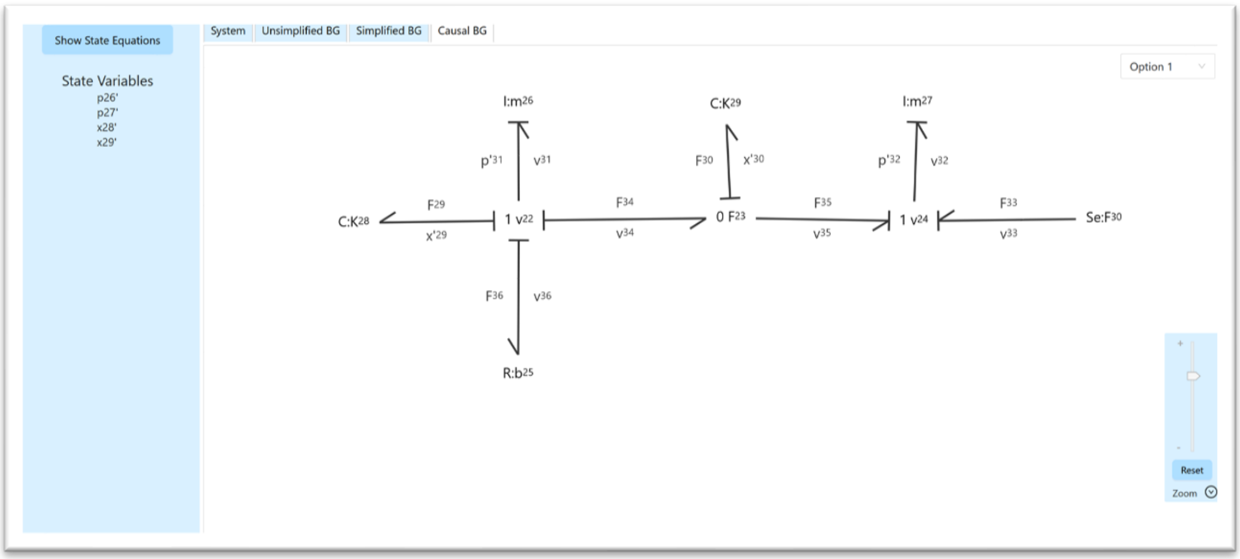

Figure 12: Screenshot of the causal bond graph generated by the program with state variables displayed on the left panel.

On the top left corner (when the *Causal BG* tab is selected), the *Show State Equations* button can be clicked to reveal the differential equations corresponding to each state variable as shown in Figure 13. In the current version, only the final differential equations are displayed. The intermediate steps will be available in future versions. The students will be able to verify their differential equations with that of the software.

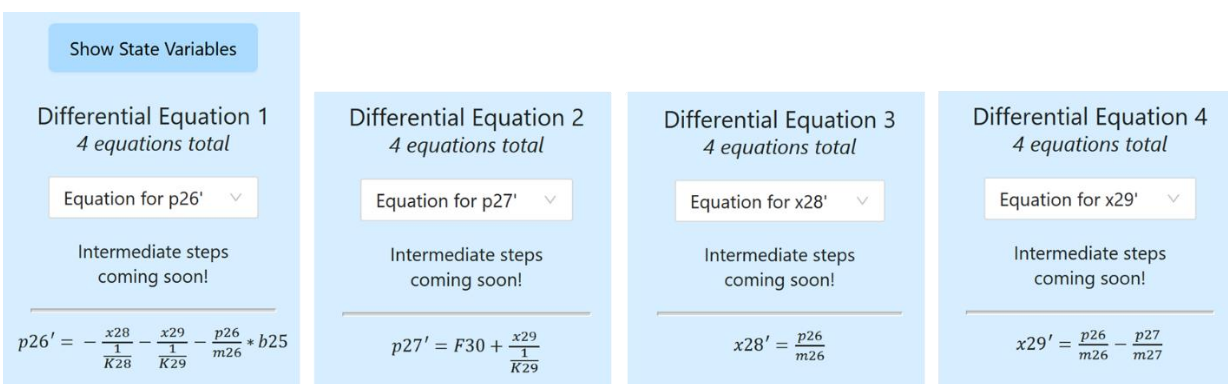

Figure 13: Screenshot of the four state equations generated by the bond graph program.

The software can generate bond graphs of systems in mechanical translation, mechanical rotation, and electrical systems. State equations (differential equations) can be generated for systems that do not involve derivative causality, which occurs in transmission systems. Future versions will include support for those systems as well.

4. User Testing

Surveys were developed for use in a course that taught bond graph modeling technique to collect user feedback about the usability of the software. The first survey evaluated the effectiveness of the tutorial, drag and drop functionality for creating system diagrams, and file sharing capabilities. This survey was conducted during Spring 2023 in the Modeling and Analysis of Mechatronics Systems course (ME/RBE 4322) at WPI. The mechanical system shown in Figure 14 was used in the survey. Students were given a brief 10-minute demonstration of the software, and they had approximately 40 minutes to complete the survey. Data was collected from 35 students primarily through multiple-choice and short answer responses. Students were asked to rate their satisfaction with individual software features as “extremely dissatisfied,” “somewhat dissatisfied,” “neither satisfied nor dissatisfied,” “somewhat satisfied,” or “extremely satisfied.” To better compare these scores, a scale of zero to five was used.

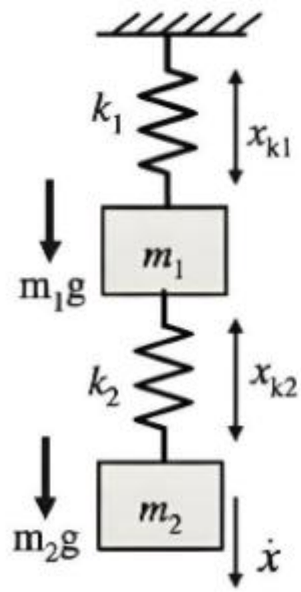

Figure 14: The mechanical system used in the survey.

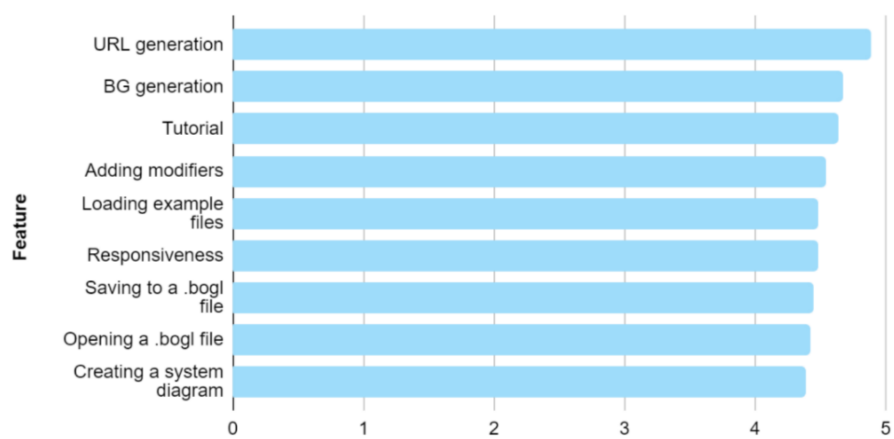

Figure 15: The average satisfaction rating for different features.

On average, each student spent 4.38 minutes making a system diagram and generating the respective bond graph. Since bond graph generation occurs with only the click of a button, the authors assume most of this time is spent making the system diagram. Eighty-one percent of users said they would be able to easily replicate any system diagram they were asked to construct. Users were asked to rate the difficulty of system diagram creation as "very easy," "somewhat easy," "neither easy nor difficult," "somewhat difficult," or "difficult." To best analyze these ratings, each of them were labeled as 1, 2, 3, 4, and 5 respectively and the average of all such rankings were found. Overall, the average difficulty rating was 1.47, indicating that user experience with system diagram creation was split between "easy" and "very easy." Students were asked to rate their satisfaction with different individual features as "extremely dissatisfied," "somewhat dissatisfied," "neither satisfied nor dissatisfied," "somewhat satisfied," or "extremely satisfied." The overall scores are shown in Figure 15. Over 91% of the users were satisfied with save-related features and 93% were satisfied with creating the system diagram. All other features earned positive reactions from at least 95% of users. URL generation had the highest satisfaction rate, with favorable reception from 100% of users.

Many students requested improvements to the application such as allowing labeling, state equation generation, as well as improvements to the modifiers in the electrical domain. There were also some bugs that were discovered by students related to open/save functionality, and bond graph generation involving gravity while using the software in Firefox and Brave browsers. Students overall had a positive reaction with many wishing they had been able to use it earlier in the course.

Some of these issues were fixed and features implemented before conducting surveys again with students in the same course during Fall 2023 and Spring 2024. A total of 81 students participated in these surveys. Over 72% and 18% of the users respectively found the tutorial very easily or somewhat easily. About 58% of the students took less than a minute to create the system diagram while around 38% of the students took anywhere between one and five minutes. Students were also asked to verify the correctness of the results of the program. The software was tested to be correct by the authors. This question was also intended to evaluate student comprehension of the topic. Around 90% of the students were able to identify that the software generated correct causal bond graphs. When asked if the variables displayed on the bonds and states were correct, over 90% identified them to be correct as well. Around 25% identified software bugs that prevented the effort and flow variables from being correctly displayed as discussed during lectures. Students were also asked to check the correctness of the generated state (differential equations). Around 70% of students deduced that the equations generated by the program were indeed correct.

This second round of surveys also collected suggestions from students. The following were some of the requested features: creating bond graphs manually; laying out the bond graphs on a grid; solving systems with derivative causality; specifying coordinate axes and velocity directions.

5. Conclusion and Future Work

The paper describes the development and the testing of a browser-based software tool to help students with bond graph modeling. Numerous features were added to the web application including a full undo/redo system, a system for exporting system diagrams using URLs, an interactive tutorial, and code that generates differential equations from bond graphs. The tool mimics the method followed in lectures and textbooks related to the bond graph technique. These features would be beneficial as a classroom assistance tool in courses where the bond graph technique is taught. When assessed for the tool's effectiveness, an overwhelming number of students had a positive opinion with many wishing they had been able to use it earlier in the course. This feedback is testament to the tool's usefulness and the authors are convinced that this would not only be the case at WPI but also at other universities.

Future additions include specifying coordinate axes, velocity directions, creating bond graphs manually, as well as generating equations from bond graphs with derivative causality. Once the additional features are integrated, the application is proposed to be used alongside the Modeling and Analysis of Mechatronic Systems course at WPI. This will only help students with various assignments but will also hopefully raise their confidence in utilizing the method for other capstone projects as well. This claim will be evaluated in future surveys once the tool is a regular feature in the course. Besides, the authors will promote the software with faculty members in other universities that teach the bond graph technique.

6. Acknowledgements

The authors would like to acknowledge the efforts of Talal Jaber, Corrin Courville and Terry Hearst in developing the BoGL Desktop application.

7. References

[1] R. Merzouki, A. K. Samantaray, P. M. Pathak, and B. Ould Bouamama, *Intelligent Mechatronic Systems: Modeling, Control and Diagnosis*. London: Springer, 2013. doi: 10.1007/978-1-4471-4628-5.

[2] Broenink, J. (1999). *Introduction to Physical Systems Modelling with Bond Graphs*. University of Twente, Dept EE, Control Laboratory.

[3] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*. John Wiley & Sons, 2012.

[4] D. Grande, F. Mancini, and P. Radhakrishnan, "An Automated Graph Grammar Based Tool to Automatically Generate System Bond Graphs for Dynamic Analysis," in *IDETC-CIE2016*, Volume 1B: 36th Computers and Information in Engineering Conference, Aug. 2016. doi: 10.1115/DETC2016-59941.

[5] F. Mancini, D. Grande, and P. Radhakrishnan, "An Automated Virtual Lab for Bond Graph Based Dynamics Modeling Using Graph Grammars and Tree Search," in *IMECE2016*, Volume 5: Education and Globalization, Nov. 2016. doi: 10.1115/IMECE2016-66110.

[6] T. A. Jaber, C. M. Courville, and H. T. Hearst, "BoGL: An Application for Generating Bond Graphs," Undergraduate Major Qualifying Project, Worcester Polytechnic Institute, May 2020. [Online]. Available: https://digital.wpi.edu/show/n583xx306

[7] G. D. Battista, *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.