



# WPI

# Improving the User Experience and Features of PMKS+:

## A Web-Based Linkage Analysis Tool for Education

**Kohmei Kadoya**

A thesis submitted in partial fulfillment  
for the degree of Master of Science in

Robotics Engineering

**May 2023**

**Advisor:**

Professor Pradeep Radhakrishnan

**Committee:**

Professor Alireza Ebadi

Professor David C. Brown

Professor Nicholas Bertozzi

## **Abstract**

Planar Mechanism Kinematic Simulator Plus (PMKS+) is a web-based tool for the analysis of planar linkages with revolute and grounded prismatic joints. Developed as a classroom and assignment assistance tool with many features and capabilities, PMKS+ is used in different Mechanical and Robotics Engineering courses at WPI.

The objectives of this thesis were to critically assess and enhance the application's user interface so that its built-in features are easily accessible and intuitive for users, thereby expanding the user-base of PMKS+. Comprehensive studies involving the application's primary users were conducted and journey maps were developed to discover key pain points. One key issue identified was the difficulty in accurately recreating the linkage users wanted to analyze. If the linkage is specified in the form of joint coordinates, then it could be easily created within PMKS+ but not when the linkage is specified in terms of link lengths and angles. Users also had trouble locating features such as plots and equations that could be used to verify or troubleshoot errors in different assignments. Besides, the existing library of link shapes had limited adjustability.

To address these concerns, a complete redesign of the application was undertaken, with a primary focus on improving the user experience for the existing feature set in PMKS+. The user interface was modernized by following modern material design guidelines. General conventions were extracted from other engineering applications used by our users to allow an easier transition to PMKS+. High-fidelity interface designs were generated and rigorously reviewed by advisors and users at various stages of the development process. In conjunction with front-end improvements, significant backend modifications were implemented to decouple analyses modules from user-interface components. Major features introduced include specifying linkages using link lengths and angles, three position synthesis, and an innovative method for altering link shapes through weldments.

Working prototypes of the redesigned application were evaluated by different groups of students at various stages. The results revealed a significant improvement in user experience with a high satisfaction score.

## **Acknowledgements**

I would first like to express my gratitude for the guidance and encouragement provided by my academic advisor, Professor Pradeep Radhakrishnan, Associate Teaching Professor in Mechanical Engineering and Robotics at WPI.

I would like to acknowledge the invaluable contributions of several individuals: Professor David C. Brown, Emeritus Professor in the Computer Science Department, for his feedback on the UI/UX; Professor Nicholas Bertozzi, Senior Instructor in Robotics Engineering, for integrating PMKS+ in his Unified Robotics I course and his feedback; and Professor Alireza Ebadi, Assistant Teaching Professor in Mechanical & Materials Engineering, for his input and also for integrating PMKS+ into his Mechanisms course next fall.

My appreciation also goes to Alex Galvan for his extensive work on PMKS+ and his assistance in acclimating me to the project while pursuing his Ph.D. I am grateful for Ansel Chang and Jacob Adamsky, who helped develop parts of the program and will continue contributing to PMKS+ in the future. Finally, I would like to thank all previous contributors to PMKS+/PMKS and to the many students who offered their feedback this year to improve PMKS+.

# Contents

<b>List of Figures</b>	<b>5</b>
<b>List of Tables</b>	<b>13</b>
<b>1 Introduction</b>	<b>14</b>
1.1 Brief overview of PMKS+ and its purpose . . . . .	14
1.2 Objective of the thesis . . . . .	14
<b>2 Background</b>	<b>15</b>
2.1 Description of PMKS+ and its features . . . . .	15
2.2 Previous Work on PMKS+ . . . . .	15
2.3 Application Ideals and Philosophy: . . . . .	17
2.4 Current (2022) Version of PMKS+ before the redesign . . . . .	18
<b>3 User Studies and Research</b>	<b>27</b>
3.1 Methodology for Conducting User Studies . . . . .	27
3.1.1 User Studies from Previous Years . . . . .	27
3.1.2 User Studies during 2022-23 . . . . .	28
3.2 User Demographics . . . . .	28
3.3 User Journey Maps . . . . .	30
3.4 Other Stakeholders . . . . .	32
3.5 Pain Points Identified . . . . .	33
3.6 User Needs . . . . .	38
3.7 Competitive Analysis of Existing Linkage Analysis Software . . . . .	38
<b>4 Redesign Process</b>	<b>42</b>
4.1 Redesign Goals . . . . .	42
4.2 Low and High Fidelity Prototyping . . . . .	43
4.2.1 Adherence to Google’s Material Design guidelines . . . . .	45
4.2.2 Incorporation of conventions from other engineering applications . . . . .	47
4.2.3 Toolbar and File Options . . . . .	48
4.2.4 Tab Designs . . . . .	50
4.2.5 Edit Panel Design . . . . .	54
4.2.6 Analyze Panel Design . . . . .	56
4.2.7 Synthesize Panel Design . . . . .	58

4.2.8	Infinately Customizable Shapes . . . . .	59
4.2.9	Color and Branding . . . . .	65
4.3	Early Feedback from Users . . . . .	67
4.4	Simple Prototype Testing . . . . .	69
4.5	Feedback from Other Stakeholders . . . . .	72
<b>5</b>	<b>Final Application</b>	<b>73</b>
5.1	Overview of Redesigned Application . . . . .	73
5.2	Solutions and Features . . . . .	76
5.2.1	Click-to-select Objects . . . . .	76
5.2.2	3 Tab Navigation . . . . .	77
5.2.3	Analysis Overhaul . . . . .	84
5.2.4	Infinately Customizable shapes . . . . .	85
5.2.5	Dynamic Grid . . . . .	88
5.2.6	Synthesis . . . . .	91
5.2.7	Responsiveness and Material Design . . . . .	91
5.2.8	Other Quality of Life Improvements . . . . .	92
5.3	Backend and Software Engineering Optimizations . . . . .	96
5.3.1	Interesting Algorithms . . . . .	99
5.4	Validation Testing . . . . .	102
5.4.1	User Testing . . . . .	102
5.4.2	Analytics and Feedback . . . . .	103
5.4.3	Accessibility Testing . . . . .	104
<b>6</b>	<b>Future Work and Development Plans</b>	<b>106</b>
6.1	Upcoming Features . . . . .	106
6.2	Awareness and Advertising . . . . .	112
6.3	Future Development Timeline . . . . .	112
<b>7</b>	<b>Conclusion</b>	<b>115</b>
<b>8</b>	<b>References</b>	<b>116</b>
<b>9</b>	<b>Appendix</b>	<b>117</b>
<b>10</b>	<b>Vita</b>	<b>118</b>

## List of Figures

1	Two Major Redesigns have been completed for PMKS. (Left) PMKS Silverlight (2010) (Right) PMKS+ (2019) . . . . .	16
2	The original application has four main sections: The Grid, Toolbar, Linkage Table, and Animation Bar . . . . .	18
3	The <i>Animation Bar</i> contains three main groups of features: Linkage Information, Simulation Controls, and Grid Controls . . . . .	19
4	The <i>Linkage Table</i> is split between three tabs, <i>Joints</i> , <i>Links</i> , and <i>Forces</i> with a row for each object . . . . .	19
5	The <i>File</i> Tab on the Toolbar contains three buttons: Open, Save, and Copy . . .	20
6	The <i>Settings</i> Tab on the Toolbar contains four features: Input Speed, Input Direction, A Gravity Toggle, and unit options. . . . .	21
7	The <i>Help</i> Tab on the toolbar contains three buttons: About, Section Info, and Templates . . . . .	21
8	The <i>Analysis</i> tab in the toolbar allows four types of analysis to be performed. . .	22
9	The analysis popup window with the three tabs: Export, Plot Charts, and Show Eqns. . . . .	22
10	<i>Force Analysis</i> Popup enables data exporting, graph generation, and options to define the positive force directions at each joint in a FBD. . . . .	23
11	The <i>Kinematic Analysis</i> Tab allows for Excel export and graph generation, as well as the generation of loop equations needed for kinematic analysis. . . . .	24
12	The right-click joint context menu offers options to delete the joint and toggle Input, Ground, and Slider. . . . .	24
13	The right-click force context menu allows users to delete the force, toggle the reference frame, and change its direction. . . . .	25
14	The right-click link context menu allows users to delete the link, edit the link shape, and add a force to the link. . . . .	25
15	Clicking <i>Edit Shape</i> opens the shape editing panel and puts PMKS+ into a shape editing mode. . . . .	26
16	The user must select one shape from the 11 shapes available in the shapes library. . . . .	26
17	A timeline of the most relevant user studies conducted since PMKS+'s inception in 2019. The studies performed as a part of this thesis are shown with an oval.	27
18	Over 3/4th of the users use PMKS+ using Google Chrome on Windows. . . . .	29

19	Over half of the users have more than 3 hours of experience with PMKS+ . . . .	29
20	Almost all users have experience with some type of engineering program. . . .	30
21	Persona 1 Struggles with finding discrepancies between their Matlab and PMKS+ results for their homework. . . . .	31
22	Persona 2 cannot adequately prototype a linkage for their final project inside PMKS+ . . . . .	32
23	Series of images showing the process to open the graph showing the velocity of a link. From left to right and top to bottom: (a,b) Select the kinematics button in the analysis tab. (c) In the popup, select the <i>Plot Charts</i> tab. (d-g) In the 3 dropdowns, select Kinematic Analysis: <i>Loop</i> , Kinematic Property: <i>Linear Link's COM Vel</i> , and Linear Link: <i>bc</i> . . . . .	34
24	Linkage table gets extremely long even with only these two links . . . . .	35
25	The only option users have to edit the length of a binary link is to enter the x and y coordinates of each joint. . . . .	36
26	A sequence of images showing the process of editing a link shape. From left to right, then top to bottom. (a,b) Right-click on a link and select <i>Edit Shape</i> in the context menu. (c) Select the new shape from the scrollable right panel showing a list of available shapes. (d-f) Use the bounding box and rotation icon to translate, rotate, and scale the new shape, making sure to include all joints within the new shape. (g) Click <i>save</i> on the shape panel to confirm the shape change. . . .	36
27	A similar four-bar linkage created at three different scales. The smallest linkage (5cm) makes the UI elements too large relative to the linkage and the largest (100cm) has the opposite problem. . . . .	37
28	A collection of images showing representative views of each application discussed in the table. This provides a general idea of each software's appearance and user experience. . . . .	40
29	Mockup one introduces the idea of clicking an object to select it. The right panel shows settings (position, grounded?) or analysis graphs for the selected object.	43
30	Mockup two leans into the idea of the three modes (Synthesis, Edit, Analyze) being a linear process that a user goes through. Users can jump to any mode by selecting the labels on the top or by using the arrow buttons on the bottom.	44
31	A flow diagram of PMKS with modes in the solid blue boxes: Synthesis, Edit, and Analyze. Arrows show the movement of users between modes along with the most common thought process for why they might move between them. .	45

32	Mockup three integrates standard Material Design elements for a consistent appearance. The top right contains 3 modes, synthesis, analysis, and settings (edit) where each panel changes based on what object is selected. . . . .	46
33	(Left) A survey of our users shows that they are most familiar with Solidworks and Working Model. (Center) Working Model puts its primary interactions on the top and left. (Right) Solidworks also puts its tree and action on the left and top but also has a dedicated settings button. . . . .	47
34	Mockup four moves the context-depended panel to the left while keeping static secondary panels like settings and equations on the right. The edit and analysis information are now stored in the Analyze/Edit panel on the left. . . . .	48
35	A list of actions located in the toolbar before the redesign. Yellow items can be moved to the settings/help panel while exporting data will get moved into the analysis panel. . . . .	49
36	The first toolbar mockup uses a <i>File</i> button on the top left which, when clicked on, shows a list of file-related actions. . . . .	49
37	The second toolbar mockup flattens the menu by putting all 5 icons on the toolbar. A hover shows the label for the icon along with the shortcut. . . . .	50
38	The last toolbar mockup shows both an icon and label in a thin toolbar with buttons justified to the left. . . . .	50
39	The first left panel design makes the entire panel hideable by clicking on the blue arrow. Upon clicking the blue arrow, the tab drawer opens from the left. Hovering over the right edge reveals a blue arrow pointing left, allowing the panel to be closed. . . . .	51
40	The second left panel design converts the three modes into large icons on the left. In descending order, the first icon represents the synthesis tab, the second represents the edit tab, and the third represents the analysis tab. . . . .	52
41	The third and final left tab design expands the three icons into tabs that take up the entire vertical space on the left. When the mode is selected the vertical 'tab' button connects to the panel that is opened and changes color. . . . .	53
42	Several input blocks and minor variations of 'blocks' were created as a modular way to build interfaces. From top to bottom then left to right: Title Block, Subtitle Block, Action Block, Toggle Block, Input Block, Dual Input Block, Radio Block, Color Block . . . . .	54
43	(Left) Edit Panel when a Joint / Tracer is selected. (Center) When a Link is selected. (Right) When a force is selected. . . . .	55



44	(Left) Analysis Panel when a Joint / Tracer is selected. (Center) When a Link is selected. (Right) When a force is selected. . . . .	57
45	From Left to Right, then Top to Bottom: (a) Entering synthesis mode, a drag- gable coupler appears. (b) The first coupler is positioned correctly, and the <i>Next Pose</i> button is pressed. (c,d) Position the other two poses until all three positions are defined. (e) Click <i>Generate</i> to create a four-bar. (f) Toggle the construction lines to view the perpendicular bisectors. . . . .	59
46	Left to Right, then Top to Bottom. (a-c) By default, links are in convex hull mode, so when a third tracer is created in the red binary link and dragged up, the red link becomes a triangle. (d) Clicking <i>Toggle Shape</i> puts the link into skeleton mode. (e) Adding another tracer results in another line appearing perpendicular to the primary axis. (f) Clicking <i>Toggle Shape</i> returns the link to convex hull mode where the four joints define the outer shape of the link. . . . .	61
47	When a link has two joints, the <i>Toggle Shape</i> button cycles the shape between a binary link, a circle centered at the first joint, and a circle centered at the second joint. . . . .	62
48	Left to Right, then Top to Bottom. (a-c) Similar to Figure 46, a tracer is added to a binary link and dragged up to create a triangle (a convex hull of the three joints). (d) When the <i>Edit Shape</i> button is pressed, the application switches into a new mode where lines with checkboxes appear over the outer edges of the shape. (e) Deselecting the bottom edge removes that edge. (f) Clicking <i>Save</i> saves the inverted V shape as the new link shape. (g) Starting at step d, if the top-left edge is deselected, some edges are now required to be selected (greyed-out check- mark) to keep all joints connected. (h) When the top-right side is deselected, an inverted T shape remains. (i) Clicking <i>Save</i> saves that shape. . . . .	63
49	From Left to Right then Top to Bottom. (a) Starting with a blue binary link and a straight 3-joint link, right-click on the shared joint. (b) Click <i>Weld Joint</i> on the con- text menu. (c) This combines the links into one shared T-shaped link. (d) When the center joint is dragged up, the shape changes as if the two base shapes are being dragged. However, it still appears as one red link with internal corners welded. (e) Right-click on the center joint and click <i>Unweld Joint</i> . (f) The two shapes are left with the blue binary link and a red triangular link. . . . .	64
50	The current app's color palette uses Ruby Blue #74AEEE and Spring Green #75FBA1	65
51	Four different color schemes of the edit panel derived from the base PMKS+ colors. . . . .	65

52	The redesigned color palette has a primary color of #3F51B5 and an accent color of #FFC107. . . . .	66
53	PMKS+'s Primary font is Roboto, a modern sans-serif font. . . . .	67
54	A study in October 2022 asking 16 participants to compare the current PMKS+ app with the mockup shows an overwhelming preference for the mockup. . . .	68
55	A curated list of written feedback received from the October 2022 study asking students to compare the current PMKS+ app with the mockup. A full list of feedback can be found in the Appendix. . . . .	69
56	An assembled four-bar in the current version of PMKS+ before the redesign. . .	70
57	The same four-bar in the new prototype version of PMKS+. . . . .	70
58	A study in December 2022 asked 30 students to compare the current application with the redesigned prototype. For editing, there was a very slight preference for the prototype. For analysis and synthesis, there was an overwhelming preference for the prototype. . . . .	71
59	A curated list of written feedback received from the December 2022 study asking students to compare the current PMKS+ app with a prototype of the redesign. A full list of feedback can be found in the Appendix. . . . .	72
60	The redesigned application has a familiar interface with the grid as the background. There is a toolbar on top and an animation bar on the bottom. Instead of a linkage table there is now three tabs on the left. . . . .	73
61	The three tabs on the left opens the left panel and the buttons on the right of the toolbar open the right panel. In this image, the analysis panel is open on the left and the settings panel is open on the right. . . . .	74
62	The New Animation bar contains 3 main groups of features: Playback Controls, Time Controls, and Grid Controls . . . . .	75
63	The New Toolbar has two sets of buttons. The left side contains file related actions and the right buttons open a right panel. . . . .	75
64	(Left) A binary link with both joints in the default state. (Right) A binary link with a joint selected. . . . .	76
65	(Left) A 3 joint triangle link in its default state. (Right) When the same link is selected, there is a yellow outline. . . . .	76
66	(Left) A compound link in its default state. (Right) When the same link is selected, a yellow outline appears and the outline of the links that make up the compound link is shown. . . . .	77

67	(Left) A force in its default (unselected) state. (Right) A selected force shows two draggable points at the start and end of the force. . . . .	77
68	When no object is selected on the grid, the edit and analyze panel show an interactive help panel. The synthesis panel always shows the four-bar generator.	78
69	When a joint is the selected on the grid, the edit panel shows several options to edit the location, and whether it's grounded or an input. When the analysis tab is selected, graphs for the position, velocity, and acceleration of the joint are shown along with forces at the joint and an input torque. . . . .	79
70	When a link is selected on the grid, the edit panel shows options to edit the length(s) and angle of the link along with a list of related joints and forces. In the analysis tab, there are kinematic properties of the link, the COM of the link, and force analysis graphs for links. . . . .	80
71	When a Force is selected, the edit panel shows options to edit the magnitude, angle, components, and reference frame. There are no analysis options for the force. . . . .	82
72	PMKS+ with the analyze and equations panel open at the same time. The analysis graphs, the linkage with labels, and the equations panel are all visible at simultaneously. . . . .	83
73	The new analysis panel allow multiple graphs to be open at the same time. There is a scrolling cursor on the graph that is linked to the simulation time. The mechanism is visible alongside the graphs allowing for better association.	84
74	A series of images showing the creation of a triangular link from a binary link. From left to right: (a) Right click on the link to bring up the context menu. (b) Click <i>Add Tracer Point</i> . (c,d) Drag the tracer point up to create the shape defined by the convex hull of the 3 points. . . . .	85
75	Other types of shapes that can be created by adding more tracer points. From left to right, there are 3, 4, 5, and 6 joints that make each link. . . . .	85
76	Two binary links can be welded together by clicking <i>Weld Joint</i> on the joint shared by them. . . . .	86
77	Three binary links can be welded together by clicking <i>Weld Joint</i> on the joint shared by them. . . . .	86
78	A <i>C</i> shape can be created by welding together 3 binary links. . . . .	87
79	A <i>T</i> shape can be created by welding together a binary link with a linear 3 joint link. . . . .	87
80	Multiple polygons can be welded together to create a compound link. . . . .	87

81	Links that are intersecting can be combined to create complex welded shapes.	87
82	Any number of links can be welded together to create compound shapes. . . .	88
83	The new grid has dynamically changing labels along with grid lines the appear to further subdivide the grid when the user zooms in. . . . .	89
84	The bottom of the <i>settings panel</i> has options to toggle the visibility of the grid, the minor grid lines, and the size of the objects on the grid. . . . .	89
85	A collection of four-bars created in the new PMKS+ with different sizes. (Left) A four bar with a coupler about 2cm. (Center) A similar four-bar where the coupler is about 5cm. (Right) The coupler is about 100cm. . . . .	90
86	A collection of the same slider crank with different object scales. (Left) Object scale is 0.5. (Center) Object scale is 1, which is the default. (Right) Object scale is 2. . . . .	91
87	(Left) The help panel shown in the analysis tab when the linkage is not valid. (Right) The help panel on the right. There is a are buttons to access tutorials and form to send feedback to the team. . . . .	93
88	An image of the new PMKS showing many possible help tooltips that are shown when the mouse is hovered over a button for a second. . . . .	94
89	An example of a four-bar linkage where all links and joints are given a custom name. Link AB is in the process of being renamed to Crank. . . . .	95
90	A grid of some of the icons designed specifically for PMKS+ to give a better visual indication of what a button will do. . . . .	95
91	The new graphics uses to represents grounded and input joints, prismatic joints, and forces. . . . .	96
92	A simplified code snippet illustrating the declarative input block components .	98
93	An Issue Tracker board showing a list of known issues and tasks assigned to team members . . . . .	99
94	A series of images showing the backend process for computing the SVG path for a simple shape. This process occurs whenever a position of any of the link's joint's position changes. . . . .	100
95	A series of images showing the process of computing the SVG path for a compound link. Similar to simple shapes, this computation is done every time any of the joint positions change. . . . .	101
96	A flowchart showing a high-level view of how the dynamic grid lines and labels get updated. . . . .	102

97	Results of the April 2023 study of five students showing that all students discovered and utilized the core functionalities of the new application without any prior experience or instruction. . . . .	103
98	A simulated view of PMKS+ through three colorblind filters: Green-Weak Deuteranomaly, Blue-Blind Tritanopia, and Monochromacy / Achromatopsia . . . . .	105
99	A redesigned linkage library popup that has an animated preview for each linkage	107
100	A welcome popup to guide the user on how to get started . . . . .	108
101	The current warning popup when accessing PMKS+ through a touchscreen interface such as a tablet . . . . .	109
102	A potential interface to show the loop equations and equations used to generate graphs . . . . .	110
103	Mockups for editing force moment references and positive force directions . . .	111
104	A suggested plan for the next few months of development for PMKS+. The next key milestone is the public release in August 2023. . . . .	113

## List of Tables

- 1 Table comparing key features in currently available linkage analysis software. Force analysis is considered only when the entire link is analyzed. Synthesis options include position-based and path-based synthesis. Quick is subjective and represents if an average user can create a linkage in less than 10 minutes without a tutorial. . . . . 41

# **1 Introduction**

In the rapidly advancing world of technology, it is crucial for an application to be not only functional but also user-friendly. By adhering to user-centered design principles, applications can be refined to provide a more positive experience for users, addressing their needs and challenges. The Planar Mechanism Kinematic Simulator Plus (PMKS+) exemplifies such an application, having undergone significant enhancements in usability in recent years.

## **1.1 Brief overview of PMKS+ and its purpose**

PMKS+ is a 2D linkage analysis tool designed to assist students in understanding the construction and modeling of planar linkages. It has become increasingly important and relevant in teaching Robotics Engineering (RBE) and Mechanical Engineering (ME) courses at Worcester Polytechnic Institute (WPI).

In addition to modeling and simulating linkages for detailed analysis, PMKS+ has always maintained a focus on education. The tool encourages collaboration among students working on a linkage, and it supports teaching staff by enabling students to check their work independently. This allows staff to concentrate on the crucial task of teaching students the underlying concepts of linkage design and analysis.

## **1.2 Objective of the thesis**

The primary objective of this thesis is to assess and enhance the usability of PMKS+. This involves identifying and addressing the challenges and user needs associated with the application. A redesigned interface, informed by user research and feedback, will be implemented alongside the addition of new features to improve the overall user experience. The effectiveness of these changes will be analyzed to determine their impact on the user experience.

The structure of this thesis is as follows: Chapter 1 serves as an introduction, providing an overview of PMKS+ and outlining the objectives of the thesis. Chapter 2 delves into the background of the application, including its development history and previous improvements. Chapter 3 presents user studies and research that guided the redesign process. Chapter 4 discusses the redesign process in detail, outlining the rationale behind the interface changes and new features. Chapter 5 showcases the final application, highlighting the key improvements and their impact on the user experience. Chapter 6 explores potential future work to further enhance the application, while Chapter 7 concludes the thesis with a summary of the findings and their implications.

## 2 Background

In this section, the prior work on PMKS+ will be discussed, as well as the version of the application from which we began this year. The organization's philosophy in developing this app and its core features will also be described.

### 2.1 Description of PMKS+ and its features

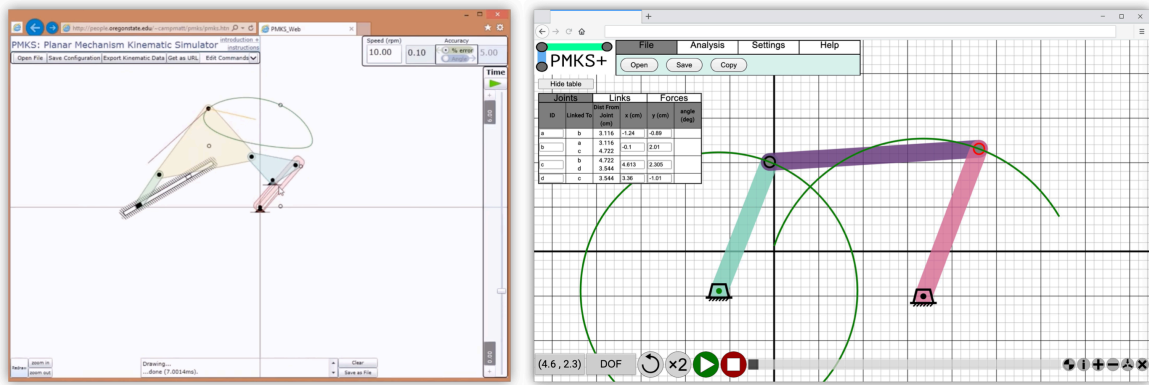
Planar Mechanism Kinematic Simulator Plus (PMKS+) is a versatile and robust tool for designing, simulating, and analyzing planar linkages. Primarily utilized as a teaching tool in engineering courses, the application offers a range of features that enable users to interact with and understand the complexities of planar mechanisms effectively. The main features of PMKS+ include:

- **Ability to create and edit planar linkages:** PMKS+ enables users to design and modify planar mechanisms using an array of tools and options, such as adding and adjusting joints, links, and other components. This feature allows users to create custom linkages and iterate on their designs seamlessly.
- **Performing kinematic and force analyses on joints and links:** The application offers comprehensive analysis tools for evaluating the performance of planar mechanisms. Users can analyze the motion and forces acting on joints and links in real-time, helping them better understand the behavior of their designs and make informed decisions to enhance their mechanisms.
- **Exporting, saving, and sharing linkages with other users:** PMKS+ provides functionality for exporting, saving, and sharing linkage designs in various formats. This allows students to easily save their work and share it with peers and teaching staff.

### 2.2 Previous Work on PMKS+

The Planar Mechanism Kinematic Simulator (PMKS) was first developed as a Microsoft Silverlight application for the simulation and analysis of planar mechanisms. Created by the Design Engineering Lab at Oregon State University, led by Dr. Matthew Campbell, PMKS was initially released in 2010. It served as the foundation for the development of PMKS+, which aimed to recreate the application on a modern web platform with an enhanced user experience and a greater focus on classroom use. [1, 2]





**Figure 1:** Two Major Redesigns have been completed for PMKS. (Left) PMKS Silverlight (2010) (Right) PMKS+ (2019)

Since the inception of PMKS+, multiple teams of students and faculty at Worcester Polytechnic Institute (WPI) have contributed to its development by refining features, enhancing the user experience for creating and editing linkages, and adding powerful analysis capabilities. Several reports have been published documenting the migration from PMKS to PMKS+, as well as the various enhancements and user interface design principles employed throughout the development process. The original PMKS and PMKS+ can be seen in Figure 1.

The 2019 report, "PMKS+: Recreating a Legacy Application," focused on converting the original PMKS into a modern web-based application with a more consistent, user-friendly interface. The report emphasized a well-documented, robust foundation for PMKS+ as an accessible and well-supported TypeScript-based application. UI changes led to improvements in user efficiency and error reduction when creating mechanisms on PMKS+ compared to the original PMKS [3].

In the 2020 report, "PMKS+: A Tool to Analyze and Synthesize Planar Mechanisms," the authors built upon previous work by implementing UX-focused updates to the interactions within the app while maintaining the overall structure. The team completed the migration by incorporating features from PMKS that were not present in PMKS+. They also added functionality to better integrate PMKS+ into the classes it was being used for, solidifying its identity as an education-focused linkage analysis tool. Furthermore, the team thoroughly tested the application on various browsers, identifying and fixing browser-specific bugs [4].

Finally, the 2022 report, "PMKS+: Enhancements," presented additional improvements and new analysis features for PMKS+. The authors focused on integrating various types of analyses within PMKS+, including kinematics and force analyses (static equilibrium and Newton's second law). They updated the user interface to display equations and data and upgraded the

backend development platform. The improved PMKS+ application was tested in numerous courses that teach mechanisms, where students compared the software-generated results with their MATLAB implementations. The availability of the software also helped the teaching team enhance the quality of feedback provided during office hours and help sessions [5].

### 2.3 Application Ideals and Philosophy:

The development of PMKS+ is guided by several core ideals and principles, which help ensure that the application remains a valuable tool for students and educators in the Mechanical and Robotics Engineering fields.

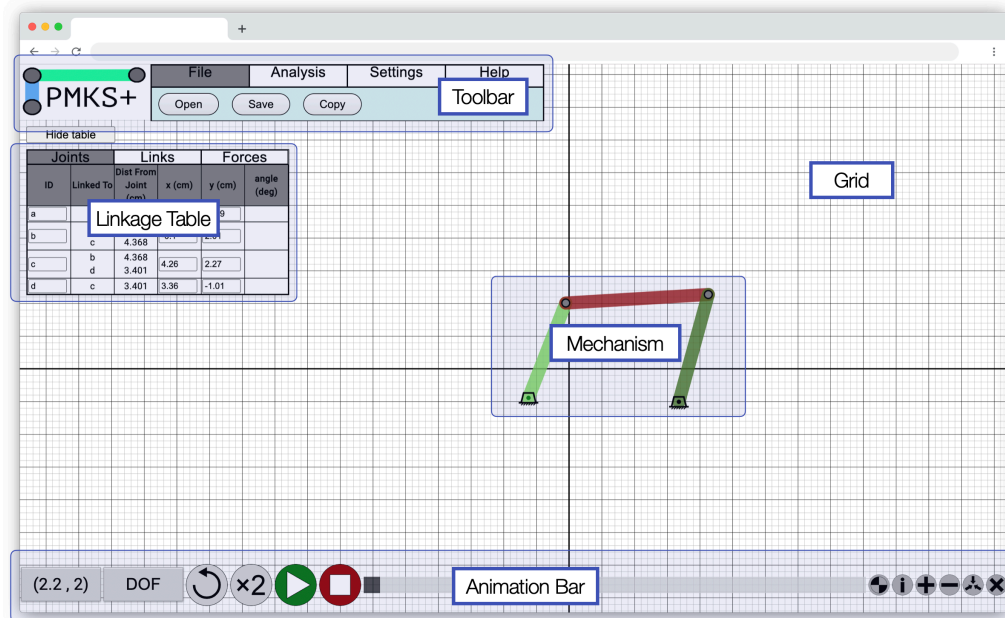
- **Educational:** The primary purpose of PMKS+ is to facilitate learning and understanding of planar mechanism concepts. The software should be easy to use, allowing students to focus on learning concepts and verifying their work. Unlike other professional tools, PMKS+ emphasizes not just the results but also the steps students can take to achieve those results.
- **Flexible:** PMKS+ should be capable of solving a wide range of problems, not just specific assignments or linkage types. The application is designed to support various types of planar mechanisms and provide diverse analysis tools, enabling students to tackle complex problems and explore different design scenarios. This flexibility makes PMKS+ a versatile tool, suitable for use in capstone or final projects.
- **Collaborative:** The application should facilitate sharing and collaboration, enabling groups to work together on a single linkage. PMKS+ allows users to save, export, and share their designs with others, promoting the exchange of ideas and fostering a collaborative learning environment. This feature is particularly valuable in educational settings, where students often work in groups and benefit from discussing their designs with their peers and instructors.
- **Accessible:** PMKS+ should be compatible with multiple operating systems, require no downloads, and be free and open source. The application is developed as a web-based tool, ensuring that it is accessible from various devices and platforms without the need for installation. Additionally, PMKS+ is available at no cost, making it an accessible resource for students everywhere.

In summary, PMKS+ is a powerful and versatile tool for designing, simulating, and analyzing planar mechanisms. The application is guided by the ideals of being educational, flexible, collaborative, and accessible, ensuring that it remains a valuable resource for students

and educators. The development of PMKS+ has been an iterative process, with continuous improvements and enhancements based on user feedback and research. By building upon the successes of the original PMKS and incorporating the lessons learned from previous reports, PMKS+ has evolved into a robust, user-centered application that effectively addresses the needs of students and teaching staff.

## 2.4 Current (2022) Version of PMKS+ before the redesign

The current version of PMKS+ (shown in Figure 2) consists of four main components: Toolbar, Grid, Linkage Table, and Animation Bar. The Grid is used to create mechanisms, the Animation Bar sets the animation and displays helpful information, the Linkage Table shows properties of joints, links, and forces, and the Toolbar offers access to resources and settings within PMKS+ [5].<sup>1</sup>



**Figure 2:** The original application has four main sections: The Grid, Toolbar, Linkage Table, and Animation Bar

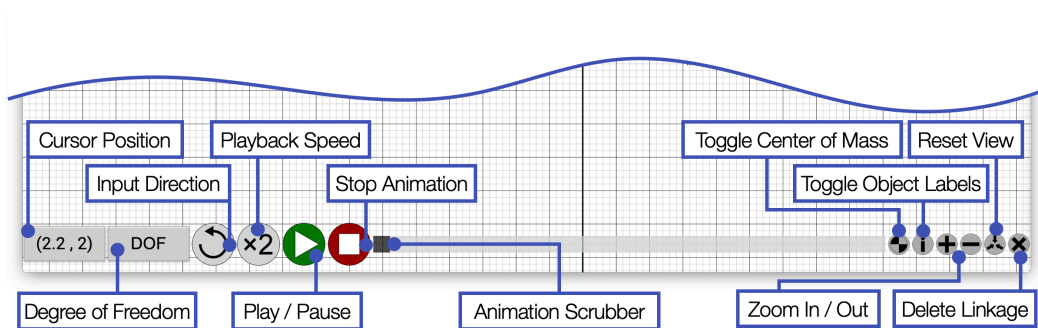
The Animation Bar (shown in Figure 3) has four sections: Helpful Info, Animation Scrubber, Playback for Animation, and Buttons Adjusting the Grid. Users can see cursor coordinates, the

<sup>1</sup>Some of these images are from the PMKS+ version provided when I started working on this project. Some of these images may appear differently than what is currently published on pmksplus.mech.website.

current degrees of freedom (DoF), and know when they have correctly created a fully defined mechanism.

Three buttons adjust the mechanism's animation: playback speed, play/pause, and stop. The animation slider updates based on the mechanism's animation state, and users can adjust joints, links, and forces properties by dragging the animation scrubber.

Six buttons on the right of the animation scrubber adjust the grid, such as toggling the visibility of center of mass and object labels. The Reset View button re-centers the grid, and the Delete Linkage button reloads PMKS+, erasing all work done.



**Figure 3:** The *Animation Bar* contains three main groups of features: Linkage Information, Simulation Controls, and Grid Controls

The *Linkage Table* (shown in Figure 4) displays properties for joints, links, and forces in the created mechanism. It features three tabs: Joints, Links, and Forces. As more objects are added to the grid, the table expands until a scroll bar is added when the table reaches the screen's height.

The image shows three tabs of the Linkage Table: Joints Tab, Links Tab, and Forces Tab. Each tab contains a table of data for the mechanism's components. Labels point to specific data points: 'Angle for Prismatic Joints' in the Joints tab, 'Moment of Inertia' and 'Center of Mass' in the Links tab, and 'Force Magnitude' in the Forces tab.

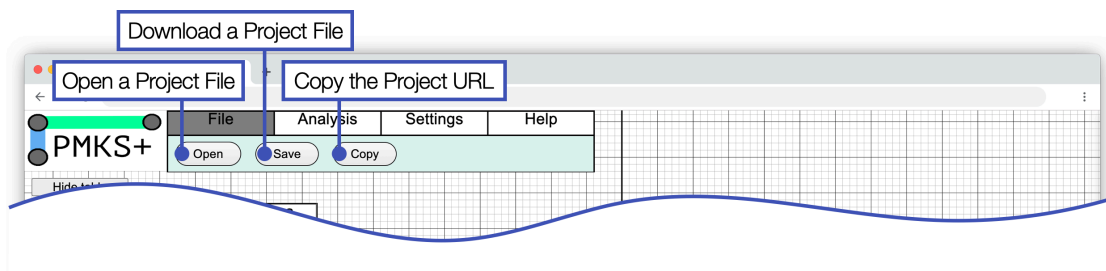
Joints Tab					
ID	Linked To	Dist From Joint (cm)	x (cm)	y (cm)	angle (deg)
a	b	3.116	-1.24	-0.89	
b	a	3.116	-0.1	2.01	
c	b	4.368	4.26	2.27	
d	c	3.401	3.36	-1.01	

Links Tab				
ID	Mass (g)	Mol (kg * cm <sup>2</sup> )	X CoM (cm)	Y CoM (cm)
ab	1	1	-0.67	0.559999E
bc	1	1	2.08	2.139999E
cd	1	1	3.809999E	0.63

Forces Tab					
ID	On Link	Angle (deg)	Mag (N)	X Pos (cm)	Y Pos (cm)
F1	bc	69.0825	1	1.92	2.13

**Figure 4:** The *Linkage Table* is split between three tabs, *Joints*, *Links*, and *Forces* with a row for each object

The Toolbar contains four tabs: File, Analysis, Settings, and Help. The file tab (shown in Figure 5) is selected by default and contains three buttons: Open, Save, and Copy. The open button allows users to open an existing PMKS+ created linkage saved as a .csv file. The save button saves the created mechanism as a Comma Separated Value (CSV) file, and the copy button saves the created mechanism as a URL file with all necessary data encoded in query strings.

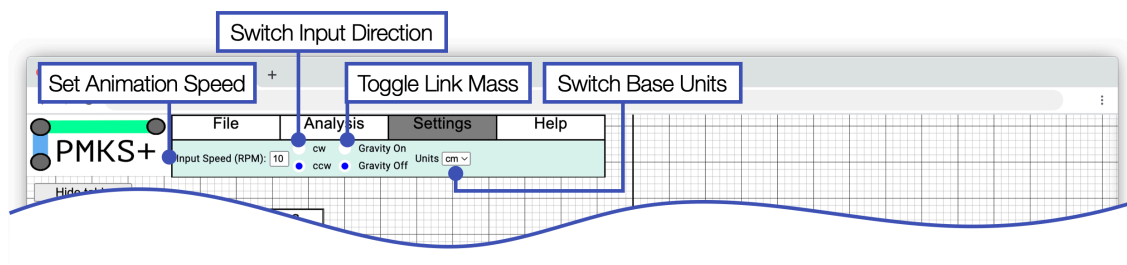


**Figure 5:** The *File* Tab on the Toolbar contains three buttons: Open, Save, and Copy

The URL created contains all the information required to recreate the mechanism and is human-readable. It follows the format of hostname:port/pathname?j= ... &l= ... &f=..., &p=..., &tp=..., and &s=....All joint data follows the “j=”, all link data follows the “l=”, all force data follows the “f=”, and all saved setting data follows the “s=”, respectively. The property values of an element are separated by “,” and “|”. The “RelatedJointID” property for links has its related joint IDs separated by “|”. Numbers are rounded to the nearest two decimal places. Booleans are shortened to “t” or “f”, the shapes of links are shortened.

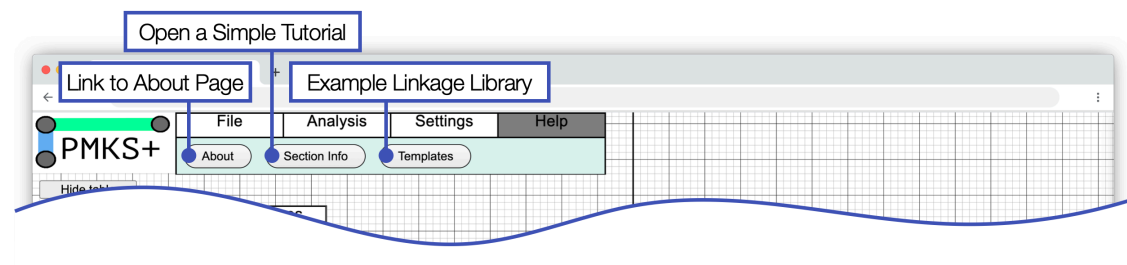
```
https://pmksplus.mech.website/?
j=A,-0.7,-0.6,AB,R,t,0,0.1,t,\%0AB,-0.72,1.6,AB,R,f,0,0.1,f,\%0A\
&l=AB,1,1,-0.71,0.5,A\%7CB,,1,-0.72,-0.6,-0.7,-0.6,-0.7,1.6,-0.72,1.6\%0A\
f=\&pp=\&tp=\
&s=10,false,false,cm
```

The *Settings* tab (shown in Figure 6) offers four functionalities: input speed specification, input direction selection, a gravity toggle, and unit options. Users can set the input speed of the crank in RPM, choose the direction of rotation, enable or disable the consideration of gravitational forces during force analysis, and select between centimeters (cm) or meters (m) as the unit system.



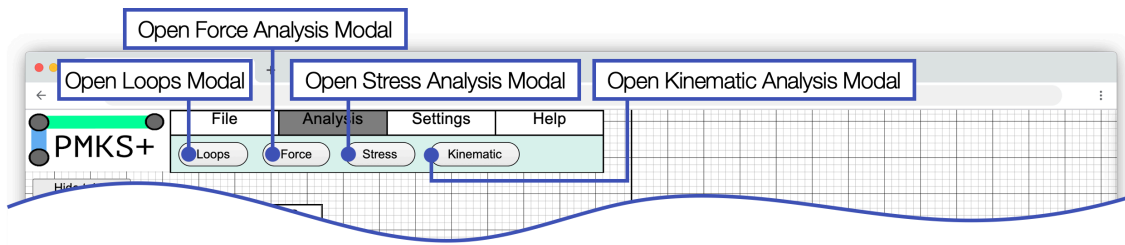
**Figure 6:** The *Settings* Tab on the Toolbar contains four features: Input Speed, Input Direction, A Gravity Toggle, and unit options.

The *Help* tab (shown in Figure 7) provides three resources: About, Section Info, and Templates. The About page offers information on PMKS+ development and its various versions, while Section Info presents a basic tutorial on PMKS+ features. Templates display example mechanisms that can be created using PMKS+.



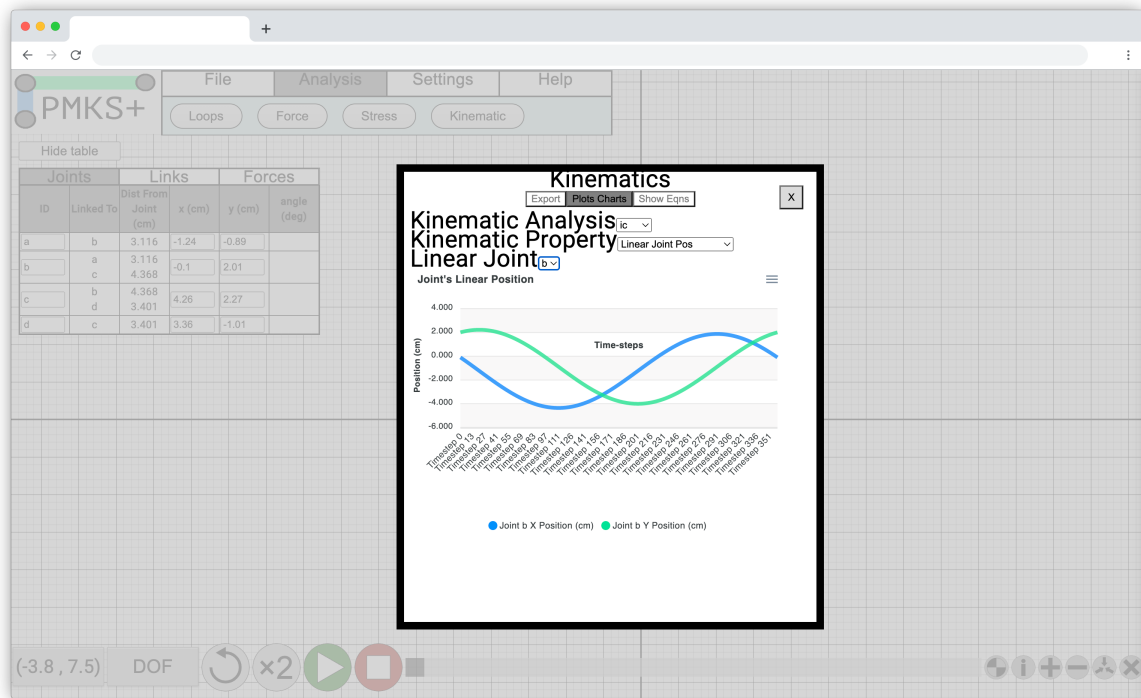
**Figure 7:** The *Help* Tab on the toolbar contains three buttons: About, Section Info, and Templates

The *Analysis* tab (shown in Figure 8) enables four types of analysis: Loops, Forces Static, Force Dynamic, Kinematics, and Instant Centers. The Loops button generates a document outlining all possible kinematic loops, while the Forces buttons compute joint forces and input torque using static equilibrium and Newton's second law methods. The Kinematics button extracts position, velocity, and acceleration data, and the Instant Centers option identifies instant centers in the system and calculates velocities using these centers. All options except Loops provide charts and associated equations.



**Figure 8:** The *Analysis* tab in the toolbar allows four types of analysis to be performed.

Clicking on any of the analysis buttons opens a popup window (shown in Figure 9) containing three tabs: *Export*, *Plot Charts*, and *Show Eqns*. The *Export* tab allows users to export selected information to a Microsoft Excel Open XML Spreadsheet (XLSX) file. The *Plot Charts* tab lets users select a property and object to display a graph of the property over the simulation time. The *Show Eqns* tab provides the steps and equations needed to generate the graph. Students can use these features to compare their analyses with PMKS+ results and identify discrepancies.

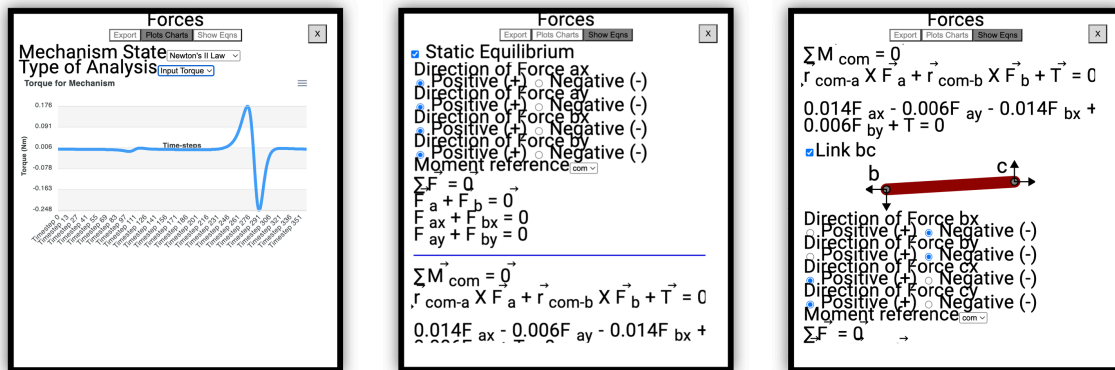


**Figure 9:** The analysis popup window with the three tabs: *Export*, *Plot Charts*, and *Show Eqns*.

Two examples of completed pop-up windows are for force and kinematic analyses. Fig-

ure 10 shows a force analysis example. Users first choose to display results using static equilibrium or Newton’s second law and then either display a joint’s internal forces or the required input torque. If joint forces are selected, the user must pick the desired joint. A line plot is created, and users can hover over values to see force/torque at a given timestep. Users can export as SVG, PNG, or CSV files.

Clicking on the option to showcase equations displays the representation used for manual analysis. First, a free-body diagram (FBD) of the link is drawn with positive x and y-axis pointing to the right and up. Underneath are the assumptions used to set up the FBD, including the direction of the internal forces and the set moment reference for a link. Equations show the sum of forces in the x and y-direction and the sum of moments.



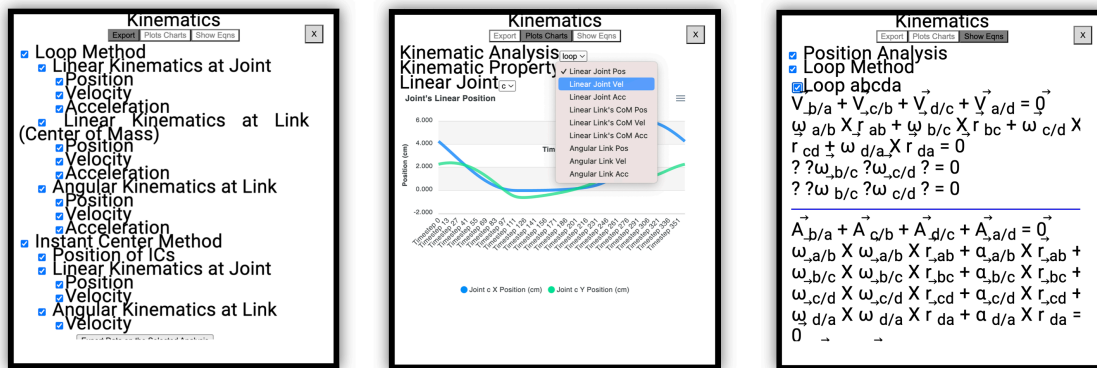
**Figure 10:** Force Analysis Popup enables data exporting, graph generation, and options to define the positive force directions at each joint in a FBD.

PMKS+ has two methods to determine kinematics analysis (shown in Figure 11) : Loop Methods and Instant Centers. Loop Methods can export linear kinematics at a joint, linear kinematics at a link’s center of mass, and angular kinematics of a link. The Instant Center method allows users to produce position, linear kinematics of a joint, and angular kinematics at the link. However, the angular position export for a link is missing.

Plot Charts showcase the analysis produced throughout a given analysis. Users first select the desired kinematic analysis method, then the desired kinematic property, and finally the desired joint or link. A line plot is produced.

Kinematics does not showcase equations for position analysis, the method for the mechanism’s next timestep, or images showing loops or instant centers used to determine analysis. Loop method and instant center method equations are separated.

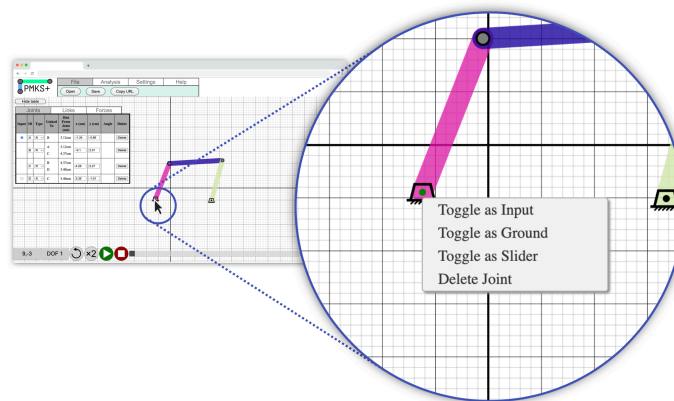




**Figure 11:** The *Kinematic Analysis* Tab allows for Excel export and graph generation, as well as the generation of loop equations needed for kinematic analysis.

Users create mechanisms by interacting with the Grid. Left-clicks move joints, links, forces, and the grid, while right-clicks open context menus for creating, deleting, or changing parameters. Context menus (joint context menu shown in Figure 12) vary depending on whether the user right-clicks on the grid, joints, links, or forces.

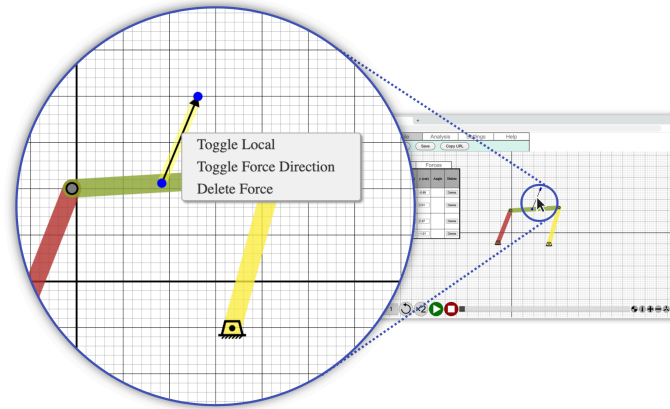
All types of joints, links, and forces have an option to delete. Deleting a joint with only one connected link will also delete the link. If the joint does not connect two links, it is a tracer point. A tracer point can toggle as ground and prismatic, allowing the user to toggle it as an input joint.



**Figure 12:** The right-click joint context menu offers options to delete the joint and toggle Input, Ground, and Slider.

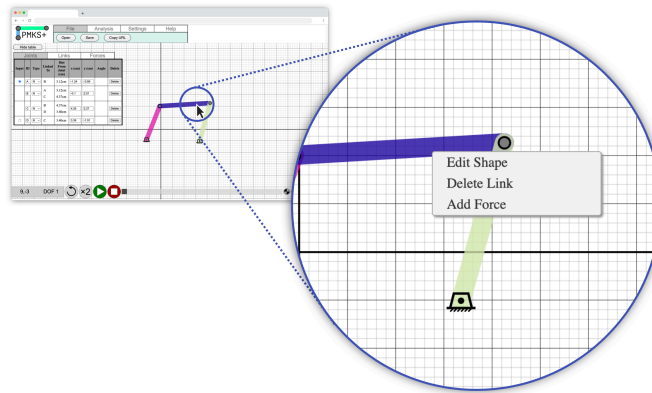
When right-clicking on a force, besides the option to delete, it opens a context menu (shown in Figure 13) that allows users to change its local/global position and direction. Selecting local

position changes the force line color from black to yellow and vice versa. Changing the force's direction moves the arrow to the opposite side.



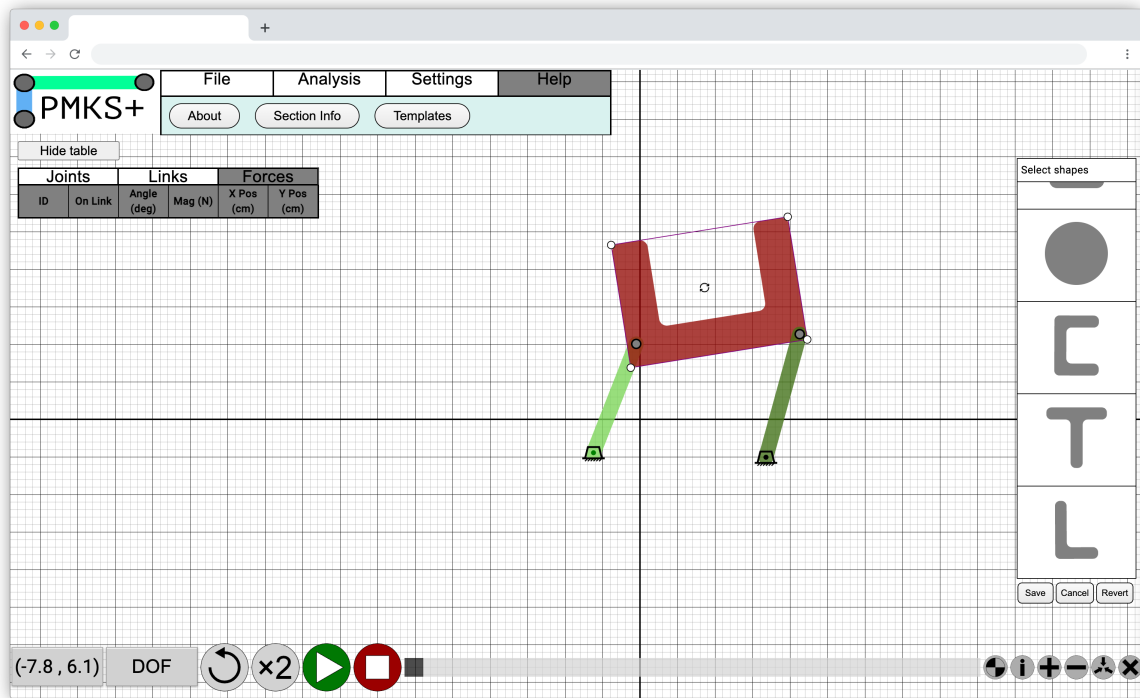
**Figure 13:** The right-click force context menu allows users to delete the force, toggle the reference frame, and change its direction.

All link context menus (an example is shown in Figure 14) contain options to edit shape, delete, and add a force.



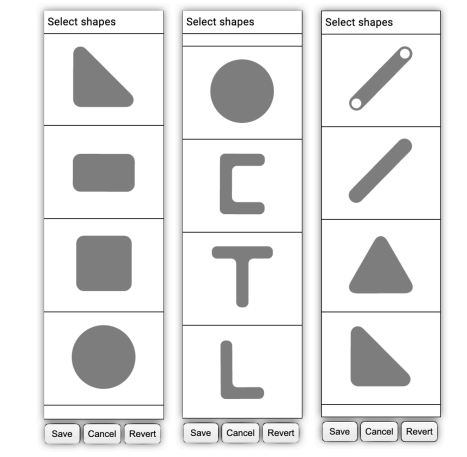
**Figure 14:** The right-click link context menu allows users to delete the link, edit the link shape, and add a force to the link.

Selecting *Edit Shape* opens an overlay with a list of shapes on the right. Users can swap the link to a new shape, and a bounding box appears around the link, allowing users to drag, scale, and rotate the shape. The interface for this is shown in Figure 15.



**Figure 15:** Clicking *Edit Shape* opens the shape editing panel and puts PMKS+ into a shape editing mode.

There are 11 shapes available in the shape library including the binary link as shown in Figure 16. Note that the entire shape can be scaled, but not individual side lengths relative to each other.



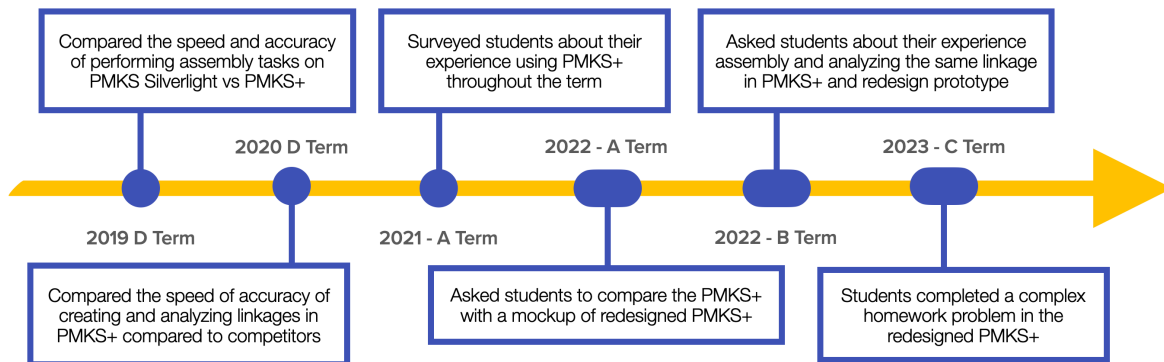
**Figure 16:** The user must select one shape from the 11 shapes available in the shapes library.

### 3 User Studies and Research

Gaining a thorough understanding of the users and the context of the application is crucial for carrying out a redesign with maximum benefit. In this section, users of PMKS are comprehended by performing user studies, sorting them into personas whose journeys are mapped to identify pain points and user needs.

#### 3.1 Methodology for Conducting User Studies

To thoroughly understand the user experience and identify the key areas for improvement, multiple user studies have been conducted over the years PMKS+ has existed, as detailed in previous PMKS+ reports [6, 3, 4, 5]. Figure 17 shows the most important ones in a timeline format.



**Figure 17:** A timeline of the most relevant user studies conducted since PMKS+'s inception in 2019. The studies performed as a part of this thesis are shown with an oval.

##### 3.1.1 User Studies from Previous Years

After the conversion from PMKS+ was complete, in the spring of 2019 (D'19), the team had 18 students complete a series of tasks that involved creating the same linkage twice, using PMKS and next using PMKS+. They recorded the number of mistakes made by users and the overall time it took to accomplish the assembly task. The task completion time for PMKS+ users decreased significantly, with an average time of 3.66 minutes compared to 6.87 minutes for PMKS users. Additionally, over 70% of PMKS+ users made three or fewer mistakes during the evaluation, highlighting the potential for reduced errors compared to the original PMKS. [3]

After the next student team finished their improvements, another user evaluation study involving 18 students was conducted in the spring of 2020 (D'20). The team received valu-

able feedback on PMKS+ and identified areas for improvement. The study, which compared PMKS+ to Working Model, showed that PMKS+ had a slight advantage in overall satisfaction, scoring 3.27 compared to Working Model's 3.0. Although not statistically significant, the feedback highlighted the intuitiveness of PMKS+ and helped the team uncover previously unnoticed errors, enabling them to refine the application further.[4]

After rounds of testing and improvements throughout 2021, in Spring 2022, the upgraded software was tested in ME/RBE 4322. The 26 surveyed students' experience was highly positive, with 80% successfully using the software without assistance. The teaching team found PMKS+ helpful for providing detailed feedback to students, comparing equations and numerical results, and identifying errors in students' MATLAB scripts. [5]

### 3.1.2 User Studies during 2022-23

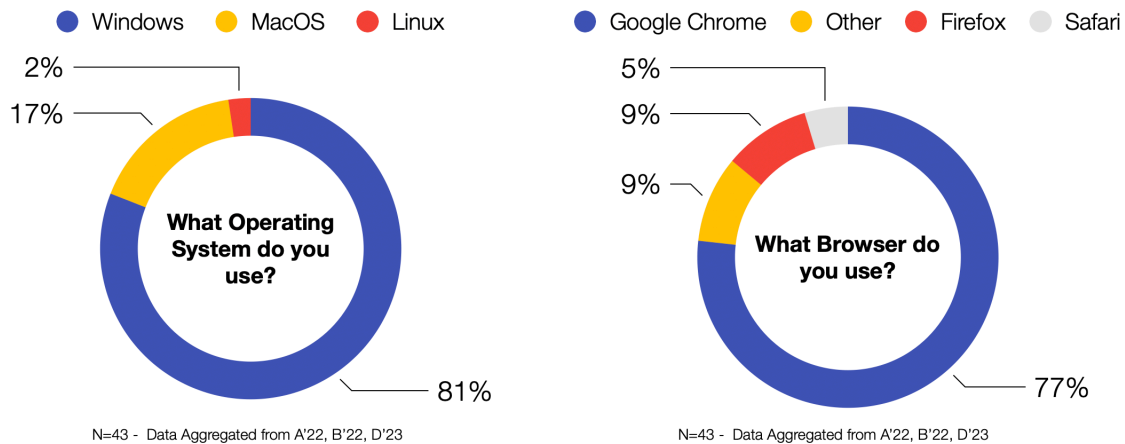
There were three user studies performed during this thesis and the first one was used to get a better idea of the current state of the application before the redesign started.

In the fall of 2022 (A'22), 15 students who had used PMKS+ for a full term while working on various projects were interviewed about their experiences. They were invited to sit down and go through a sample problem they had worked on to gain a comprehensive understanding of how current users interact with the application. Through this process, two distinct user types were identified, each with a unique set of challenges.

## 3.2 User Demographics

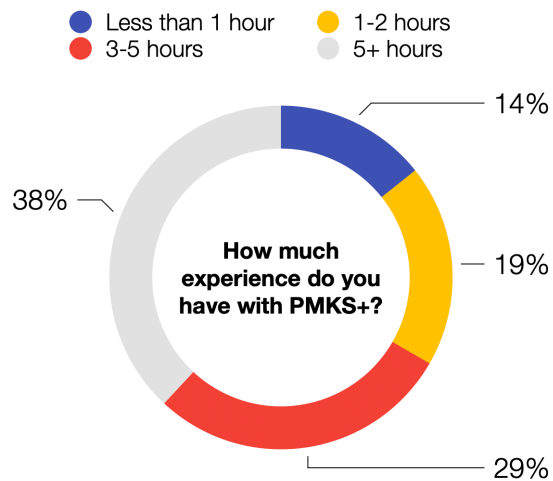
First, based on aggregated data from the aforementioned studies, we are able to get a better idea of the current user base of the application. Here are some key insights:

- **Close to 100% of users are WPI students.** Since PMKS+ has only been promoted through word of mouth, only users who are affiliated with WPI use PMKS+.
- **Most users are Mechanical Engineering or Robotics Engineering Majors.** Students who need to perform linkage analysis as part of their coursework or project tend to be ME or RBE students.



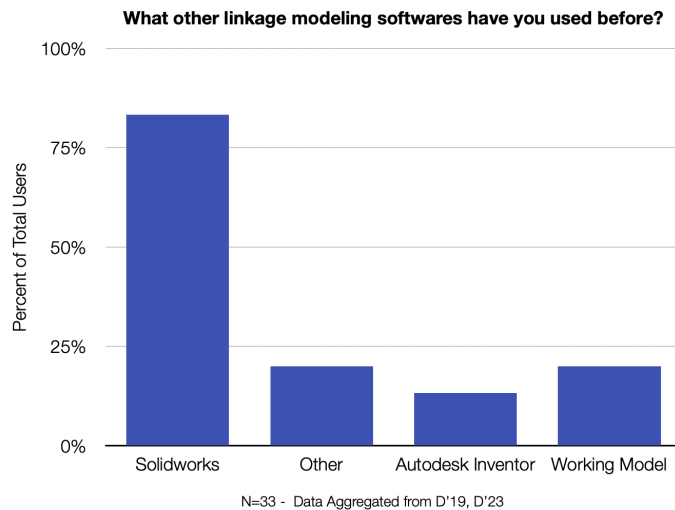
**Figure 18:** Over 3/4th of the users use PMKS+ using Google Chrome on Windows.

- **Users are running PMKS+ in Google Chrome on Windows computers.** As shown in Figure 18 Engineering students tend to use Windows for better compatibility with other applications they need for their coursework. Our users are no different and reflect that demographic. MacOS operating system and Firefox browser users are a distant second.



**Figure 19:** Over half of the users have more than 3 hours of experience with PMKS+

- **Most users have 3+ hours of experience.** As evidenced by Figure 19, since students tend to use PMKS+ for a full term at WPI, they have a large amount of experience with the application.



**Figure 20:** Almost all users have experience with some type of engineering program.

- **Most users are familiar with Solidworks.** Throughout their curriculum, WPI ME and RBE students get trained with Solidworks, a very comprehensive solid modeling computer-aided design and computer-aided engineering application published by Dassault Systèmes. Data from Figure 20.

### 3.3 User Journey Maps

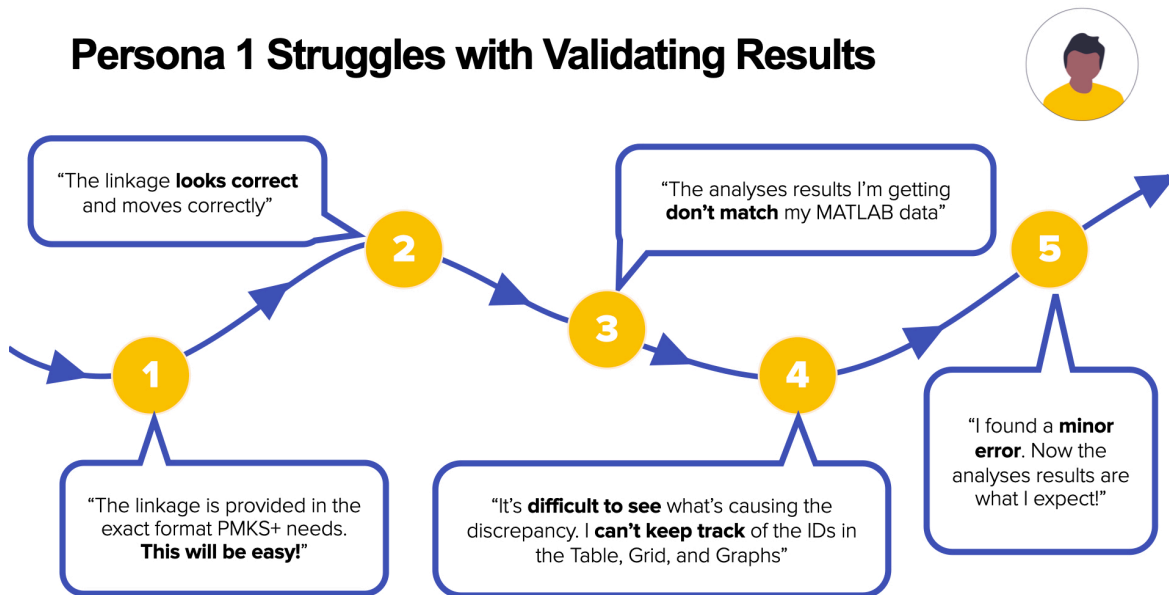
Based on these comprehensive surveys, along with multiple unstructured interviews with other stakeholders, two user personas were developed to represent the primary user groups:

- **Persona 1:** A WPI student in ME/RBE courses who is required to use PMKS+ as part of their coursework. These students use PMKS+ to validate their MATLAB results by recreating and analyzing mechanisms provided in assignments.
- **Persona 2:** An engineering student working on Major Qualifying Projects (MQPs) or other final projects who want to design a linkage. For these users, PMKS+ is a useful tool for saving time on rough calculations and validation.

Journey maps were created for each persona to identify the most significant problems they encountered during their interaction with the application. Although there are touchpoints that come before they start using the app (like learning about it from a friend or through a class) and after (like validating the accuracy of the data with a physical model or more advanced

simulation software), we are focusing on the journey for one project, between when the user opens the app and when they receive all the data they need.

Broadly, each journey map can be split into two stages: the creation (or editing) of the linkage in the app and the analysis, where they run the simulation with the linkage to extract the data they need.

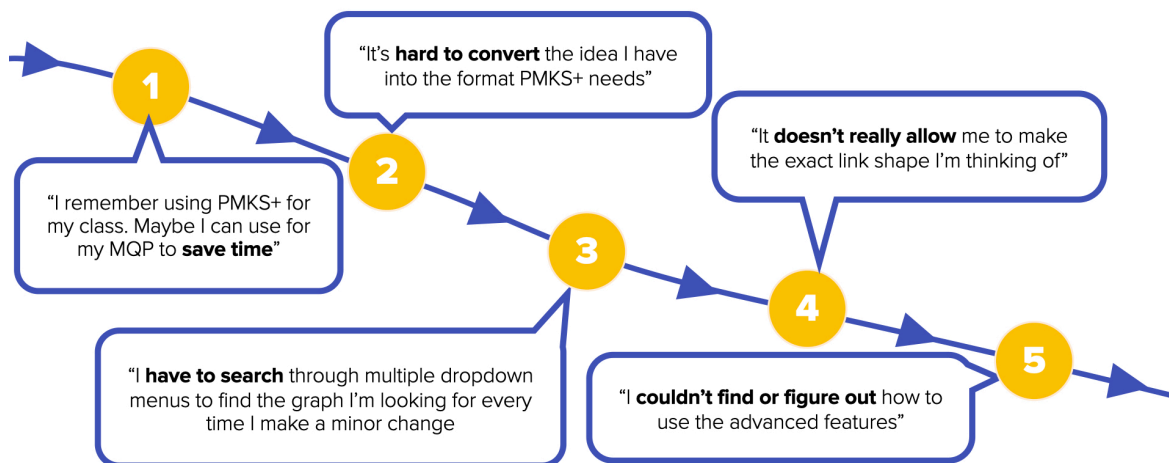


**Figure 21:** Persona 1 Struggles with finding discrepancies between their Matlab and PMKS+ results for their homework.

Persona 1 needs to be able to create a linkage quickly in any format they are given. They have a clear idea of what this linkage looks like since it is provided to them in the assignment, although different problems present the linkage in various formats. They need quick access to the analysis data and want to complete this process as quickly as possible. Since PMKS+ is often used as a method to verify work a student performs in another program or by hand, Persona 1 often finds a discrepancy in the two results. When this happens, since PMKS+ only provides the final results, they have no easy way to figure out which steps went wrong. They become frustrated and submit the linkage with errors or bring the whole problem to the teaching staff for resolution. Persona 1's full journey map is shown in Figure 21.



## Persona 2 doesn't have enough options to prototype in the application



**Figure 22:** Persona 2 cannot adequately prototype a linkage for their final project inside PMKS+

Persona 2, on the other hand, needs PMKS+ to act as a versatile prototyping tool. They have an idea of what the linkage they want to create does, but not the specific shape or dimensions. They want to use a simple sketching tool to draw multiple ideas and go back and forth between editing and analyzing until they end up with a linkage that meets their needs. They want the app to be flexible and versatile, allowing many types of joints and link shapes. Persona 2's journey map is shown in Figure 22.

Based on a survey with a convenience sample at WPI, most users currently fall into the Persona 1 category since the primary method of discovery for this application is through its required use in certain classes. For both personas, the biggest barrier to getting the desired data from the application is the time it takes to create a linkage model that accurately represents their design on paper or their thoughts.

### 3.4 Other Stakeholders

Although the users of the application are the primary stakeholder for PMKS+, there are two other groups that should be considered.

- **Professors teaching classes where PMKS+ is used for assignments, along with other teaching staff.**

Although PMKS+ is introduced to a curriculum with the intent to allow students to be more self-sufficient as they validate their answers and find discrepancies, sometimes this is not the case. Students who struggle to use the application may need additional 1-on-1 help fixing issues with the software or the project file. Students who are able to create the linkage but do not find or understand the powerful analysis tools might submit their work after only doing a visual check of the mechanism. When these things happen, course staff have to spend more time grading to figure out what went wrong, which leads to them spending time fixing issues with the application instead of spending their time doing more important things like teaching concepts.

- **Developers and maintainers of the application.**

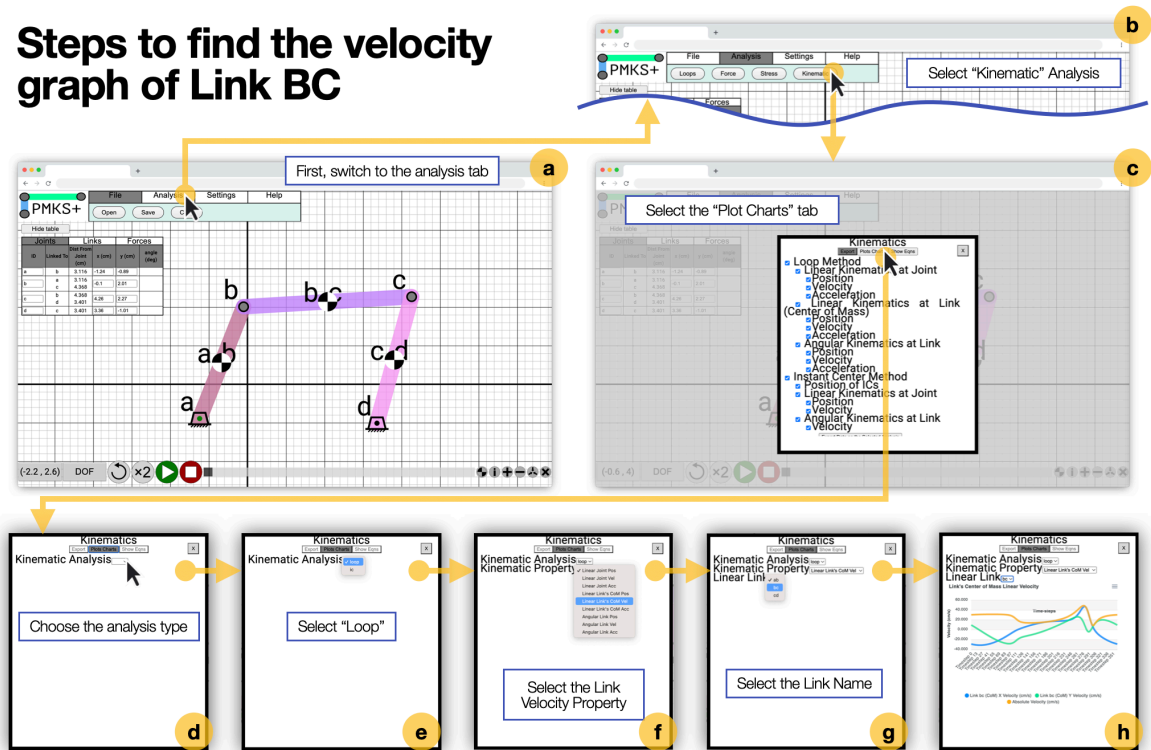
Although this impacts a significantly smaller number of people, as this project has been maintained for many years and will be updated for many more, we need to make sure the developer experience is as frustration-free as possible so that the team can focus on shipping features that impact users. This means a good development environment out of the box with documentation for code explaining why certain things were done and what else was considered.

### **3.5 Pain Points Identified**

By getting a comprehensive understanding of what our users look like along with how they use the app, several key pain points in PMKS+ were identified.

1. **Lack of discoverability and access to analysis features:** Users currently need to switch tabs, open a popup, and select from two dropdown menus to access analysis features.

## Steps to find the velocity graph of Link BC



**Figure 23:** Series of images showing the process to open the graph showing the velocity of a link. From left to right and top to bottom: (a,b) Select the kinematics button in the analysis tab. (c) In the popup, select the *Plot Charts* tab. (d-g) In the 3 dropdowns, select Kinematic Analysis: *Loop*, Kinematic Property: *Linear Link's COM Vel*, and Linear Link: *bc*.

This cumbersome process (shown in **??**) requires too many clicks to access a critical aspect of the application. As more joints and links are added, the process becomes increasingly difficult to manage, making it hard to remember which label is associated with which object since users cannot see them simultaneously. One user mentioned:

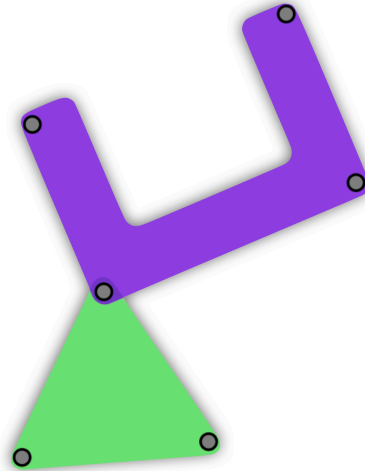
I didn't even really use the analysis because even though it was accessible in the top bar, it was still kind of hidden.

Other important features that are currently hidden under the analysis tab also include the equation display for students to validate their work. Additionally, a few important features (such as seeing all available loops for the loop equation are only viewable through a csv export).

2. **Linkage table issues:** The linkage table is a feature carried over the original PMKS Sil-verlight without reevaluation of its effectiveness. As shown in Figure 24, It occupies a

significant amount of usable grid space and expands as more information is displayed for each object or as more joints and links are added.

Joints		Links		Forces	
ID	Linked To	Dist From Joint (cm)	x (cm)	y (cm)	angle (deg)
b	c	2.799			
	e	2.761	-0.973	0.762	
c	b	2.799	1.816	0.997	
	e	2.735			
e	b	2.761			
	c	2.735			
	f	4.109	0.249	3.238	
	i	2.721			
	j	4.968			
f	e	4.109	4.02	4.87	
	i	4.918			
	j	2.729			
i	e	2.721	-0.82	5.74	
	f	4.918			
	j	4.139			
j	e	4.968	2.976	7.391	
	f	2.729			
	i	4.139			



**Figure 24:** Linkage table gets extremely long even with only these two links

Users reported issues with the limited number of digits displayed in the table. One user commented:

The table only shows two or three digits; it would be useful to be able to see more digits and user define the size of the table.

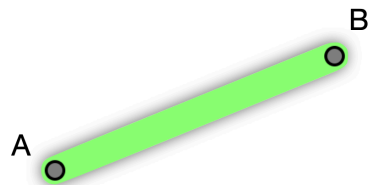
### 3. Difficulty in creating linkages:

A typical user spends the largest amount of their time creating the initial linkage design. Without any automated method to generate a linkage, having an easy way for users to transfer an idea or schematics for a linkage into PMKS+ is vital. Currently, users must drag joints to create binary links or enter the coordinates of each endpoint. As shown in Figure 25, there is no option to define joints relative to another using link lengths or angles. Dragging does not offer snapping, making it challenging to make precise angles for links.

Joints		Links			Forces
ID	Linked To	Dist From Joint (cm)	x (cm)	y (cm)	angle (deg)
a	b	4.126	-1.86	0.96	
b	a	4.126	1.96	2.52	

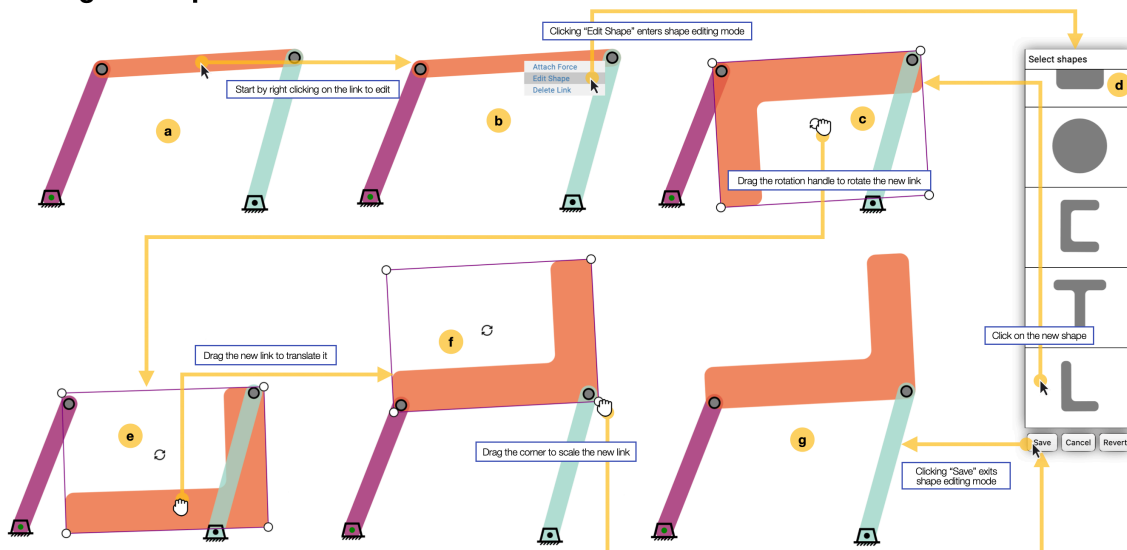
  

Joints		Links		Forces	
ID	Mass (g)	Mol (kg * cm <sup>2</sup> )	X CoM (cm)	Y CoM (cm)	
ab	1	1	0.0499999	1.74	



**Figure 25:** The only option users have to edit the length of a binary link is to enter the x and y coordinates of each joint.

### Editing the shape of a link

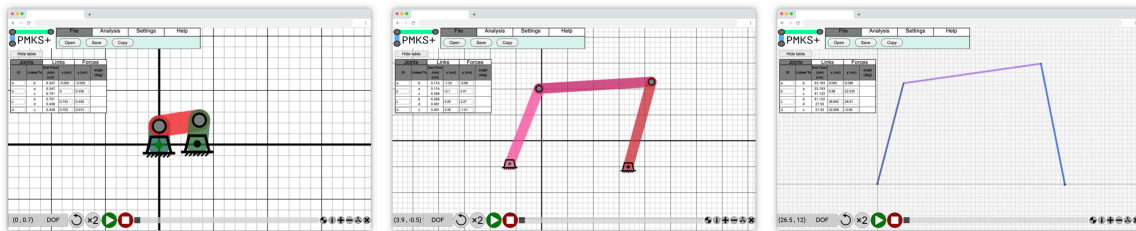


**Figure 26:** A sequence of images showing the process of editing a link shape. From left to right, then top to bottom. (a,b) Right-click on a link and select *Edit Shape* in the context menu. (c) Select the new shape from the scrollable right panel showing a list of available shapes. (d-f) Use the bounding box and rotation icon to translate, rotate, and scale the new shape, making sure to include all joints within the new shape. (g) Click save on the shape panel to confirm the shape change.

Students struggle to create the exact shape of the link they want. As shown in ??, when a non-binary link is desired, the drag-to-move joint interaction pattern becomes unworkable. Users must switch to a different mode, enter edit shape mode, and click save or cancel when finished. Selection is limited to a list of predetermined shapes, and users must scale, move, and rotate the shape to fit over the joints. A user remarked:

It would be nice to be able to say that points A, B, and C are all part of the same link and have a triangular link created with each of the points as corners. Currently, the method of placing the shapes on top of the points is not very intuitive and looks very strange.

4. **Limited grid and workspace range:** The grid has a limited usable area meaning users cannot make linkages that are too small or too large. This forces users to perform additional conversions to ensure their entire linkage fits within this area. As shown in Figure 27, when a users zoom out, the linkage becomes smaller and thinner, making it difficult to interact with and see, or too large when zoomed in.



**Figure 27:** A similar four-bar linkage created at three different scales. The smallest linkage (5cm) makes the UI elements too large relative to the linkage and the largest (100cm) has the opposite problem.

The distance between each grid line is constant, making grid lines impractical to count unless the zoom is perfect. One user stated:

[PMKS+] could only handle small-scale models. The mechanism from Homework 1 could not fit into PMKS+ on a 1:1 scale, which in turn made everything more confusing in terms of units.

5. **Various quality of life issues and bugs:** Small bugs and issues can have a significant impact on user experience. For instance, links and forces cannot be renamed, making it challenging to keep link names consistent with assignments. Users must memorize link or joint IDs and select them from a list to analyze an object. Additionally, users cannot input 1 inch into a textbox when the global unit is metric, forcing them to use 2.45 cm instead. The colors of the links are randomized each time the linkage is loaded, creating further confusion. Many icons and labels lack clarity, and tooltips are absent. Although more bugs and issues exist than can be listed here, additional raw quotes can be found in the reference section. This highlights the importance of addressing these quality-of-life concerns to improve the overall user experience.

These pain points highlight key areas for improvement in PMKS+ and guide our efforts to create a more user-friendly and efficient experience.

### **3.6 User Needs**

One of the primary user needs is to simplify the process of creating a linkage from their assignment. Users should be able to easily input information and create linkages without encountering unnecessary complexities or confusion. By streamlining the process, the application can ensure that users can focus on understanding the underlying concepts and principles, rather than spending excessive time learning how to navigate the software.

Another important user need is to allow for a linkage to more closely resemble typical homework problems. By providing an interface and tools that enable users to create linkages that mirror their assignments, the application can enhance the learning experience and help users better understand the concepts being taught. The software should be designed with the user's perspective in mind, ensuring that it caters to their academic requirements and facilitates their learning process effectively.

Additionally, users need an easy way to resolve discrepancies in a linkage. When there is a discrepancy between the analysis they perform on another program or by hand and the results of PMKS+, students need a way to see the steps taken and equations used by PMKS+ to arrive at the final result. By incorporating features that help users pinpoint and correct errors, the application can significantly improve the overall user experience and make it more efficient for students to complete their assignments or projects. This, in turn, reduces the load on teaching staff who would otherwise have to manually check the students' work.

Finally, allowing prototyping to happen inside the app by making the switch between editing and analysis more seamless is a vital user need. Users should be able to effortlessly transition between these two modes to refine their designs and test their linkages. A seamless transition between editing and analysis encourages users to experiment, iterate, and fine-tune their work, ultimately leading to better understanding and improved outcomes. By prioritizing this aspect of the software, the application can foster a more productive and engaging learning environment for users.

### **3.7 Competitive Analysis of Existing Linkage Analysis Software**

. This section presents a comprehensive analysis of the existing linkage analysis software to identify the gaps in the current offerings and highlight the areas where a new PMKS+ could outperform the competition. The main objective of this analysis is to understand the capabil-

ities and limitations of the current software tools and propose a new solution that addresses these shortcomings.

The following factors are considered essential for users when selecting a linkage analysis application:

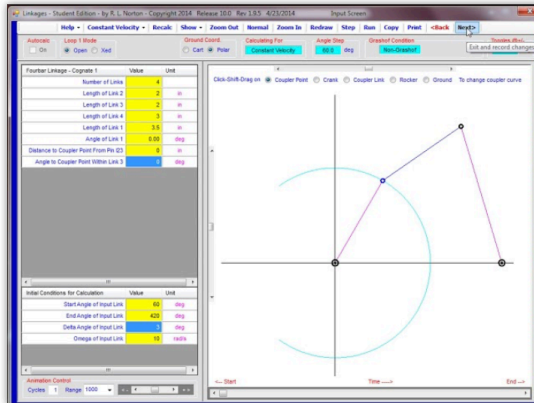
1. User Interface and Ease of Use
2. Licensing and Accessibility
3. Kinematic Analysis
4. Force Analysis
5. Linkage Synthesis
6. Educational Focus

Based on the aforementioned criteria, Table 1 summarizes the capabilities of the reviewed software tools. Currently, there is no free and easy-to-use software that is education focused while providing powerful kinematic and force analysis capabilities. A more thorough overview of each software has been conducted in our previous work [5].

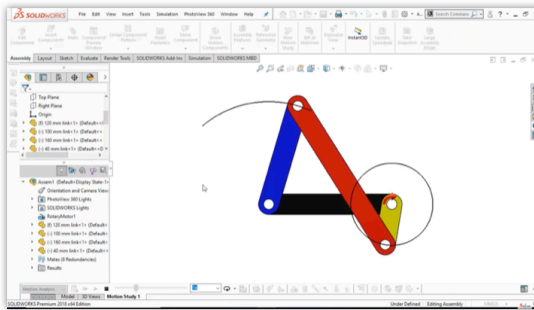


## Popular Linkage Analysis Tools

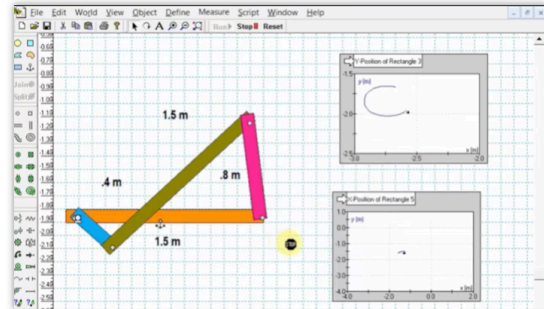
### Norton Linkages



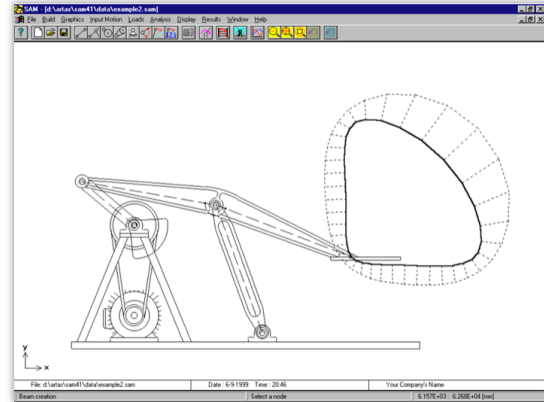
### Solidworks



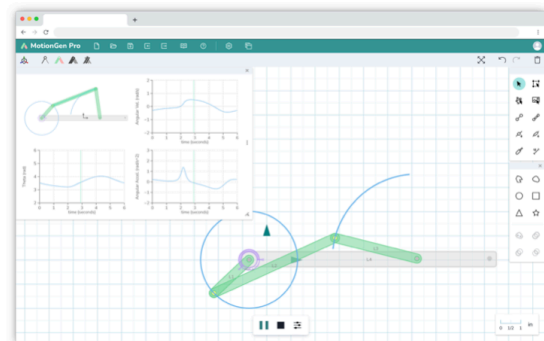
### Working Model



### SAM



### MotionGen Pro



**Figure 28:** A collection of images showing representative views of each application discussed in the table. This provides a general idea of each software's appearance and user experience.

As illustrated in Table 1, none of the existing software tools pictured in Figure 28 fully satisfy the requirements presented. This reveals a significant opportunity for the development of a new software tool that addresses these limitations and offers a comprehensive solution for mechanism design and analysis.

For instance, SolidWorks, despite being a popular and powerful CAD software, lacks ca-

**Table 1:** Table comparing key features in currently available linkage analysis software. Force analysis is considered only when the entire link is analyzed. Synthesis options include position-based and path-based synthesis. Quick is subjective and represents if an average user can create a linkage in less than 10 minutes without a tutorial.

Features / Software	Working Model	Norton Linkages	SAM	Solidworks	MotionGen Pro
Kinematic Analysis	✓	✓	✓	✓	✓
Force Analysis		✓	✓		
Synthesis Options			✓		✓
Quick to Create Mechanism			✓		✓
Quick to Analyze Mechanism		✓	✓		✓
No Installation Required					✓
Free To Use					✓
Display Analysis Equations for Checking Work					
Source code available					

pabilities in motion synthesis and path and function synthesis. Furthermore, its complexity results in a steep learning curve. Additionally, SolidWorks encounters licensing and accessibility issues due to high costs and difficulties in setup. In contrast, MotionGen and SAM, which provide motion synthesis and path and function synthesis respectively, do not cover all necessary analysis techniques and have their own limitations. Lastly, Working Model and Norton Linkages, although offering some analysis functions, lack synthesis capabilities and have outdated, hard-to-approach user interfaces. [7, 8, 9, 10, 11]

There is a clear demand for a free, web-based, comprehensive, and user-friendly software tool that addresses all aspects of mechanism design, analysis, and synthesis. The PMKS+ software, under development at WPI, presents a unique opportunity to fill this gap by incorporating the desired features and remedying the shortcomings of existing tools.

In conclusion, the competitive analysis demonstrates that none of the existing software tools adequately support the coursework and teaching methods related to mechanism design and analysis at WPI. By capitalizing on the open-source nature of PMKS+ and integrating the required features, there is an opportunity to develop a tool that effectively bridges this gap and provides a superior solution for students and educators alike.

## 4 Redesign Process

The research findings played a crucial role in shaping the redesign process, leading to the development of a comprehensive plan that prioritized user-centricity. By continuously seeking feedback from a diverse range of stakeholders, including students, educators, and other users, the redesign process aimed to create a final product that would address their needs effectively and cater to their preferences.

### 4.1 Redesign Goals

The redesign was centered around one primary goal and three secondary goals, all of which were geared towards enhancing the user experience and the software's functionality.

#### **Primary Goal:**

- *Improve the showcase of new features:* The existing UI did not effectively highlight the newly added features, which were crucial to the software's capabilities. The redesign sought to make features such as analysis and synthesis more discoverable, promoting organic user discovery and enhancing the software's competitive advantages.

#### **Secondary Goals:**

- *Address unintuitive quirks:* Certain features and tasks in the existing UI were hindered by unintuitive quirks, which required users to undergo extensive training or experimentation to understand. The redesign aimed to resolve these issues and create a more user-friendly experience.
- *Modernize the UI:* Updating the UI to incorporate contemporary design conventions was a key secondary goal. By aligning with modern design principles, the software would become more visually appealing and user-friendly.
- *Facilitate linkage creation:* The redesign sought to make it easier for users to create linkages, regardless of their specific format. This improvement would streamline the process, enabling users to work more efficiently and effectively.

By focusing on these goals, the redesign process aimed to create a more intuitive, engaging, and efficient experience for users, ultimately supporting their learning and understanding of mechanical concepts through the use of the PMKS+ software.

## 4.2 Low and High Fidelity Prototyping

Initial sketches and basic wireframes were created to explore potential UI elements that addressed the primary redesign goals. These rough ideas were informed by a deep understanding of user needs and preferences.



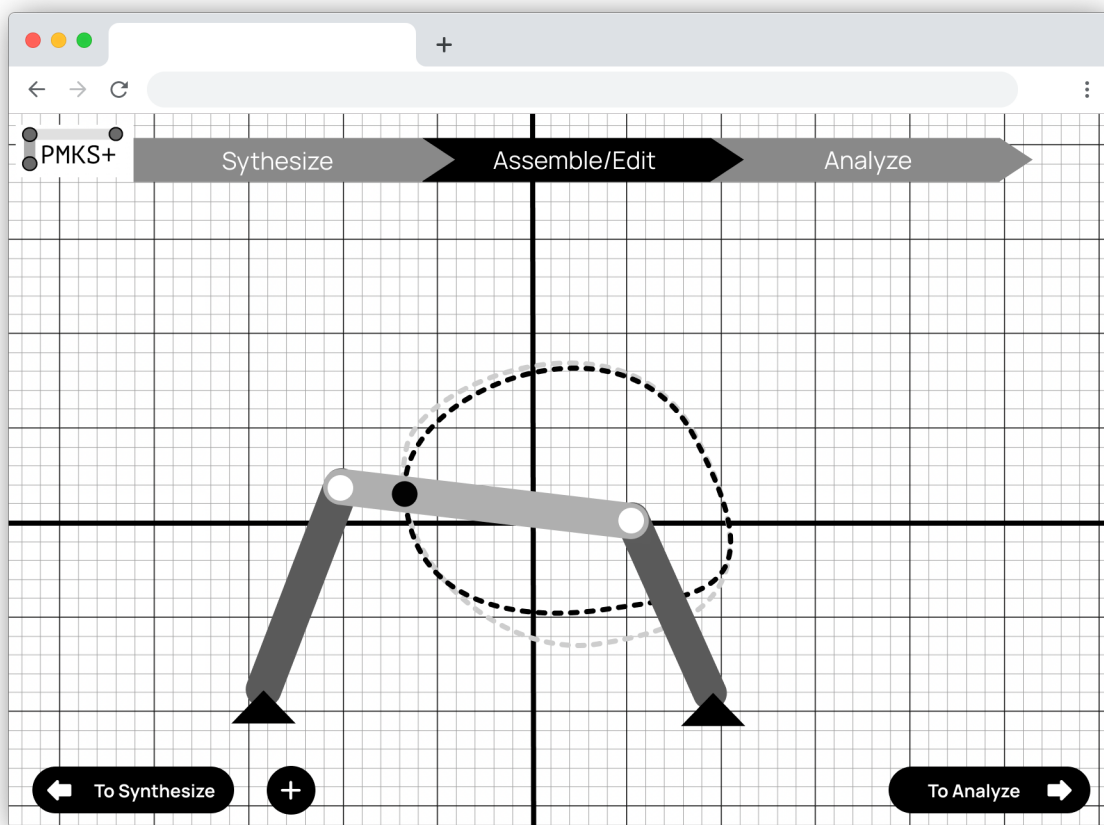
**Figure 29:** Mockup one introduces the idea of clicking an object to select it. The right panel shows settings (position, grounded?) or analysis graphs for the selected object.

The first high-fidelity mockup (shown in Figure 29) reimagined the interface for entering and editing joint and link properties. The linkage table, a legacy of the original Silverlight version, displayed extraneous information that often lacked immediate relevance to users. Replacing this with a progressive disclosure approach, the mockup introduced the concept of an active or selected object, displaying only its information on the screen. This change offered several benefits, including increased screen real estate for linkages, enhanced legibility of text and input boxes, and the ability to display and edit more object properties.

Furthermore, the availability of analysis options in the same area as editing options sig-

nificantly improved their discoverability, a key advantage of this mockup. However, the design also presented some drawbacks, such as the right panel occupying a constant amount of space, reducing the overall visible grid area. This limitation could prove challenging when dealing with more complex linkages, where the ability to view the full linkage on the browser screen is crucial.

Additionally, this design did not provide a natural location for incorporating a synthesis panel or tab. Such a feature would be dedicated to synthesizing an entire linkage, rather than being associated with a specific object (i.e., a joint, link, or force).

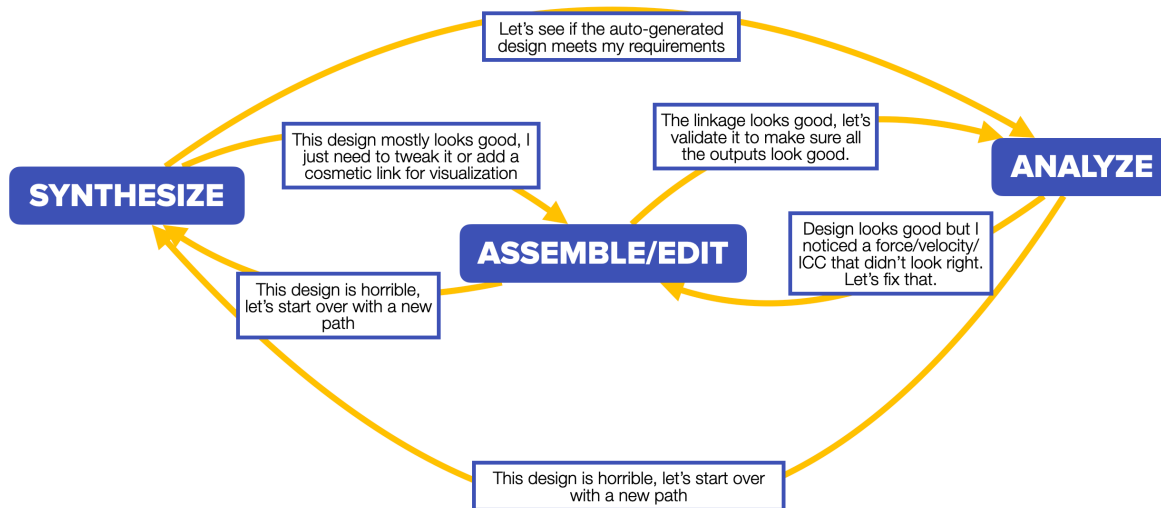


**Figure 30:** Mockup two leans into the idea of the three modes (Synthesis, Edit, Analyze) being a linear process that a user goes through. Users can jump to any mode by selecting the labels on the top or by using the arrow buttons on the bottom.

The second mockup (Figure 30) was developed to emphasize the linear progression of synthesizing, editing, and analyzing a linkage. Users typically synthesize a linkage when starting with a blank file and can only simulate and analyze the linkage after assembly. This prototype successfully addresses the main design goal of feature discoverability and guides users

through the expected sequence of events. Moreover, it implies that users should complete the analysis to finalize their assignment or project. While the bottom buttons enable users to switch to adjacent modes, the top tab allows direct access to any step. Additionally, this design permits the grid to occupy the entire screen area, with editing and analysis panels appearing only when an object is clicked.

However, the unidirectional popup approach presents a significant drawback: it discourages prototyping within the app. The user flow diagram in Figure 31 illustrates the numerous reasons for users to transition between the three modes.



**Figure 31:** A flow diagram of PMKS with modes in the solid blue boxes: Synthesis, Edit, and Analyze. Arrows show the movement of users between modes along with the most common thought process for why they might move between them.

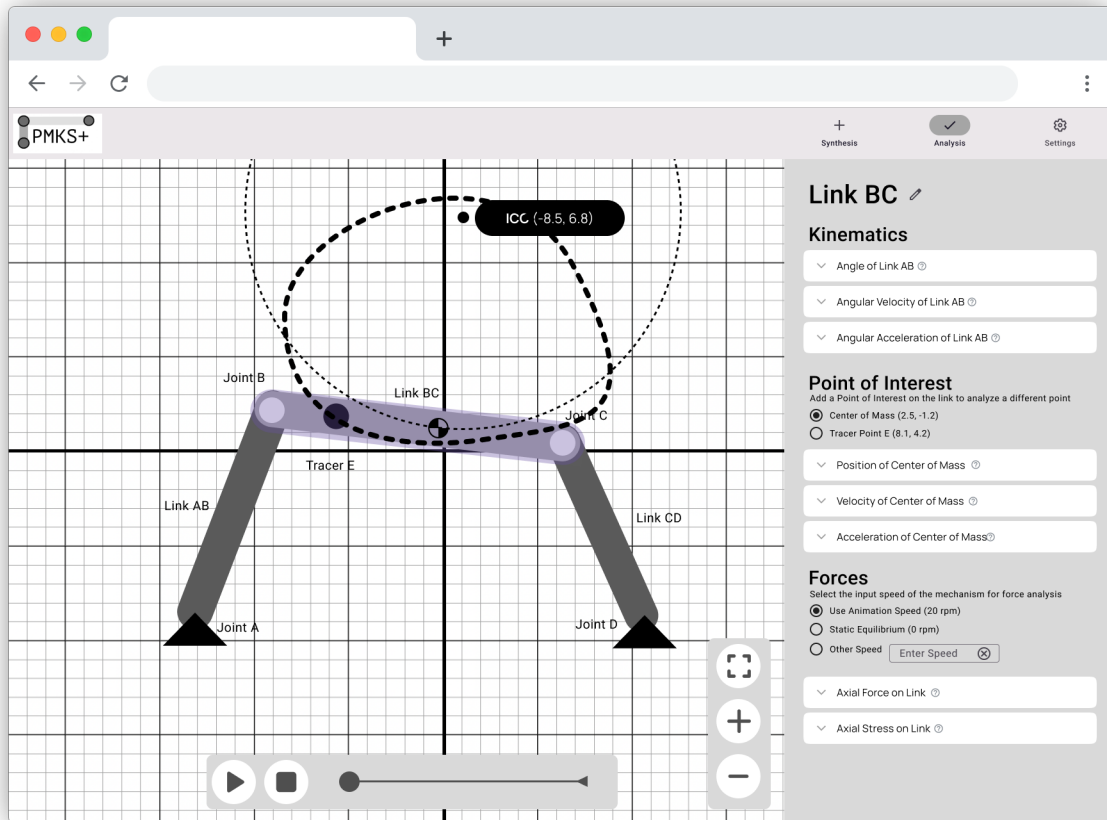
Constraining users to a strictly forward progression renders each step more final and challenging to undo. This limitation is particularly detrimental to persona 2 type users, as it obscures the possibility of alternating between editing and analysis to fine-tune linkages.

Previous versions also suffered from inconsistency in design elements, such as button sizes, colors, border radii, and text styles. To facilitate a more accessible and intuitive user experience, a design framework was adopted.

#### 4.2.1 Adherence to Google's Material Design guidelines

Given that the target users frequently interact with Google services, the Material Design guidelines were selected to provide a cohesive framework for constructing visually appealing

and functional interfaces [12]. This approach aims to minimize the learning curve by enabling users to rely on pre-existing knowledge of UI conventions.



**Figure 32:** Mockup three integrates standard Material Design elements for a consistent appearance. The top right contains 3 modes, synthesis, analysis, and settings (edit) where each panel changes based on what object is selected.

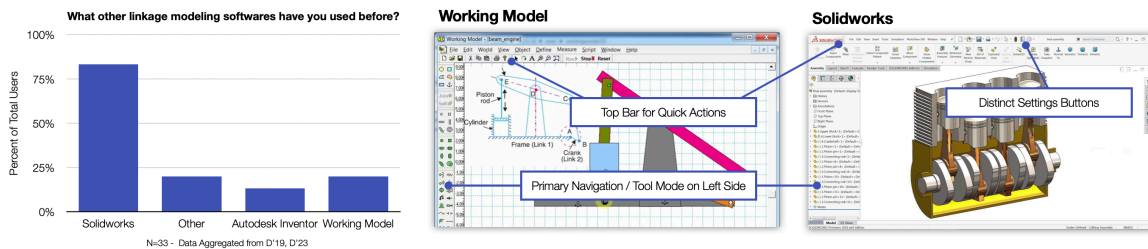
The three distinct modes are retained, but the directionality is removed to allow for more flexible mode selection and encourage exploration and trial-and-error. Figure 32 also introduces a concept for synthesis using the 3-position method and permits the right panel to be closed by clicking the active tab button, thus maximizing grid visibility.

Nonetheless, some issues arose with these modifications. For instance, the content of the tabs on the top right would change depending on the selected item. When the background was selected (i.e., clicking on the grid), the analyze tab would display whole-linkage analysis (e.g., degrees of freedom and loops). In contrast, selecting an object would show object-specific analysis (e.g., kinematic properties throughout the simulation). Similarly, the settings panel would display linkage settings (e.g., units and input direction) when the grid was selected and

object properties (e.g., position and grounding) when an object was chosen. This dynamic change in panel content confused some users, as demonstrated by a user struggling to locate the joint position input field when the settings panel was opened with no object selected.

#### 4.2.2 Incorporation of conventions from other engineering applications

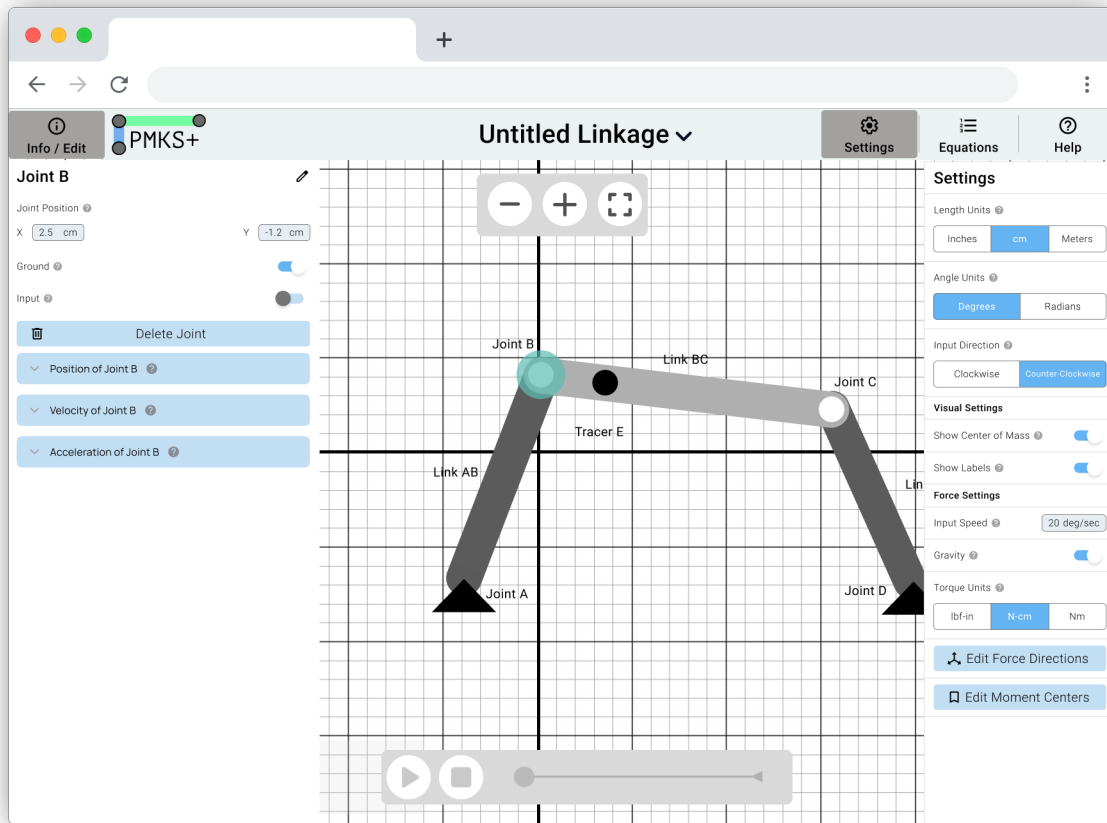
To address these issues, a survey was conducted to identify other engineering applications frequently used by the target users. By adopting conventions (pictured in Figure 33) from these applications, the transition to the new design was expected to be more seamless.



**Figure 33:** (Left) A survey of our users shows that they are most familiar with Solidworks and Working Model. (Center) Working Model puts its primary interactions on the top and left. (Right) Solidworks also puts its tree and action on the left and top but also has a dedicated settings button.

The survey results revealed several key differences between the mockup and other applications. Most notably, the primary information display in other applications was located on the left side. Moreover, the settings panel in these applications was distinct from the tabs that edit properties. Consequently, the primary panel was relocated to the left, and the settings panel was separated from the property-editing tabs, utilizing the now available space on the top right.





**Figure 34:** Mockup four moves the context-dependent panel to the left while keeping static secondary panels like settings and equations on the right. The edit and analysis information are now stored in the Analyze/Edit panel on the left.

In this version shown in Figure 34, primary actions for analysis and editing were moved to the left, while secondary actions were assigned to the right. Dividing the panels between the left and right sides of the screen allowed users to view information from both panels and the grid simultaneously.

This layout served as the basis for the final general interface. With the locations and interactions of all elements established, the design and size of each element were refined.

### 4.2.3 Toolbar and File Options

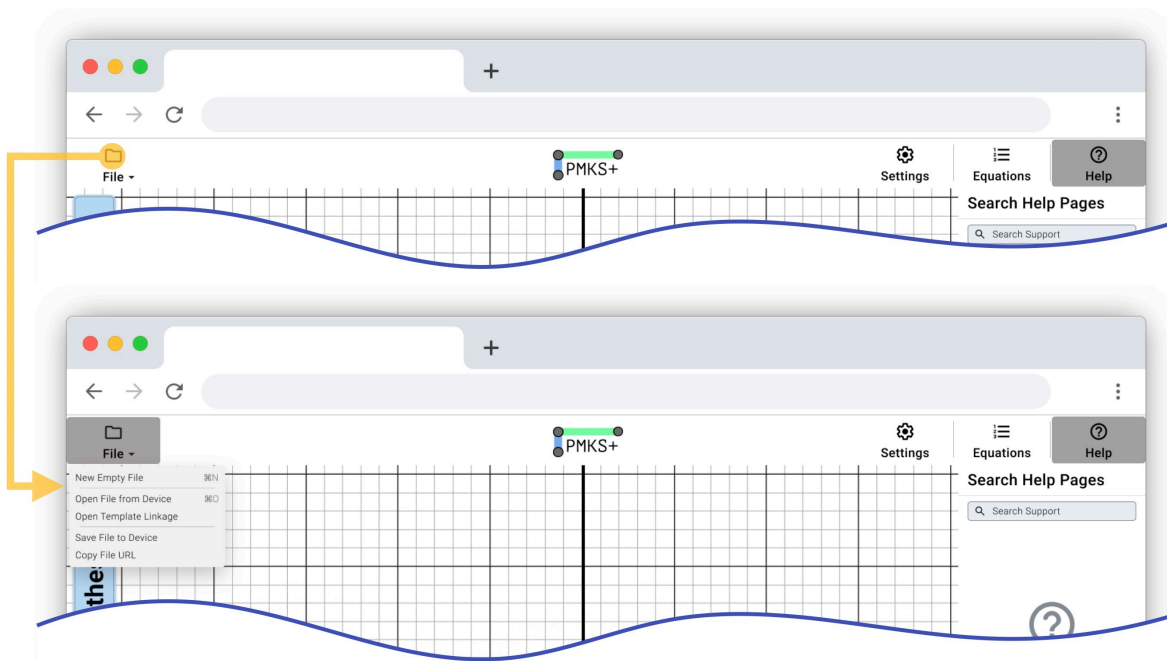
First, a list of actions currently being performed using the toolbar was compiled into Figure 35.

### List of actions performed in the toolbar

- Open File
- Save File
- New File
- Share File (Copy URL)
- Open Example (Template)
- Restart Tutorial
- About page / Help Videos
- Bug Report
- Export Data

**Figure 35:** A list of actions located in the toolbar before the redesign. Yellow items can be moved to the settings/help panel while exporting data will get moved into the analysis panel.

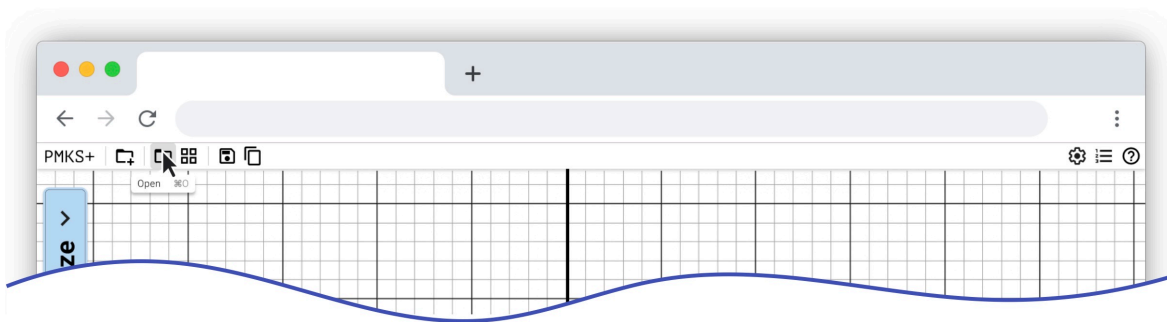
Compared to the original PMKS+, many of these actions have been relocated to the settings panel and analysis panel, streamlining the number of toolbar actions. The first design started by examining other applications users are familiar with. Traditional desktop applications, which utilize a standard File - Edit - View top bar structure, were considered in the redesign. Since all required actions fit within the file category, only the *File* button was added. This mockup is shown in Figure 36.



**Figure 36:** The first toolbar mockup uses a *File* button on the top left which, when clicked on, shows a list of file-related actions.

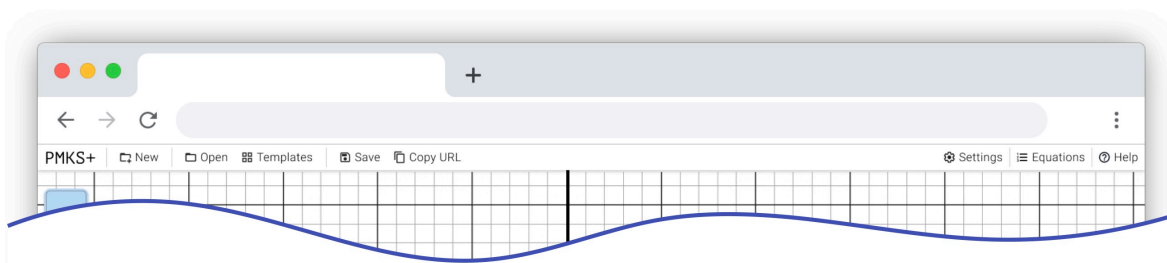
In a simulated browser, however, the top toolbar appeared too large and occupied excessive grid space. Furthermore, the additional click needed to reveal the list of options seemed

unnecessary given the available space and the small number of options.



**Figure 37:** The second toolbar mockup flattens the menu by putting all 5 icons on the toolbar. A hover shows the label for the icon along with the shortcut.

This version (shown in Figure 37) presented a flattened file menu with icons representing the available toolbar actions. Tooltips displaying text labels appeared when the user hovered over the buttons.

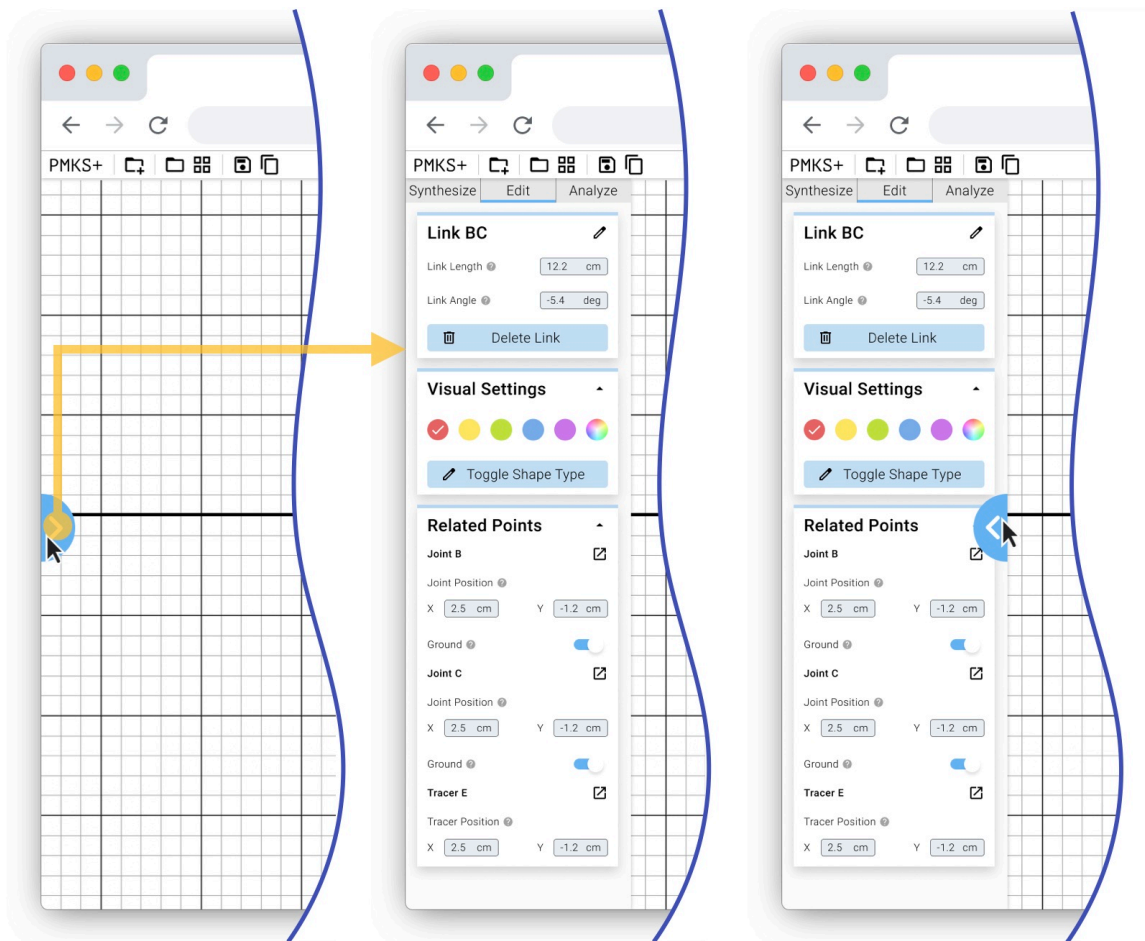


**Figure 38:** The last toolbar mockup shows both an icon and label in a thin toolbar with buttons justified to the left.

As it was challenging to create clear, descriptive icons for each action, the final mockup (shown in Figure 38) utilized buttons with both icons and labels. This design ensured a more intuitive user experience by allowing users to quickly identify the function of each button while maintaining a sleek and organized interface.

#### 4.2.4 Tab Designs

The subsequent challenge was determining how users should switch between the three primary modes of the application: Synthesis, Edit, and Analyze. The initial design featured three horizontal tabs, with the selected tab being highlighted:

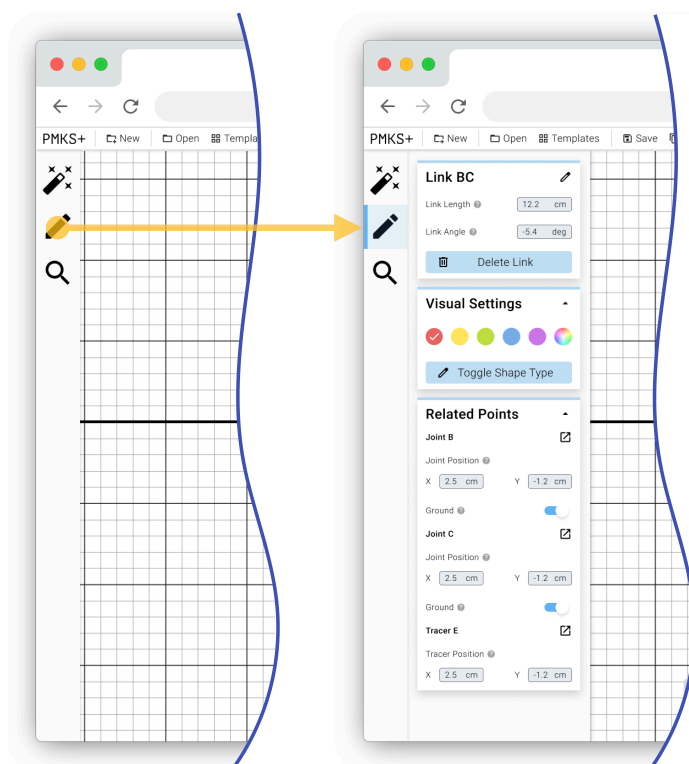


**Figure 39:** The first left panel design makes the entire panel hideable by clicking on the blue arrow. Upon clicking the blue arrow, the tab drawer opens from the left. Hovering over the right edge reveals a blue arrow pointing left, allowing the panel to be closed.

In this design (Figure 39), users can press a button on the left side of the screen to reveal a left panel. Once open, the top bar contains three tabs for switching between the three modes. To hide the left panel and create more space on the grid, users can hover their mouse over the right edge of the panel, which makes a button appear that can be used to hide the panel.

This design meets the two requirements of having a hidable panel that allows switching between the three modes, but it has several drawbacks. First, the button to expand and hide the panel is small and not visible enough for a panel containing nearly all of the app's features. Second, because the tabs are spelled out horizontally, they are too small, and users might not notice the different modes or their current mode. Lastly, three rows of buttons are located in the top left of the screen, potentially overwhelming users and inhibiting their ability to use muscle memory to remember the series of actions in the top left corner.

To address these issues, a vertical tab was added on the left-hand side of the screen as shown in (Figure 40):

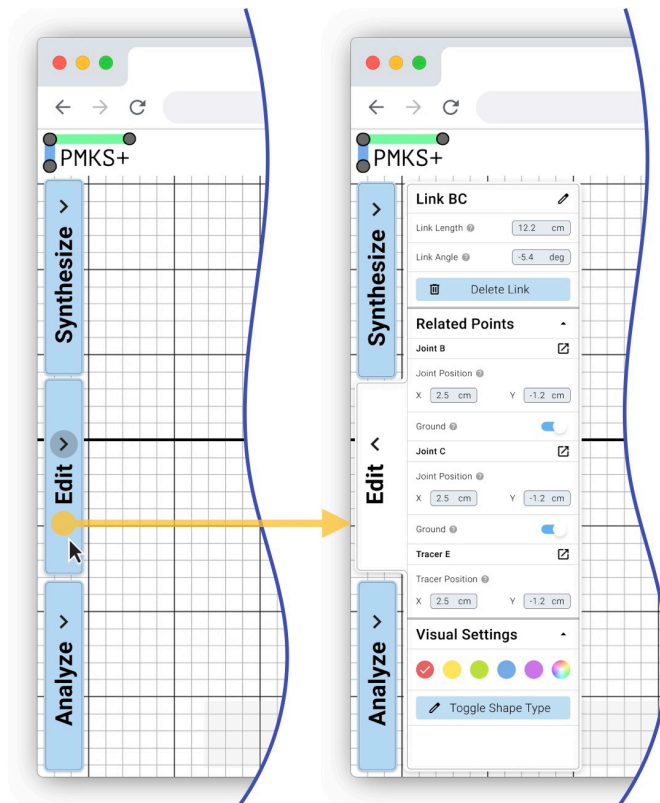


**Figure 40:** The second left panel design converts the three modes into large icons on the left. In descending order, the first icon represents the synthesis tab, the second represents the edit tab, and the third represents the analysis tab.

The three primary panels—Synthesize, Edit, and Analyze—are represented by icons, with tooltips displaying labels upon hovering. When users click on any of these icons, the corresponding panel expands, and the button is highlighted, indicating an open panel. Clicking on the currently open panel's button hides the panel again.

The primary issue with this version is that the buttons could be confusing, as there is no ideal icon to describe synthesizing a linkage or creating analysis graphs. Additionally, the buttons are not prominent or enticing enough for such a core feature of the app. Finally, there is a considerable amount of wasted white space on the left toolbar, taking up grid space without serving a purpose.

The final version pictured in Figure 41 resolves all of these issues:



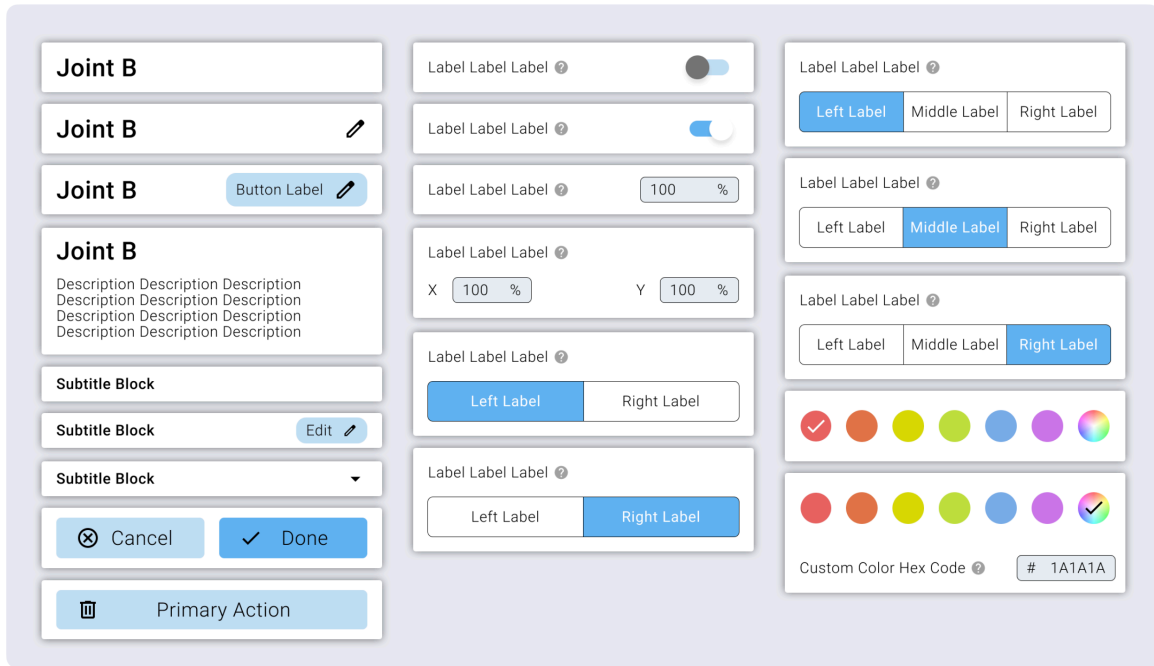
**Figure 41:** The third and final left tab design expands the three icons into tabs that take up the entire vertical space on the left. When the mode is selected the vertical 'tab' button connects to the panel that is opened and changes color.

This version spells out the three primary actions vertically on the left along three tabs. By including rotated labels for the tabs, only a small horizontal portion of the left screen is occupied, while ambiguity about the buttons' functions is eliminated. The large, bright blue buttons appropriately signal the importance of these buttons for the app's functionality. Expansion indicators on the tabs suggest that these tabs contain more content. The tab color changes significantly when opened, and a skeuomorphic 'tab-in-a-file-folder' design connects it to the expanded menu, making it clear which tab is currently open. Lastly, the expansion arrow switches direction when a tab is open, indicating that it can be closed.

Ultimately, the tab design reflects the design goal of making the app feel intuitive. These subtle visual cues rely on users' pre-existing knowledge, ensuring they have a good idea of how to interact with the tabs as soon as they open the application.

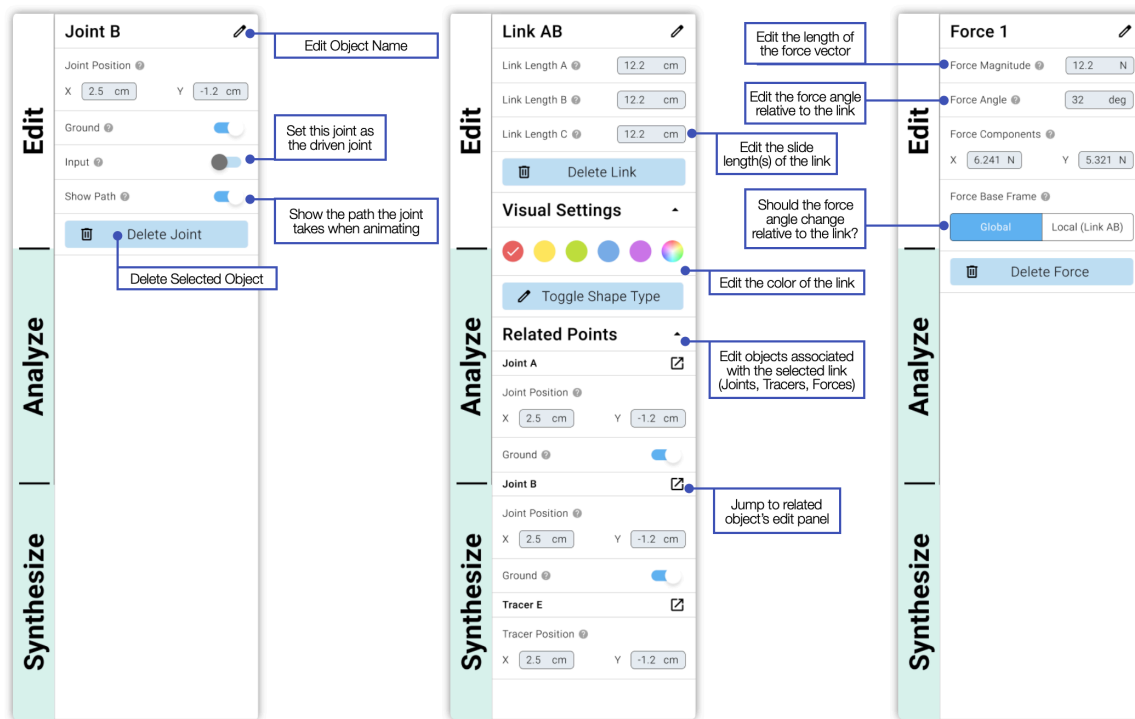
#### 4.2.5 Edit Panel Design

To develop the interface of the panels, a set of standard input components (shown in Figure 42) was needed for use throughout the app. Based on Google’s material design guidelines, a collection of components with multiple variants for each was designed. These components can be combined to create any UI element. An example is shown in Figure 92.



**Figure 42:** Several input blocks and minor variations of ‘blocks’ were created as a modular way to build interfaces. From top to bottom then left to right: Title Block, Subtitle Block, Action Block, Toggle Block, Input Block, Dual Input Block, Radio Block, Color Block

These blocks were designed to have a consistent aesthetic, ensuring that UIs feel cohesive while being completely modular, allowing blocks to be arranged in any order while maintaining a visually pleasing appearance. By combining a series of blocks vertically, the edit panel interface was created:



**Figure 43:** (Left) Edit Panel when a Joint / Tracer is selected. (Center) When a Link is selected. (Right) When a force is selected.

As shown in Figure 43, regardless of the object (Joint, Link, Force) selected, the name of the object is displayed with a pen icon for editing the object's name. There is also a button to delete the selected object.

When the edit panel is open and a joint is selected, a dual input block allows users to see and edit the joint's position. Three toggles enable the joint to be grounded or ungrounded, set as the input, or toggle the visibility of the path of a joint when simulated.

When a link is selected, all external side lengths are visible and editable<sup>2</sup>. Visual settings allow the color of the link to be changed, and the same type can be toggled (discussed in subsection 4.2.8). There is also a list of related objects, such as joints that are part of the link, forces acting on the link, and tracer points, with a truncated menu for editing the most common options for each object.

Lastly, when a force is selected, input boxes allow users to edit the force, angle, and x and y components, as well as the reference frame for the force angle.

<sup>2</sup>Link AB is a triangle, so there are three side lengths that can be edited.

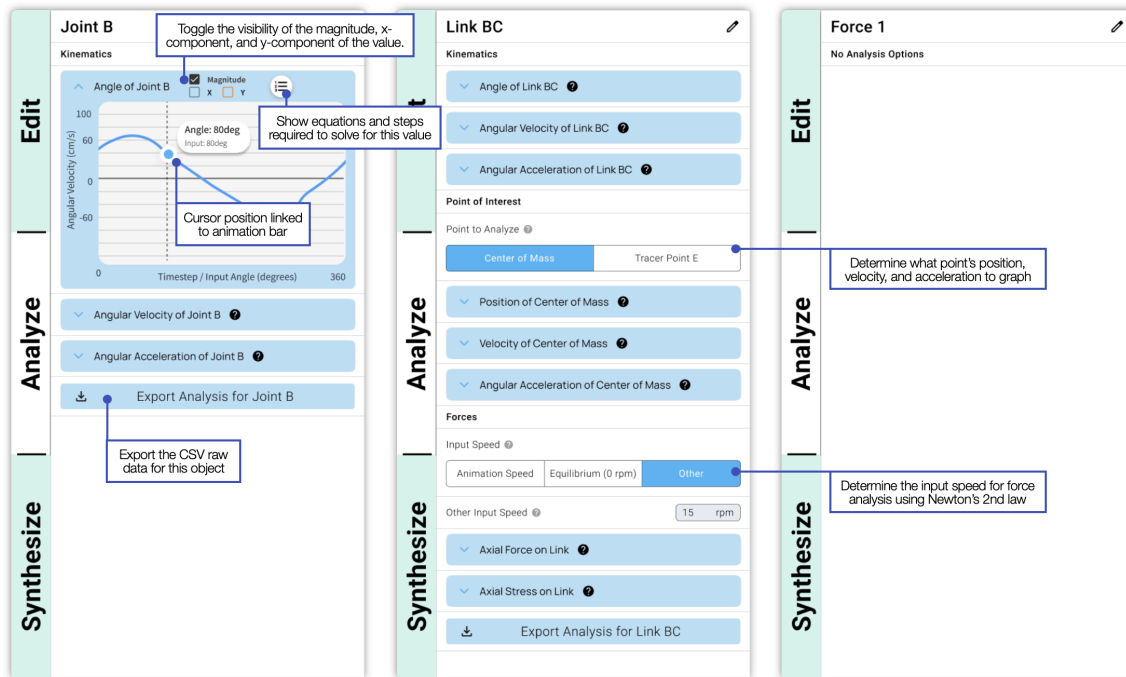


#### 4.2.6 Analyze Panel Design

To create an analysis panel that followed the design goals, a list of available analysis graph options needing implementation in the application was compiled:

- X and Y Position of a Joint
- X and Y Component and Magnitude of a Joint's Velocity
- X and Y Component and Magnitude of a Joint's Acceleration
- X and Y Component and Magnitude of the forces experienced by a Joint
- Input Torque at a Joint
- The angle of a Link
- The angular velocity of a Link
- The angular acceleration of a Link
- X and Y Position of a Link's Center of Mass
- X and Y Component and Magnitude of a Link Center of Mass's Velocity
- X and Y Component and Magnitude of a Link Center of Mass's Acceleration
- The axial force experienced by a Link
- The axial stress experienced by a Link
- X and Y Position of a Tracer
- X and Y Component and Magnitude of a Tracer's Velocity
- X and Y Component and Magnitude of a Tracer's Acceleration
- X and Y Component and Magnitude of the forces experienced by a Tracer

The first step in simplifying the analysis process and ensuring that all these graphs were easily accessible was to split them by object type. Unlike the current version of synthesis, where users find the graph they are looking for by selecting through a series of dropdowns (See Figure 23), the click-to-select approach significantly narrows down the options.



**Figure 44:** (Left) Analysis Panel when a Joint / Tracer is selected. (Center) When a Link is selected. (Right) When a force is selected.

As shown in Figure 44, when a user clicks on an object, the relevant analysis graphs pertaining to that object are displayed. The list of graph headings is split into subsections: Kinematic, Center of Mass, and Forces. Each of these graph headings can be clicked on to expand, which will show the graph. When a graph is expanded (if available), the user can toggle a checkbox to hide the X and Y components as well as the magnitude lines. Clicking the equations button in the graph will open the right-hand panel, which displays a series of steps and equations used to derive the graph. An export analysis option allows users to download a CSV file with the raw data used to create the graphs or images of the graphs.

This design approach for the Analyze Panel makes it more user-friendly and intuitive, enabling users to easily access and interact with various analysis graphs based on the selected object. The panel design ensures that the application remains organized and visually appealing, while providing a comprehensive set of tools for users to work with. Overall, the Analyze Panel design aligns with the design goals by creating a seamless and efficient user experience.

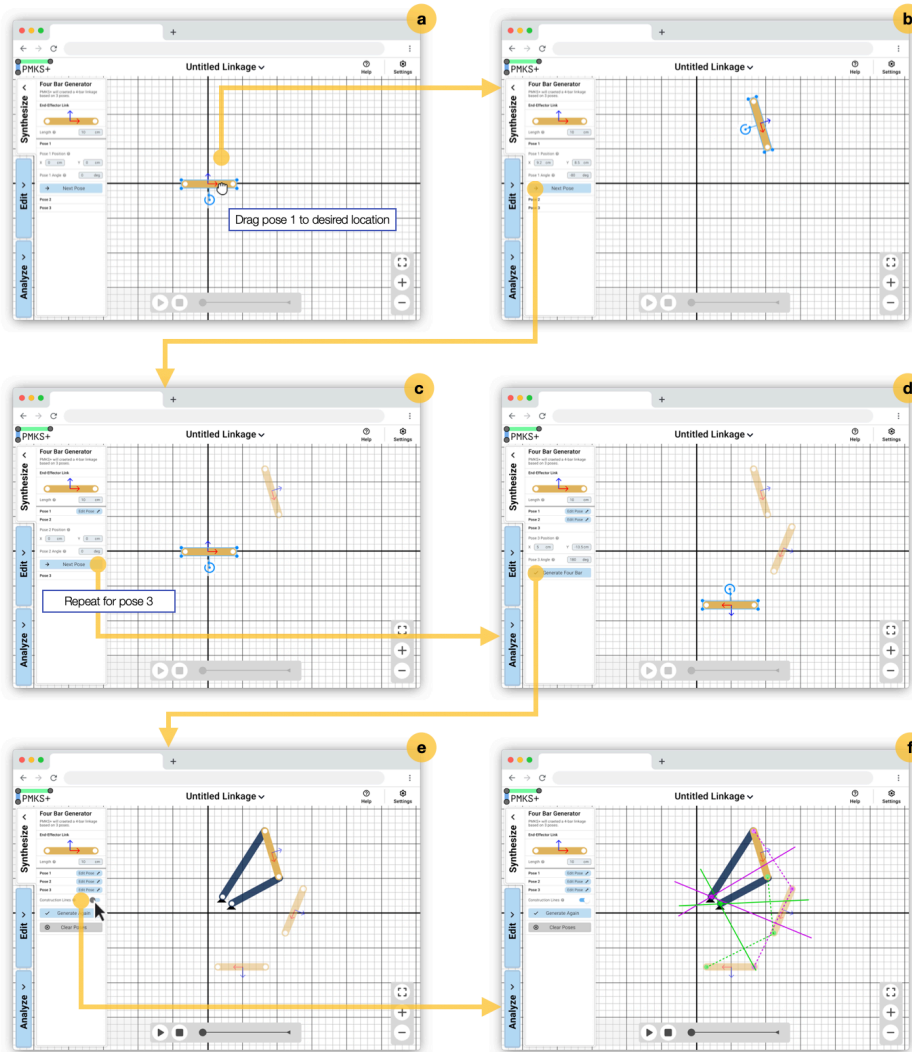
#### 4.2.7 Synthesize Panel Design

The last of the 3 left tabs is the synthesis section. Unlike the other two, this panel is not context-dependent, meaning it does not change no matter what object is selected. ?? shows how a user might start with a blank file and have PMKS+ synthesize a four-bar using 3 position synthesis.<sup>3</sup> This was the only synthesis option considered since it is the most common one taught at WPI.

---

<sup>3</sup>3 Position Synthesis asks a user for 3 poses for the coupler of a four-bar to compute it. More information: <https://www.irjet.net/archives/V4/i5/IRJET-V4I5297.pdf>

## Synthesis Panel Flow Diagram



**Figure 45:** From Left to Right, then Top to Bottom: (a) Entering synthesis mode, a draggable coupler appears. (b) The first coupler is positioned correctly, and the *Next Pose* button is pressed. (c,d) Position the other two poses until all three positions are defined. (e) Click *Generate* to create a four-bar. (f) Toggle the construction lines to view the perpendicular bisectors.

### 4.2.8 Infinitely Customizable Shapes

A major pain point to address for this redesign (Discussed in subsection 3.5) was the lack of shape customization and options and the difficulty of defining joint positions by link lengths. The intent behind the current solution (shown in Figure 26) can be found in Section 8.1 of the 2020 PMKS+ report [4].

To determine the types of shapes required by users, sample problems from different textbooks and homework problems assigned to previous ME and RBE classes were analyzed. The shapes used were then categorized into various general categories and ranked by their frequency of use:

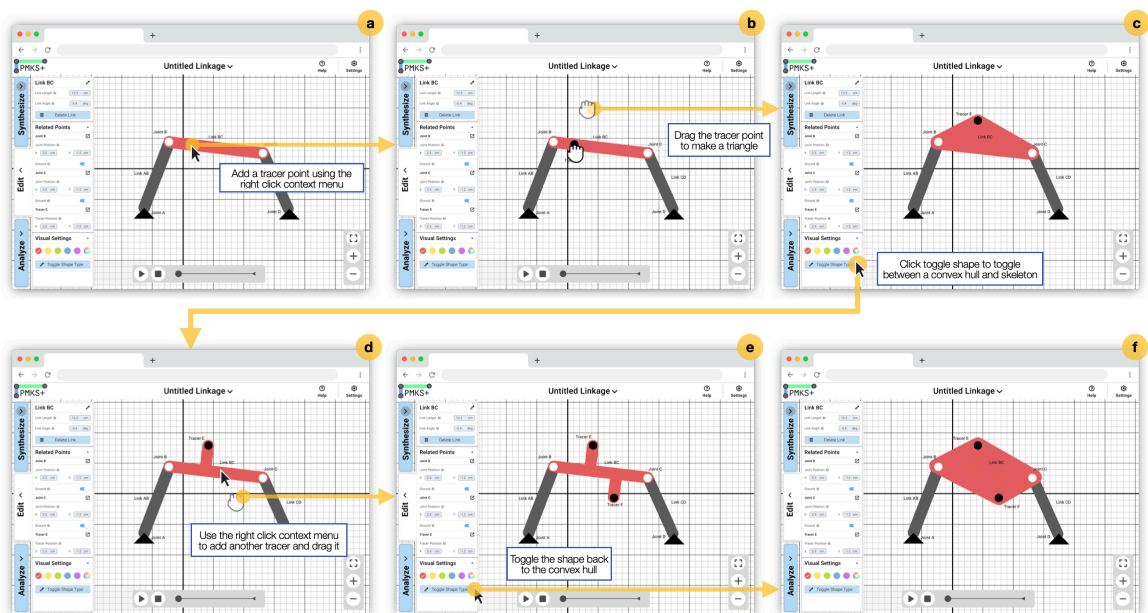
1. Binary Links (Bars connecting two joints)
2. Polygons (Convex Hulls)
3. Circles
4. Letter Shapes (Ts, Us, Cs)
5. Gears
6. Cams / Rolling Contact

Based on discussions with stakeholders and the context of other related projects<sup>4</sup>, the focus was directed toward developing a custom shapes system that could handle the first 4 types of shapes.

The first iteration of the shapes system (shown in **??**) allows users to toggle between two shape modes: convex hull and skeleton:

---

<sup>4</sup>Another student in our lab is working on a gear train generation program



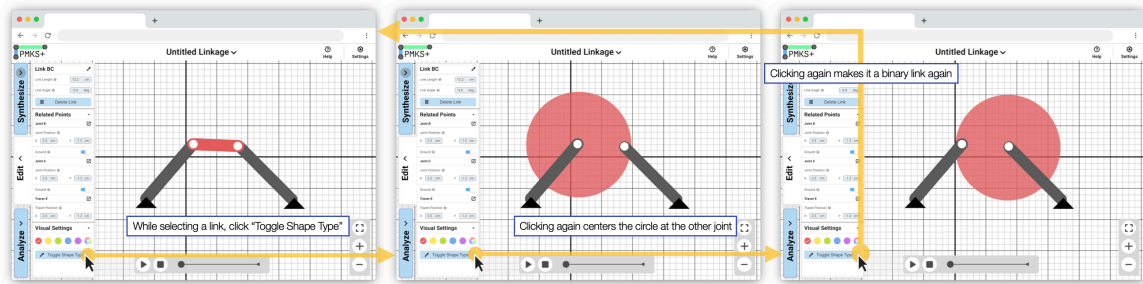
**Figure 46:** Left to Right, then Top to Bottom. (a-c) By default, links are in convex hull mode, so when a third tracer is created in the red binary link and dragged up, the red link becomes a triangle. (d) Clicking *Toggle Shape* puts the link into skeleton mode. (e) Adding another tracer results in another line appearing perpendicular to the primary axis. (f) Clicking *Toggle Shape* returns the link to convex hull mode where the four joints define the outer shape of the link.

In this design, users can create shapes by adding additional links to existing ones. By default, links are in convex hull mode, so when a user creates a third link connected to the red binary link and drags it up, the red link becomes a triangle (Figure 46). In skeleton mode, the link positions are defined by the length of the binary links, allowing users to create custom shapes by setting specific link lengths.

To further extend the customization options for shapes, we introduced an interface for users to create circles and letter shapes. This enables users to create a wider variety of shapes for their mechanisms, which can be particularly useful for specific applications or when replicating real-world mechanisms.

This system allows users to make all polygons and most letter shapes. However, there are two major issues. The first issue is the algorithm for defining the skeleton shape. The system finds the primary axis and makes a line perpendicular to the main line to connect all other joints. The primary line for the skeleton seems obvious (the line connecting the two white dots); however, as users start creating links connected to more than two other links, it gets more complicated for the program and the user to identify the primary line. The second issue is creating circles. As shown in Figure 47, this is done by clicking the toggle shape button

while selecting a link with two joints:

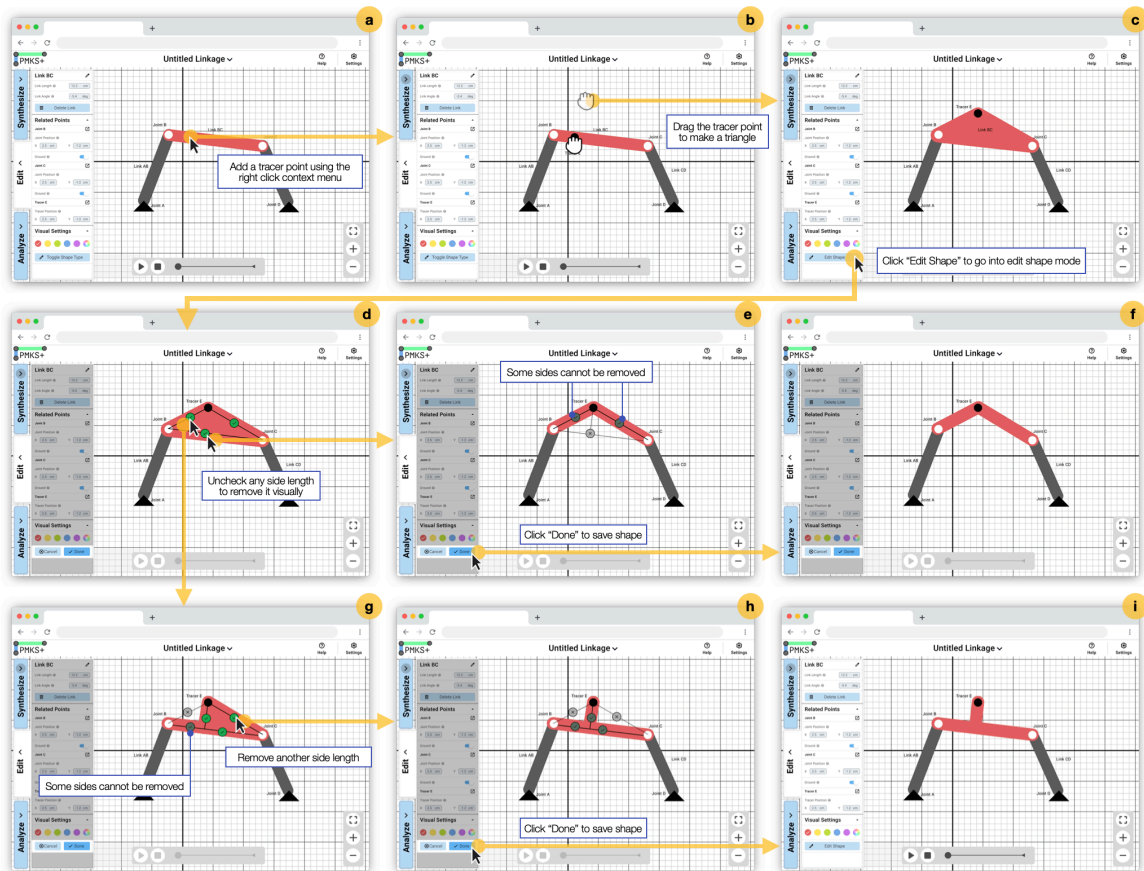


**Figure 47:** When a link has two joints, the *Toggle Shape* button cycles the shape between a binary link, a circle centered at the first joint, and a circle centered at the second joint.

Incorporating circles into the system adds complexity since the behavior of the button changes based on the number of joints in the selected link. Additionally, the number of shape states increases from 2 to 3 depending on whether the user is in circle or non-circle mode.

The most significant limitation of this approach is the restricted range of shapes that can be created. Given that each set configuration of joints only has two possible shapes, shapes like L or V cannot be created, along with many other simple-to-describe concave shapes.

An alternative solution was developed to offer more customizability:

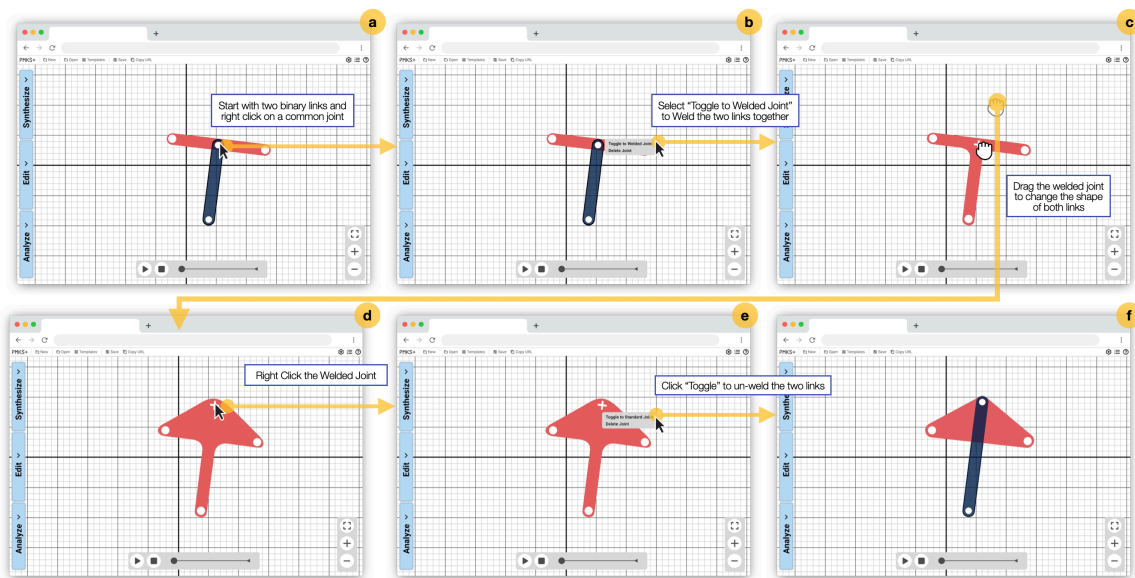


**Figure 48:** Left to Right, then Top to Bottom. (a-c) Similar to Figure 46, a tracer is added to a binary link and dragged up to create a triangle (a convex hull of the three joints). (d) When the *Edit Shape* button is pressed, the application switches into a new mode where lines with checkboxes appear over the outer edges of the shape. (e) Deselecting the bottom edge removes that edge. (f) Clicking *Save* saves the inverted V shape as the new link shape. (g) Starting at step d, if the top-left edge is deselected, some edges are now required to be selected (greyed-out checkmark) to keep all joints connected. (h) When the top-right side is deselected, an inverted T shape remains. (i) Clicking *Save* saves that shape.

This solution (shown in Figure 48) allows the creation of virtually any non-curved shape by toggling the visibility of sides. However, introducing a different mode to accomplish this adds an unnecessary layer of complexity. Implementation would also be challenging since the sides that can or cannot be removed need to be calculated as the user changes the shape. Furthermore, since users are allowed to drag joints, it would be difficult to maintain the state of toggled sides while the entire shape changes.

Consequently, a final alternative (Figure 49) was devised to reduce complexity while still increasing the possible number of shapes that can be created:



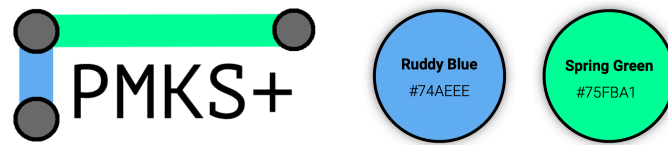


**Figure 49:** From Left to Right then Top to Bottom. (a) Starting with a blue binary link and a straight 3-joint link, right-click on the shared joint. (b) Click *Weld Joint* on the context menu. (c) This combines the links into one shared T-shaped link. (d) When the center joint is dragged up, the shape changes as if the two base shapes are being dragged. However, it still appears as one red link with internal corners welded. (e) Right-click on the center joint and click *Unweld Joint*. (f) The two shapes are left with the blue binary link and a red triangular link.

By welding and unwelding simple links at the joints, users can combine basic shapes to create any shape that is a combination of the two. This solution addresses many of the previously mentioned issues: It does not require the user to enter a different mode but instead allows them to use the familiar context menu to Weld Joints. Recomputing the shape when a joint is dragged is relatively easier since it's a simple union of shapes. Additionally, this approach does not necessitate tracking the primary axis. This final solution was selected to enable users to customize shapes.

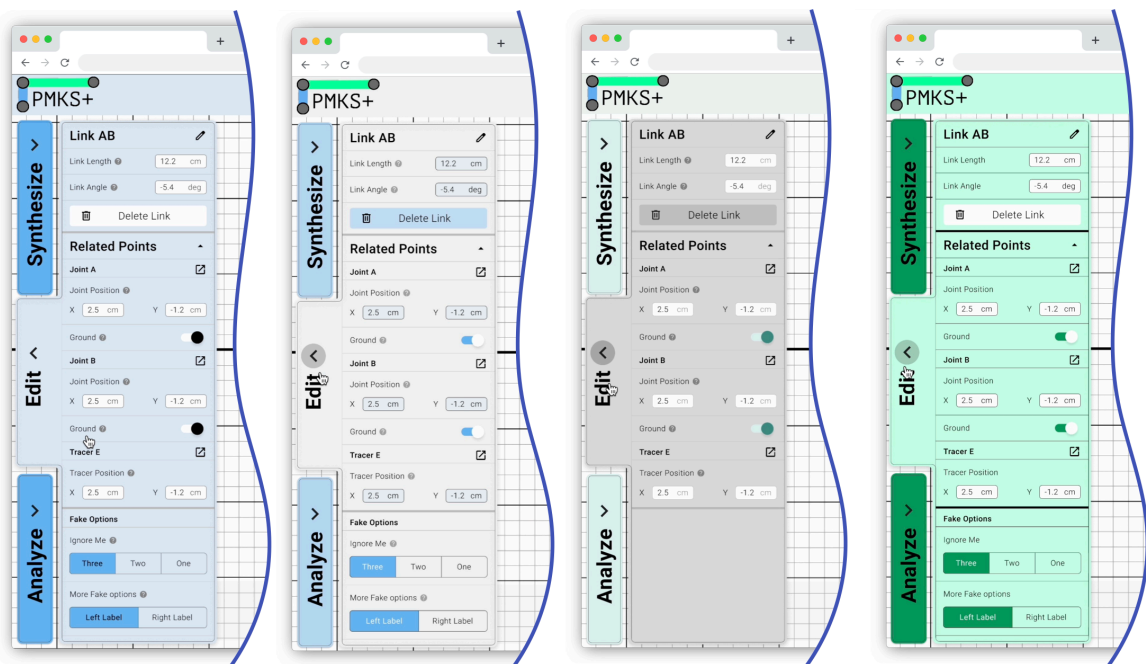
By providing users with this simple yet powerful ability to weld joints, the pain point of limited shape customization and difficulty in defining joint positions by link lengths is addressed. This enhancement to the application not only improves user experience but also expands the tool's capabilities for a wider range of applications and use cases. The custom shapes system effectively caters to the needs of users working with binary links, polygons, and letter shapes and is designed to be intuitive and user-friendly.

## 4.2.9 Color and Branding

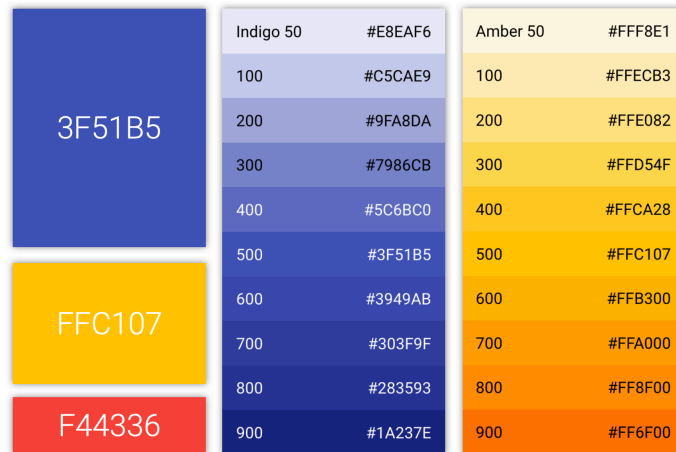


**Figure 50:** The current app's color palette uses Ruby Blue #74AEEE and Spring Green #75FBA1

The current app adopts these two colors from the Silverlight version of PMKS+ for its logo (pictured in Figure 50), but they are not used elsewhere in the user interface. Although attempts to use the current color scheme were taken (Figure 51), as a ground-up redesign of the interface was undertaken, it was decided to start from scratch and craft a new identity for PMKS+ that distinguishes it from the original PMKS.



**Figure 51:** Four different color schemes of the edit panel derived from the base PMKS+ colors.

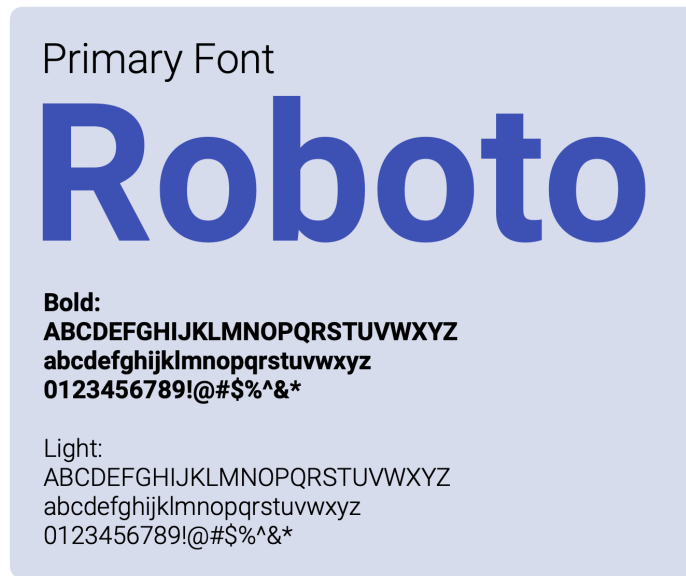


**Figure 52:** The redesigned color palette has a primary color of #3F51B5 and an accent color of #FFC107.

The new user interface of PMKS+ incorporates Indigo (3F51B5) as the primary color and Amber (FFC107) as the accent color, reflecting the organization’s philosophy of creating educational, flexible, collaborative, and accessible software. Indigo evokes feelings of stability, wisdom, and confidence, resonating with the educational and accessible aspects of PMKS+. As a deep, rich hue, Indigo conveys the reliability and trustworthiness of the software, encouraging students to focus on learning concepts and verifying their work. The redesigned color palette is shown in Figure 52.

Amber, the accent color, brings warmth, energy, and optimism to the design, aligning with the flexible and collaborative principles of the organization. This vibrant hue stimulates creativity and collaboration, inspiring users to explore various applications and work together on linkages. The combination of Indigo and Amber creates a harmonious balance between stability and excitement, capturing the essence of PMKS+.

The warning color (F44336) is used sparingly as a tertiary accent color or to signify a consequential or irreversible action. It is worth noting that all three of these color palettes are selected from the suggested Material Design V2 color palette, making the entire interface feel more familiar to brand-new users.



**Figure 53:** PMKS+'s Primary font is Roboto, a modern sans-serif font.

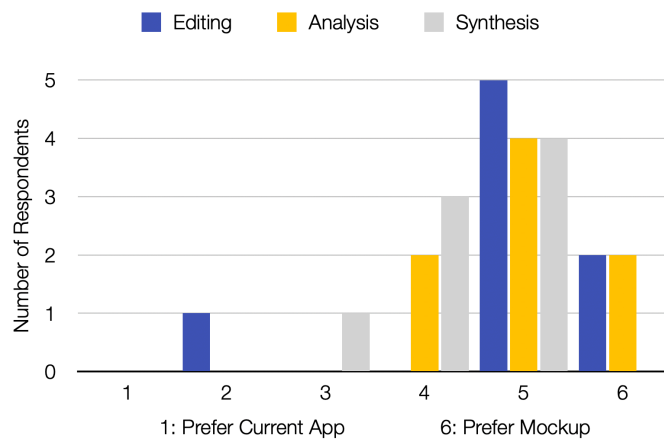
To complement the color palette, the Roboto font (shown in Figure 53) has been selected for its modern, clean, and approachable appearance. Roboto, a sans-serif font, ensures readability and versatility, enhancing the user experience and reinforcing the educational, collaborative, and accessible nature of PMKS+. The strategic use of colors and typography in PMKS+ reflects the organization's commitment to providing a user-friendly and engaging learning platform for students and professionals alike.

### **4.3 Early Feedback from Users**

As mentioned in previous sections, obtaining constant feedback from users was emphasized during this redesign. As soon as the mockup version of the redesign was completed, feedback was sought.

On October 10, 2022, feedback was gathered from users at Higgins Lecture Hall and the Innovation Studio Robotics Lab. These users had experience with the current PMKS web application. A demo of the mockup was presented, and quick interviews and questionnaires were conducted to gather their opinions. The participants included 16 WPI students from ME4320 (Mechatronics) and RBE2001 (United Robotics 1) courses.

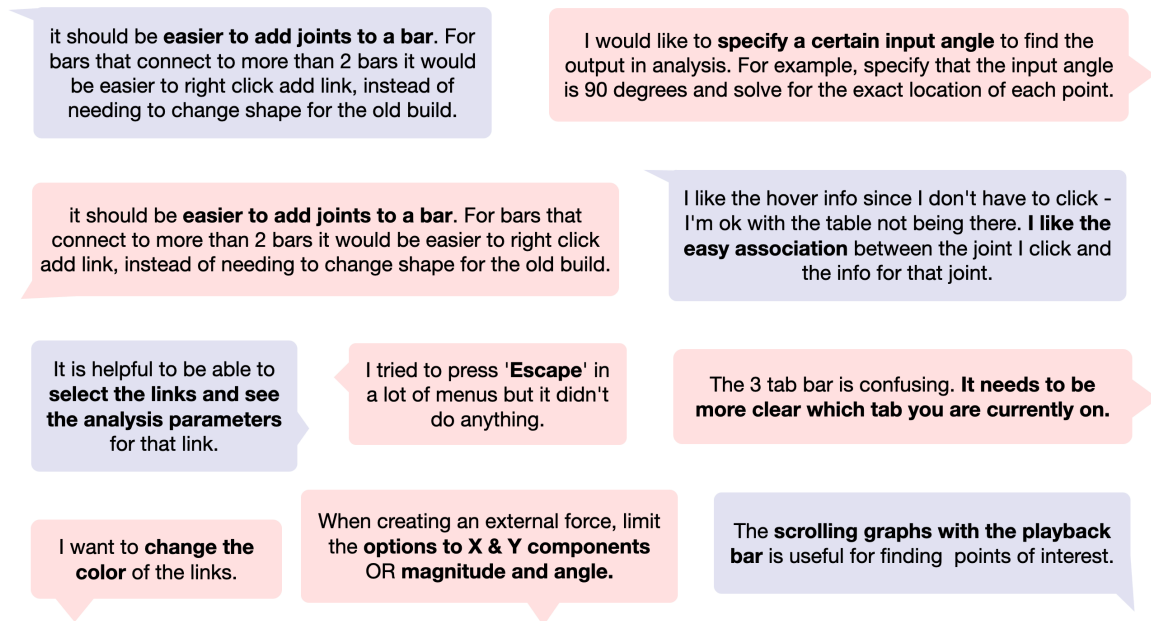
**Comparing the mockup with the current application:  
Which one is better for Edit, Analyze, and Synthesize**



**Figure 54:** A study in October 2022 asking 16 participants to compare the current PMKS+ app with the mockup shows an overwhelming preference for the mockup.

Overall, as shown in Figure 54, users showed a strong preference for the new editing options, scoring it 4.875/6. Similarly, all participants preferred the new analysis options, giving it a score of 4.375/6. For synthesis, students gave a 5/6. Note that a neutral score (indicating no preference between the PMKS+ and the mockup) would be 3/6.

More importantly than the generally positive feedback, numerous quotes and images were collected from this study, providing valuable insights into the users' experiences with the mockup. Figure 55 shows a collection of some of the most actionable and helpful feedback that was used to improve the mockup further as well as the actual application:



**Figure 55:** A curated list of written feedback received from the October 2022 study asking students to compare the current PMKS+ app with the mockup. A full list of feedback can be found in the Appendix.

#### 4.4 Simple Prototype Testing

Encouraged by the positive feedback, a basic prototype of the click-to-edit model was developed (shown in Figure 57), replacing the linkage table with a click-based editing panel. The primary goal was to validate the concept of selecting an object, which was the fundamental change in this redesign. To do this, feedback was sought as soon as the prototype was completed. The old PMKS+ interface (shown in Figure 57) was compared with the new version that incorporated the edit panel (Figure 57).

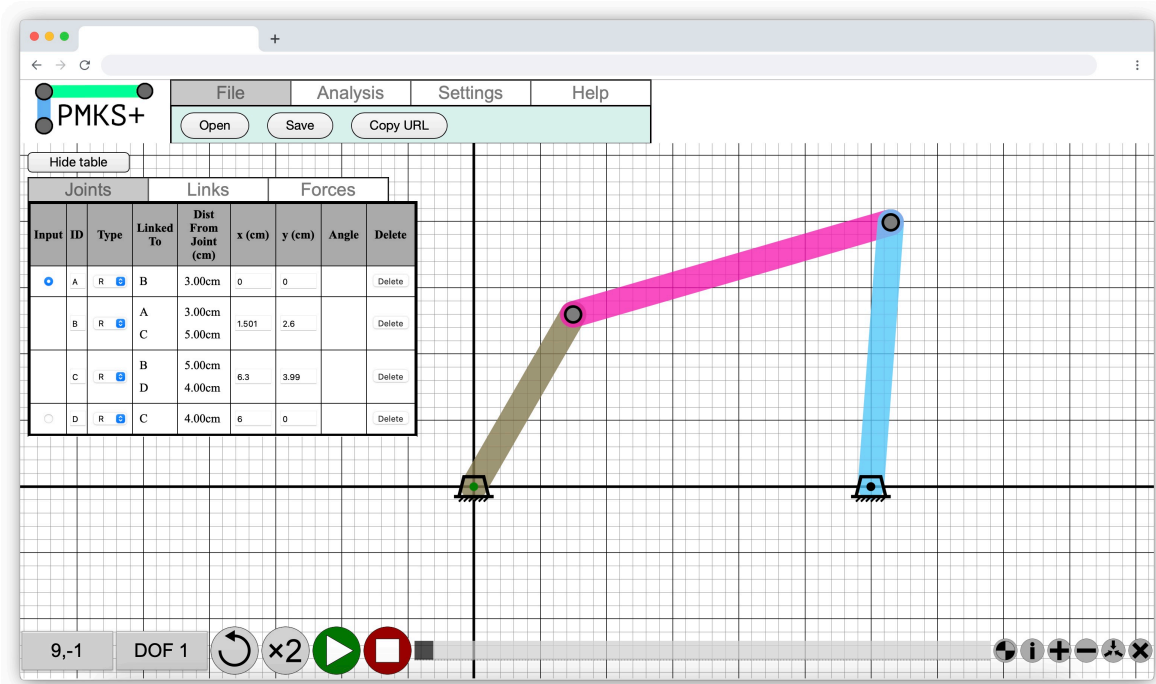


Figure 56: An assembled four-bar in the current version of PMKS+ before the redesign.

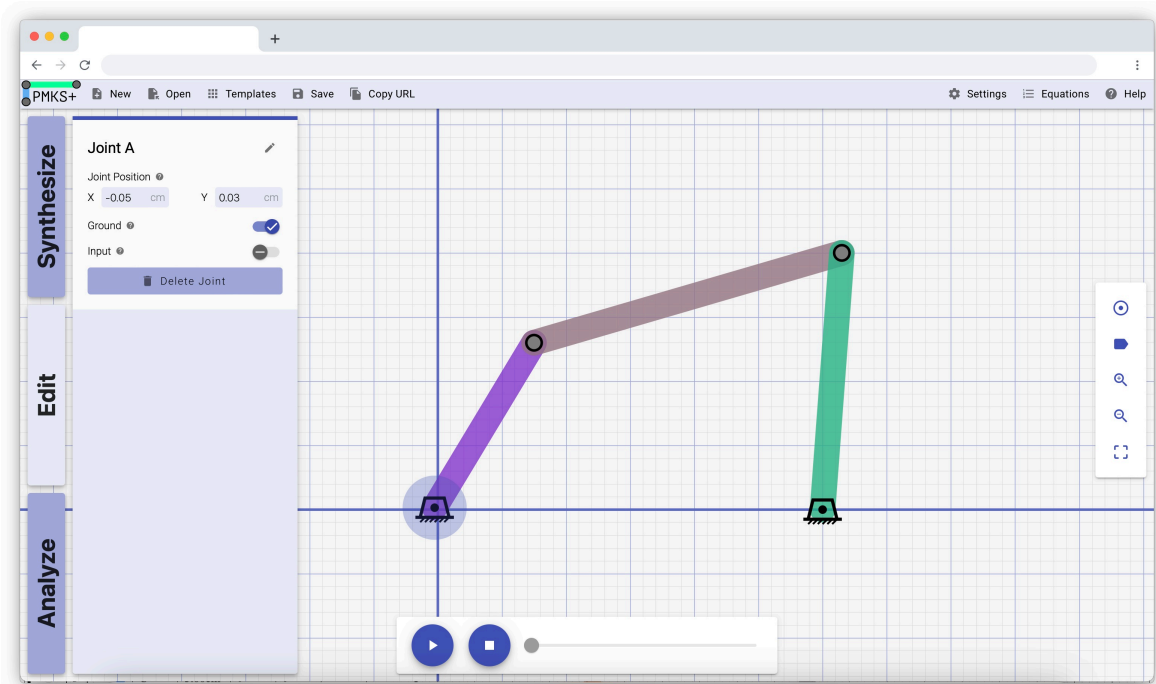
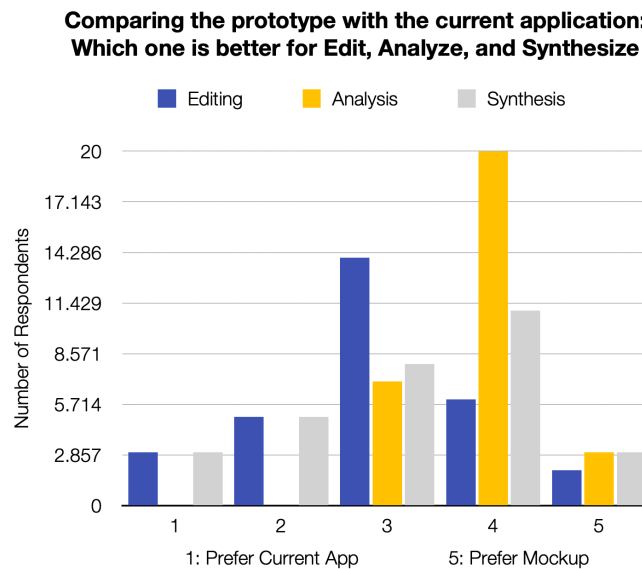


Figure 57: The same four-bar in the new prototype version of PMKS+.

This user test was conducted from December 11, 2022, to December 19, 2022. This test was carried out remotely, with participants receiving PDF instructions (available in the appendix) with a task and an online survey. The untimed task required testers to create a four-bar linkage based on the dimensions provided, then they were asked to perform some analysis tasks and submit the answers they got. They also had the option to provide screen recordings. The participants were 30 students from the ME4320 (Advanced Engineering Design Project) course, with 85% being senior RBE/ME/BME students, and the rest being juniors. A total of 17 optional screen recordings were received.



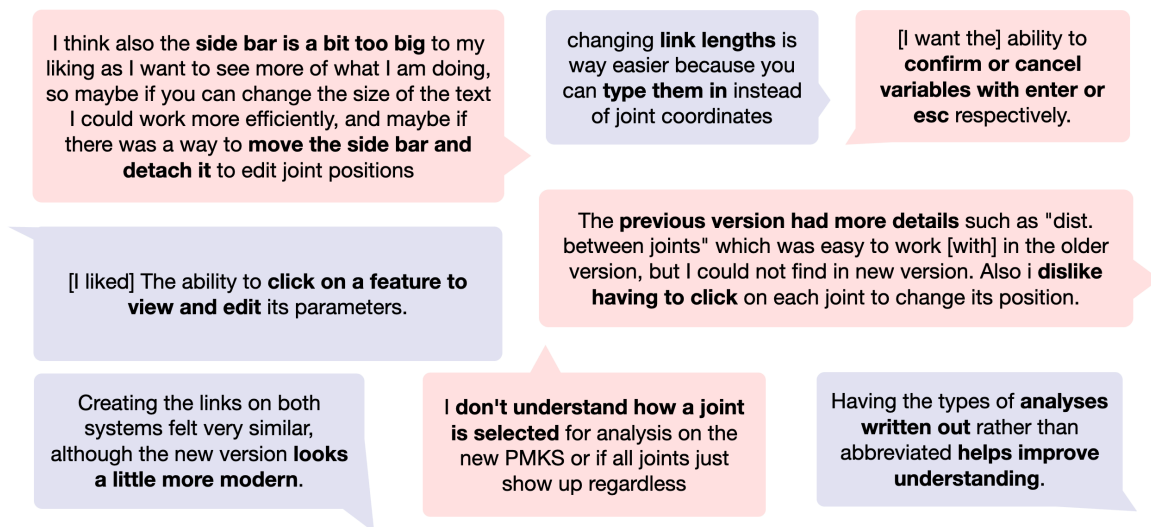
**Figure 58:** A study in December 2022 asked 30 students to compare the current application with the redesigned prototype. For editing, there was a very slight preference for the prototype. For analysis and synthesis, there was an overwhelming preference for the prototype.

The results of this user test (shown in Figure 58) indicated a slight preference towards the new editing process, with a score of 2.96/5, where a no-preference score would be 2.5. While analysis and synthesis were not implemented in the prototype, feedback echoing the results from the previous study was still received, with a score of 3.87/6 and 3.2/6 respectively.

Interestingly, the score for editing significantly decreased compared to when the mockup was presented. It is believed that the rating was impacted due to the number of bugs in the program rather than the actual interactions. Despite the number of bugs, the fact that students preferred the prototype over the current application was a positive sign. More importantly, valuable feedback was gathered from the questionnaire at the end and the screen recordings, which helped identify aspects of the user interaction that required improvement. Figure 59 is



a collection of key quotes that were used to further refine the app:



**Figure 59:** A curated list of written feedback received from the December 2022 study asking students to compare the current PMKS+ app with a prototype of the redesign. A full list of feedback can be found in the Appendix.

## 4.5 Feedback from Other Stakeholders

Throughout the term, bi-weekly meetings were attended with parts of the thesis committee, which included professors currently using PMKS+ in their courses and a UX expert. In each meeting, a progress report was presented, and feedback on the features was received. Discussions covered various topics, ranging from the selection of features to include, to the sizing and color of specific buttons and text labels.

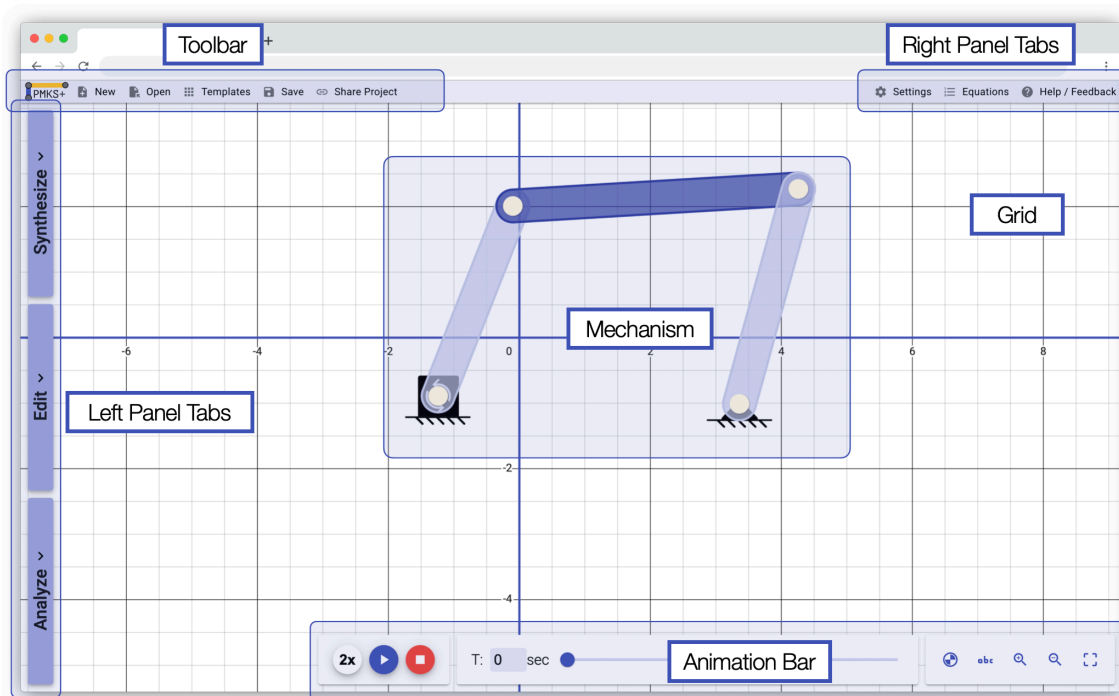
Additionally, consultations were held with developers who had previously worked on the project for major architectural changes, as well as unaffiliated software developers. This was done to ensure that the redesigned code would be as maintainable as possible and easy to work with for future development.

## 5 Final Application

This section presents the features of the redesigned application, offering a comprehensive overview of the software.

### 5.1 Overview of Redesigned Application

The redesigned application, shown in Figure 60, maintains a familiar interface, featuring a grid that occupies the entire screen, accompanied by a toolbar at the top and an animation bar at the bottom. The toolbar provides quick access to common tasks, while the animation bar allows users to control the simulation of the linkage animation and their view of the mechanism. In contrast to the original linkage table, the updated version incorporates three tabs on the left and three buttons on the right that open corresponding panels.

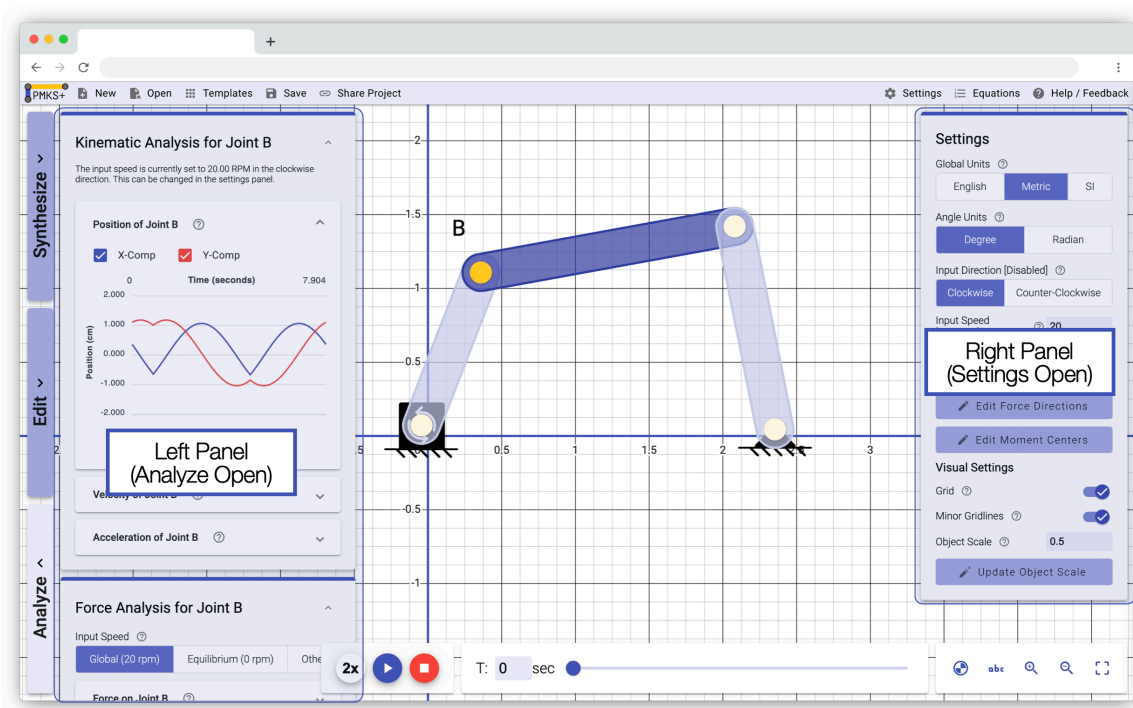


**Figure 60:** The redesigned application has a familiar interface with the grid as the background. There is a toolbar on top and an animation bar on the bottom. Instead of a linkage table there is now three tabs on the left.

Only one tab can be open at a time for each group (left and right). The left tabs can be closed, or opened to edit, analyze, or synthesize. The synthesis tab enables users to generate

a linkage based on the desired position of one of the links automatically. When the edit tab is open, users can modify the properties of the selected object (highlighted in amber color). The analysis panel, when open like in Figure 61, allows users to view kinematics and force analysis graphs for the selected object.

The right tab group can be closed or opened to display settings, the help panel, or the equations panel. The settings panel (open in Figure 61) enables users to modify overall mechanism settings, such as units, input speed, and enabling gravity, as well as visual settings like mechanism size and grid line visibility.

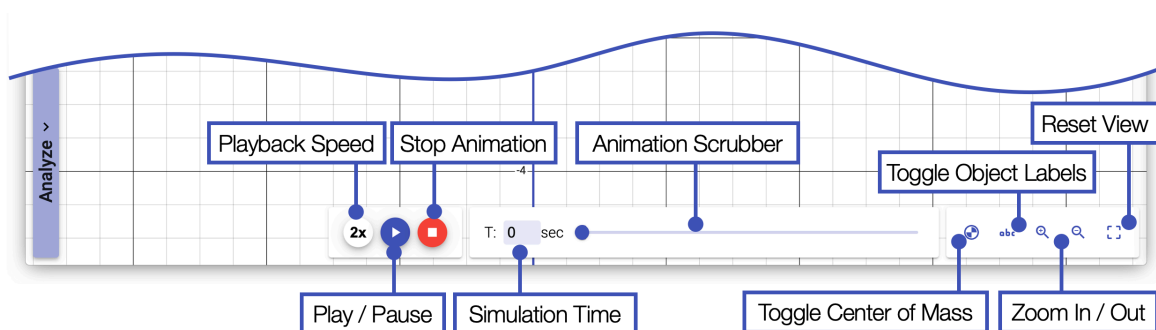


**Figure 61:** The three tabs on the left opens the left panel and the buttons on the right of the toolbar open the right panel. In this image, the analysis panel is open on the left and the settings panel is open on the right.

The animation bar (shown in Figure 62) features a series of buttons to control the playback of the linkage simulation. It is divided into three groups. The first group on the left contains the playback speed button, which defaults to a 2x 'medium' speed, completing a full rotation of the input link in just under three seconds. Clicking this button switches it to 3x, increasing the animation speed, and clicking it again switches it to 1x, the slowest setting. The indigo play button allows users to play and pause the animation, while the red stop button not only pauses the animation but also resets it to T=0, allowing the linkage to be modified since it cannot be

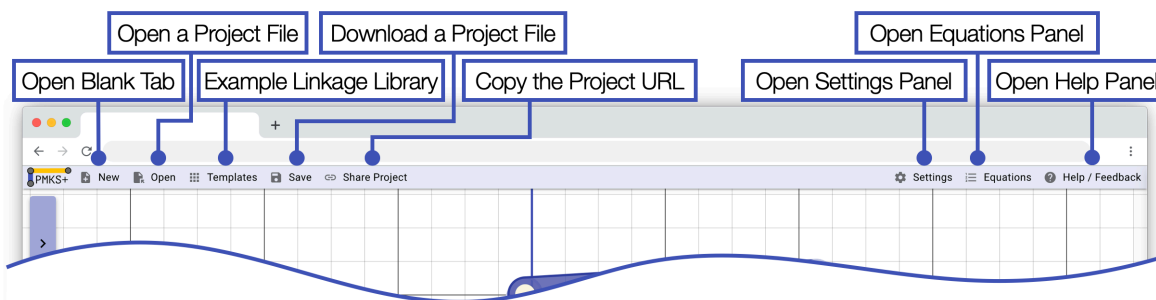
altered during simulation.

The second group of buttons displays and controls the simulation time. Instead of using time steps as in the original application, the redesign employs seconds as the time unit. The input box on the left displays the current simulation time in seconds, enabling users to enter a new value to jump to a specific point in the simulation. The scrubber to the right of the input box indicates the current time relative to the minimum ( $T=0$ ) and maximum time (when the mechanism completes a full revolution and returns to its starting position). Similar to a standard video slider, users can drag the thumb on the scrubber to seek through the simulation.



**Figure 62:** The New Animation bar contains 3 main groups of features: Playback Controls, Time Controls, and Grid Controls

The toolbar, located on the left side of the top bar and shown in Figure 63, features a PMKS+ logo, a button for opening a new blank file in another tab, a button for loading a previously downloaded PMKS+ file in the current tab, a button for accessing the Linkage Library containing numerous example linkages, a button for downloading a copy of the currently open project to the user's device, and a shape button for copying the URL, which encodes all project data into a single URL.



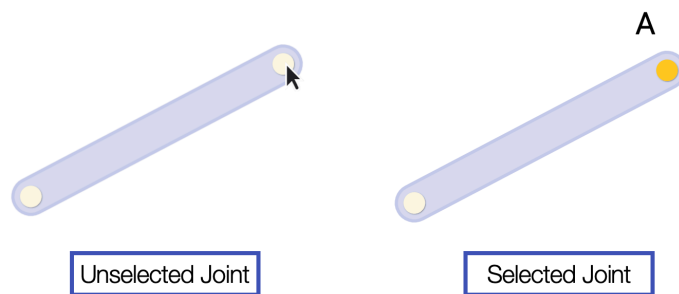
**Figure 63:** The New Toolbar has two sets of buttons. The left side contains file related actions and the right buttons open a right panel.

## 5.2 Solutions and Features

The application updates can be broadly categorized into seven solutions, addressing the pain points and fulfilling user needs:

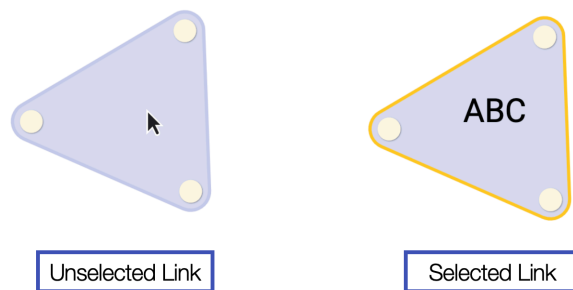
### 5.2.1 Click-to-select Objects

The redesigned application introduces the concept of selecting objects on the grid. Users can click on any object to select and highlight it. Pressing the escape key or clicking on the background deselects the object.



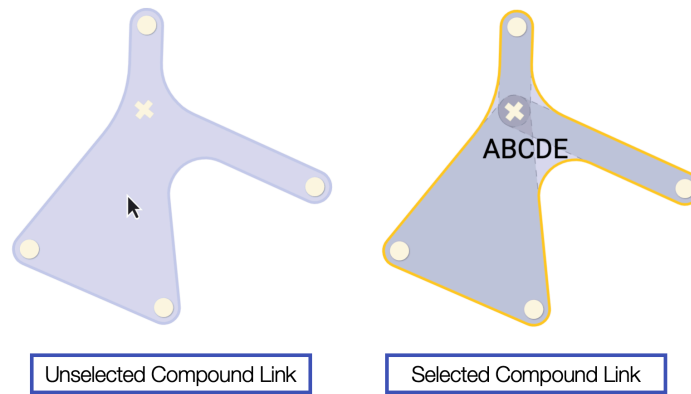
**Figure 64:** (Left) A binary link with both joints in the default state. (Right) A binary link with a joint selected.

As shown in Figure 64, selected joints turn amber, and their ID is displayed next to them.



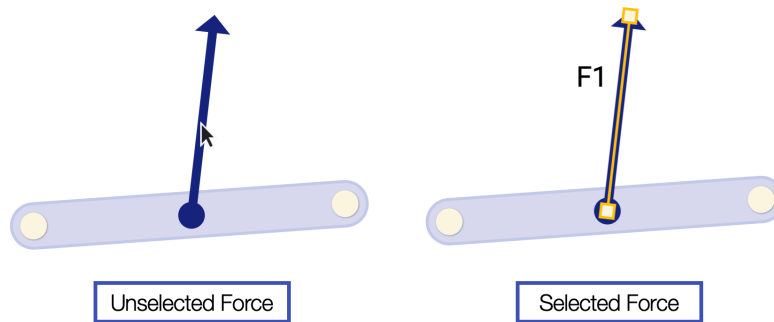
**Figure 65:** (Left) A 3 joint triangle link in its default state. (Right) When the same link is selected, there is a yellow outline.

As shown in Figure 65, when a link is selected, its outline changes from the original color to amber. The link's ID is shown at its center of mass.



**Figure 66:** (Left) A compound link in its default state. (Right) When the same link is selected, a yellow outline appears and the outline of the links that make up the compound link is shown.

The compound link (created by welding a joint with more than one link, discussed later in subsection 5.2.4) behaves similarly to a simple link but also displays the outline of the simple links that form the final shape. This is shown in Figure 66.



**Figure 67:** (Left) A force in its default (unselected) state. (Right) A selected force shows two draggable points at the start and end of the force.

Finally, as shown in Figure 67, a selected force displays two draggable handles at the start and end points, connected by an amber line.

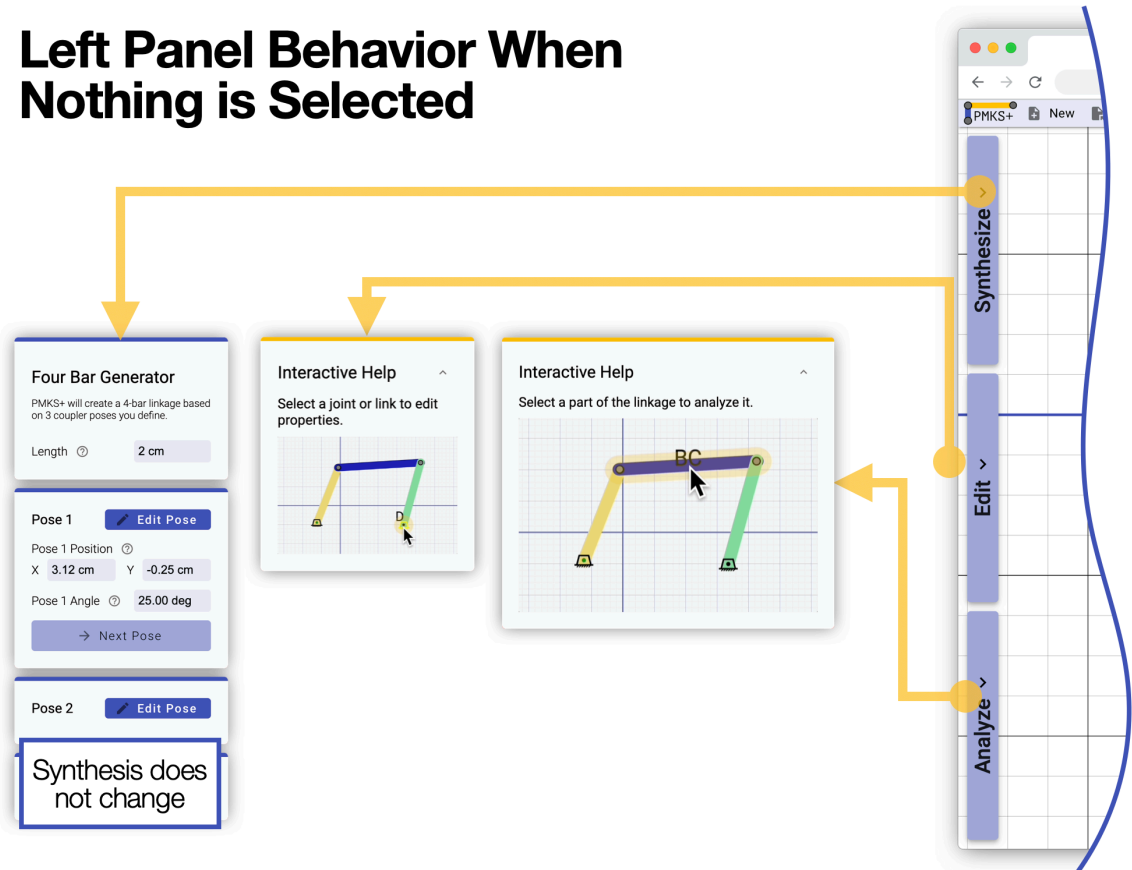
The ability to select an object allows users to establish their context and access options related to the selected object. This concept is further discussed in the following subsection:

### 5.2.2 3 Tab Navigation

The numerous benefits of switching to the 3 tabs on the right are explained in subsection 4.2.4, primarily making the three core features of the application (Synthesis, Editing, and Analysis) more discoverable.

Only one of the 3 tabs can be opened at a time, and they can be opened by clicking on them. Since PMKS+ contains a wealth of information and aims to maximize usable grid space, the tabs are content-aware and change based on the selected object. The synthesis tab allows users to automatically generate a linkage based on the desired position of one of the links. The edit tab enables users to edit the properties of the currently selected object. The analysis panel displays kinematic and force analysis graphs for the selected object.

## Left Panel Behavior When Nothing is Selected



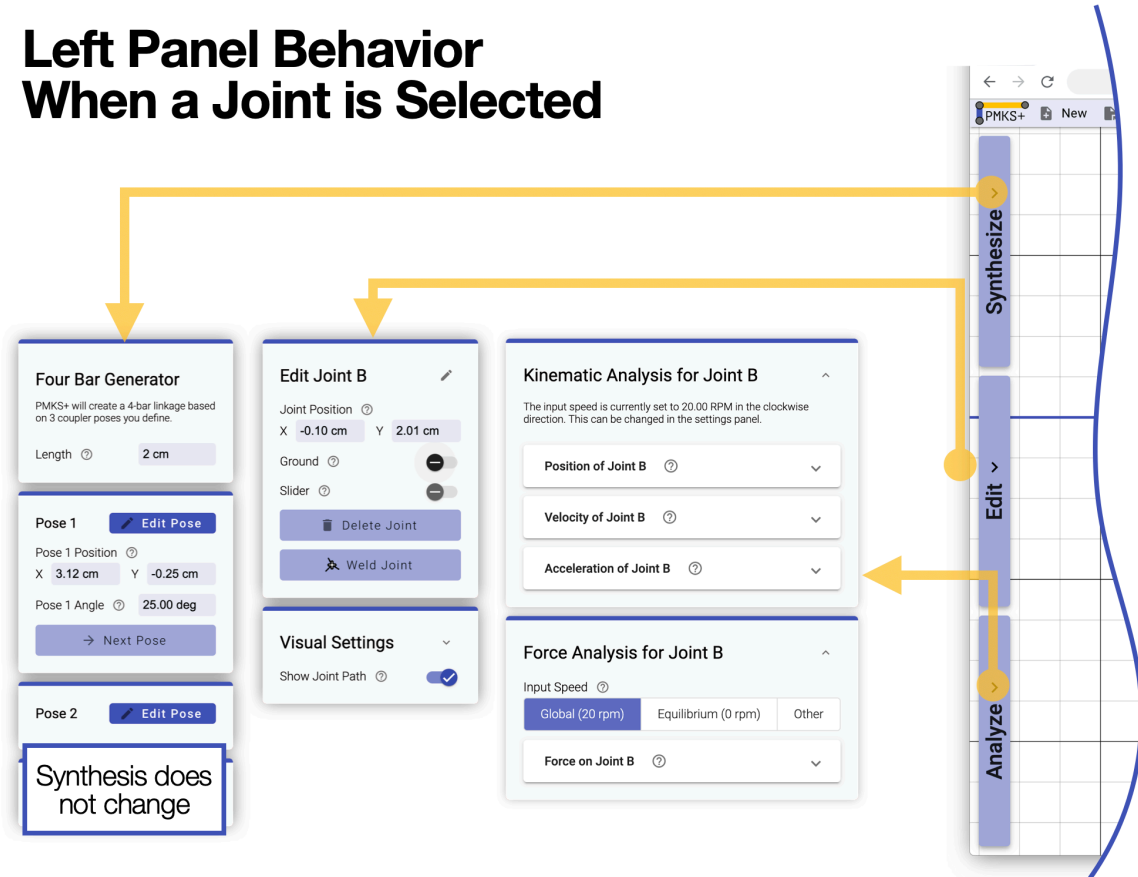
**Figure 68:** When no object is selected on the grid, the edit and analyze panel show an interactive help panel. The synthesis panel always shows the four-bar generator.

As shown in Figure 68, when nothing is selected (the user clicks on the background), the synthesis panel displays the panel for the synthesis process, explained further in Figure sub-subsection 4.2.7. Note that this panel does not change based on the selected object since it pertains to the entire mechanism as opposed to one part.

If the edit or analyze panels are open, they show an expandable help box prompting users to select an object to edit or analyze. When the explanation panel is opened by clicking on

the top right carrot, the panel expands, displaying an animation of the required action. Other types of "Interactive Help" pop-ups guide users with common problems, discussed further in Section subsection 5.2.8.

## Left Panel Behavior When a Joint is Selected



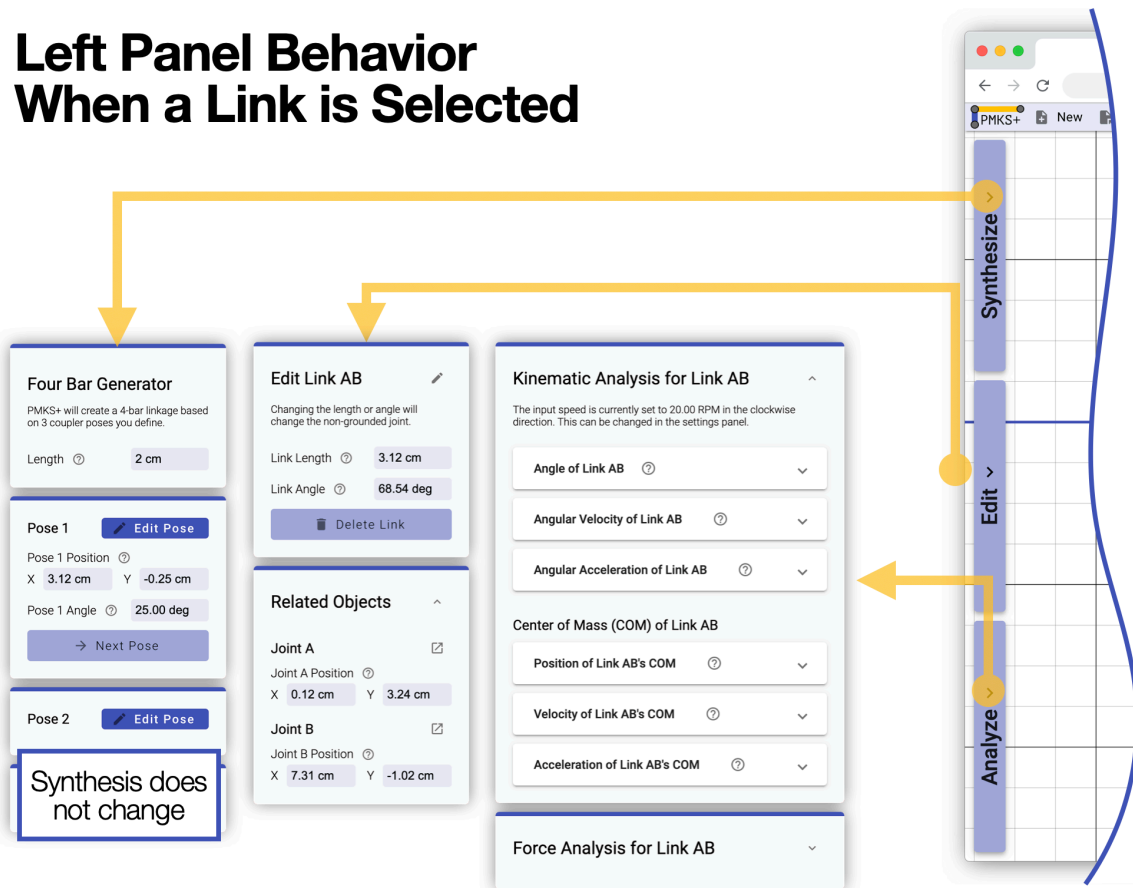
**Figure 69:** When a joint is the selected on the grid, the edit panel shows several options to edit the location, and whether it's grounded or an input. When the analysis tab is selected, graphs for the position, velocity, and acceleration of the joint are shown along with forces at the joint and an input torque.

As show in Figure 69, when a Joint or a tracer is selected and the edit tab is active, the 'Edit Joint' panel opens. The title informs users they are editing a joint and displays its name. Next to that is a pencil icon, allowing the user to edit the joint's name. The next item has two modifiable input boxes for viewing and editing the x and y locations of the joint. Then, a series of toggles enable users to set whether the joint is grounded, an input, or a slider. Buttons allow users to delete the joint or weld links connected to that joint together. Lastly, a new section contains visual settings for displaying the path the joint takes during the animation. Note that the visibility of these options changes depending on the joint's context (e.g., a non-grounded joint cannot be an input).



When the analysis tab is selected and the corresponding panel is open, two collapsible sections are shown: Kinematic Analysis and Force Analysis. The kinematic analysis options include joint position, velocity, and acceleration. Each graph heading can be expanded to show the corresponding graph, but they start collapsed. In the Force Analysis section, the type of analysis can be switched between dynamic analysis (solving for forces experienced at the joint while the linkage is in motion) and static analysis (solving for forces while the linkage is not moving) using a 3-option radio control. The force analysis options include the forces experienced by the joint or the input torque required for the input at that joint.

## Left Panel Behavior When a Link is Selected



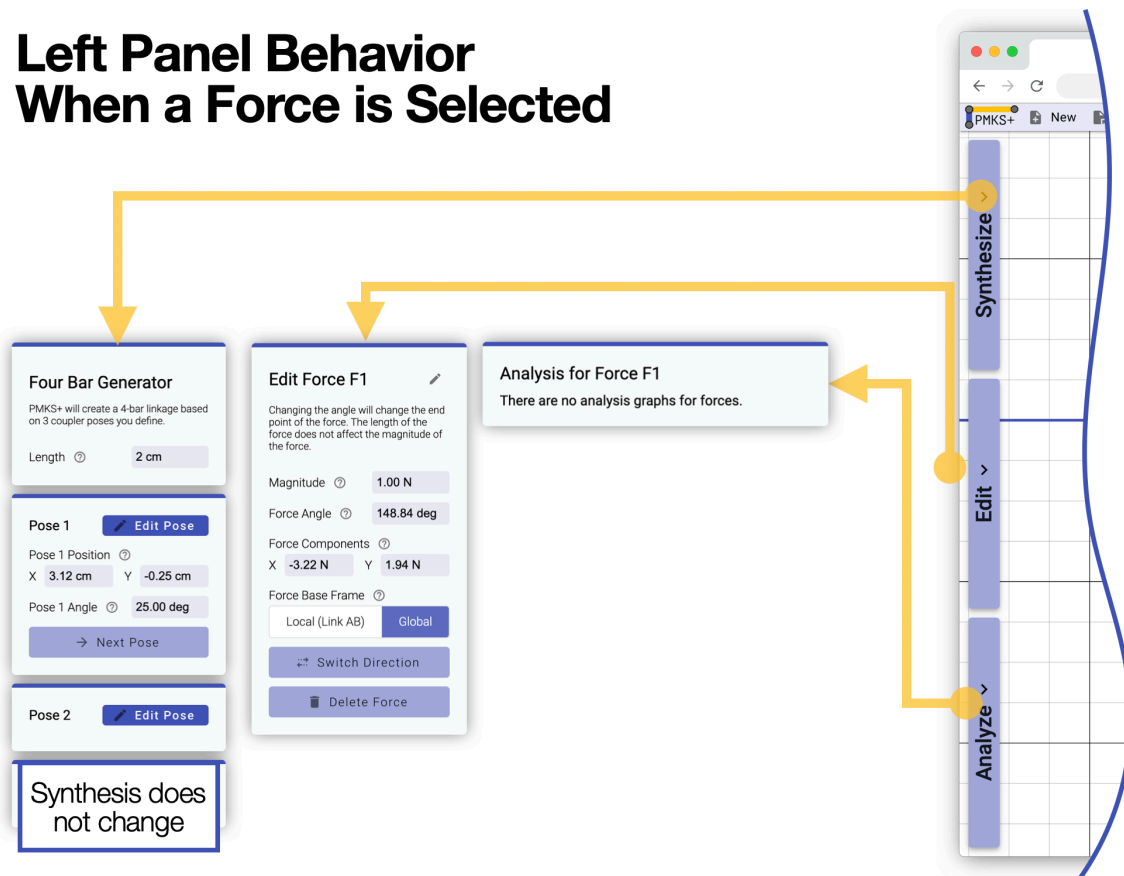
**Figure 70:** When a link is selected on the grid, the edit panel shows options to edit the length(s) and angle of the link along with a list of related joints and forces. In the analysis tab, there are kinematic properties of the link, the COM of the link, and force analysis graphs for links.

As shown in Figure 70, when a Link is selected and the Edit tab is activated, the 'Edit Link' panel is displayed. The title indicates to users that they are editing a link and lists the name of the link. Next to the title, a pencil icon allows the user to edit the name of the link. The

panel contains modifiable input boxes for the link length and link angle, enabling users to view and edit these properties. Additionally, a 'Related Objects' section lists all joints and forces associated with the link. Each related object features a small menu for adjusting the most common settings, such as the position of a joint, and a button that navigates directly to the panel of that particular object. This comprehensive Edit Link panel ensures users can easily modify link properties and navigate to relevant objects.

When the Analysis tab is selected with a link, the corresponding panel unveils three collapsible sections: Kinematics, Center of Mass (COM), and Forces. The Kinematics section presents the link's angle, angular velocity, and angular acceleration analyses. Each of these graph headings can be expanded to display the corresponding graph, but they initially appear collapsed. The COM section provides the position, velocity, and acceleration of the link's center of mass, allowing users to analyze the link's behavior in detail. The Forces section includes axial stress and axial forces analyses, which enable users to understand the forces acting on the link and the stress experienced as a result. Overall, the Link Analysis panel delivers comprehensive information on a link's kinematics, center of mass, and forces, equipping users with the necessary data for thorough and accurate linkage analysis.

## Left Panel Behavior When a Force is Selected

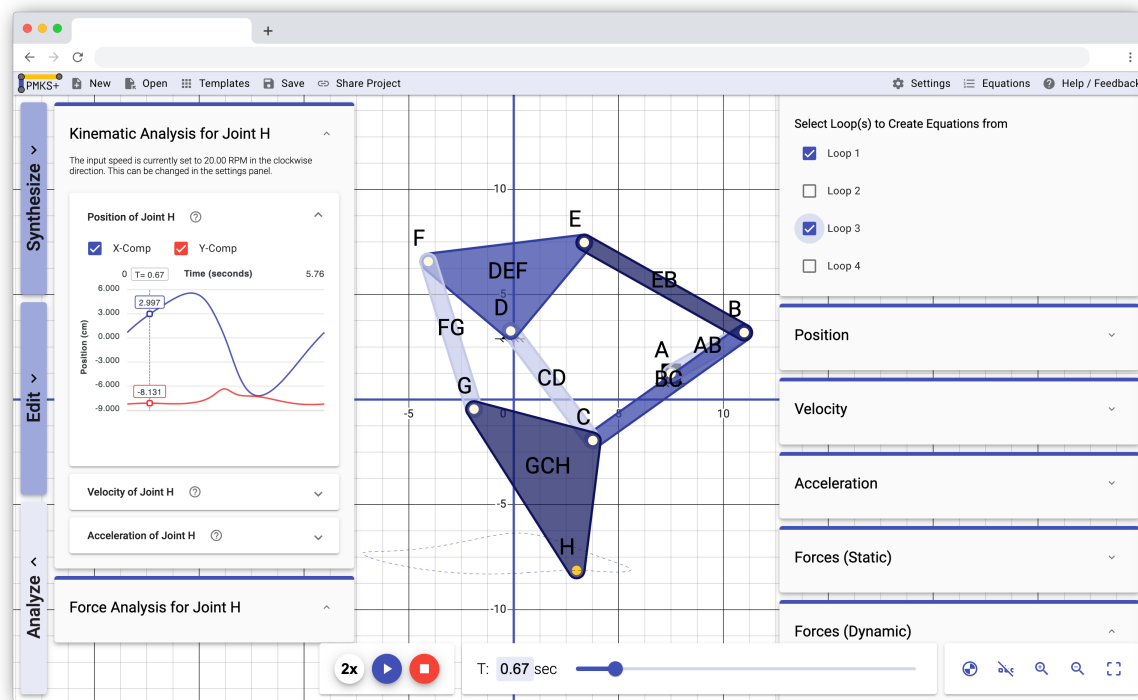


**Figure 71:** When a Force is selected, the edit panel shows options to edit the magnitude, angle, components, and reference frame. There are no analysis options for the force.

As shown in Figure 71, when a Force is selected and the Edit tab is activated, the 'Edit Force' panel is displayed. The title informs the users that they are editing a force, and it lists the name of the force. The panel contains modifiable input boxes for the force magnitude and angle, enabling users to view and edit these properties. Additionally, the X and Y components of the force can be adjusted to provide greater control over the force vector. A crucial feature in the Edit Force panel is the reference frame selection, which allows users to choose between Global or Local reference frames. This option determines whether the force maintains a consistent direction (like a weight hanging on the link) or rotates along with a link. A switch direction button is also available to quickly flip the force direction, while a delete button allows users to remove the force entirely.

When the Analysis tab is selected with a force, no force analysis options appear in the analyze panel. This is because forces are already being analyzed in the context of the link and joint analyses. Users can refer to the Link and Joint Analysis panels for information related

to forces, as they provide comprehensive details on the forces acting on the links and joints within the mechanism. By keeping the force analysis within the Link and Joint Analysis panels, users can better understand the system's behavior and the influence of forces in the overall mechanism performance.



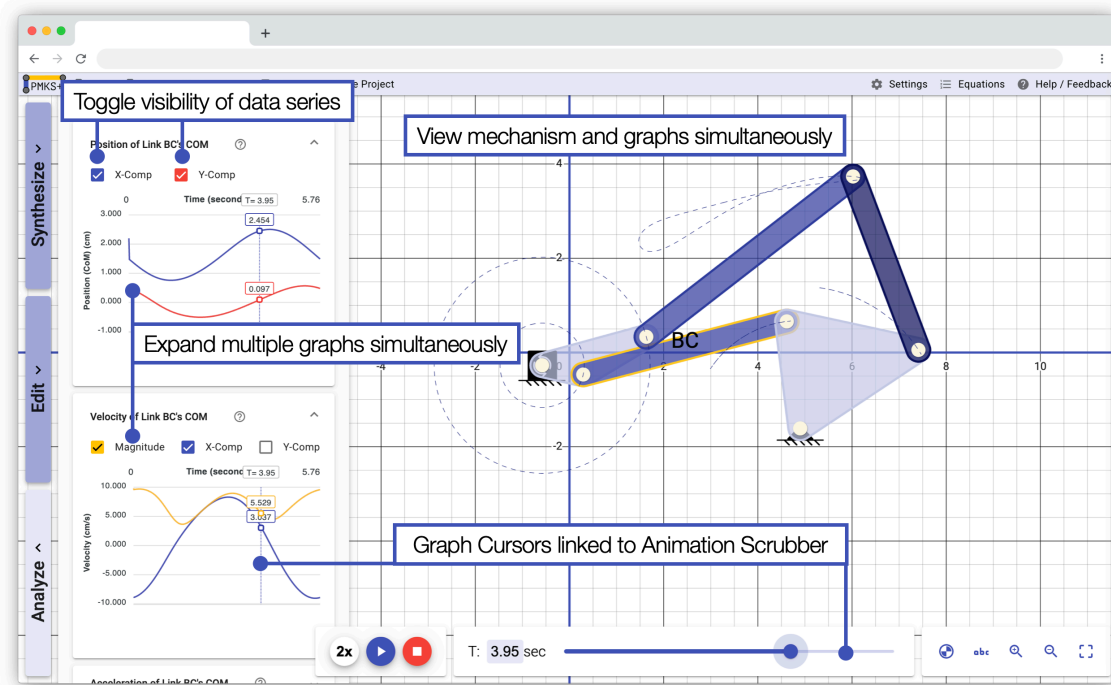
**Figure 72:** PMKS+ with the analyze and equations panel open at the same time. The analysis graphs, the linkage with labels, and the equations panel are all visible at simultaneously.

As pictured in Figure 72, moving away from a popup for analysis enables users to view the linkage, analysis graphs, and equations simultaneously, making the app feel less restrictive and encouraging experimentation. This allows the user not to have to remember the ID number to find the correct graph. Instead, they can just click on the object they are interested in to see properties. When the linkage is not animating, users can also drag joints and change link shapes and see the open analysis graph update in real time, allowing them to make minor modifications and iterate quickly to improve their design. Lastly, this design scales better with more objects than a table or list, since the information for only one object is shown at a time. It also allows room for expansion since more properties can be added to objects and still fit in the left panel.

### 5.2.3 Analysis Overhaul

As shown in Figure 73, the analysis graphs themselves have been greatly improved with this redesign. Now, data that have multiple components are grouped into a single graph with each series being manually toggleable. Multiple graphs can be expanded at the same time to compare the relationships between kinematic or force properties over time.

The design also leans into the visibility of the graphs and mechanism simultaneously and has a scrolling bar that follows the visible position of the mechanism and the animation scrubber. This means a user can simulate the linkage to a specific point and see the kinematic or force values of the linkage at that point in the simulation.

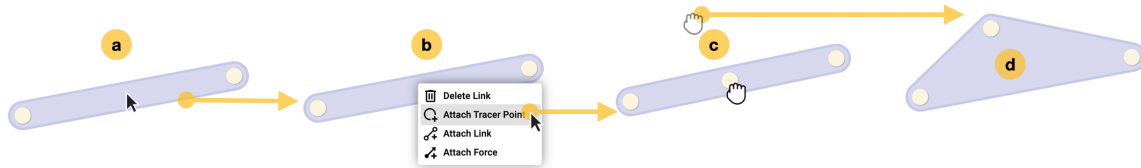


**Figure 73:** The new analysis panel allow multiple graphs to be open at the same time. There is a scrolling cursor on the graph that is linked to the simulation time. The mechanism is visible alongside the graphs allowing for better association.

This analysis overhaul enhances the user experience by providing clearer visualizations, streamlined navigation, and a more intuitive interface for understanding the interplay between kinematic and force properties within the mechanism. By allowing users to view and compare multiple velocity graphs at once, the new design empowers them to gain deeper insights into their linkage's performance, leading to more informed decisions and ultimately better designs.

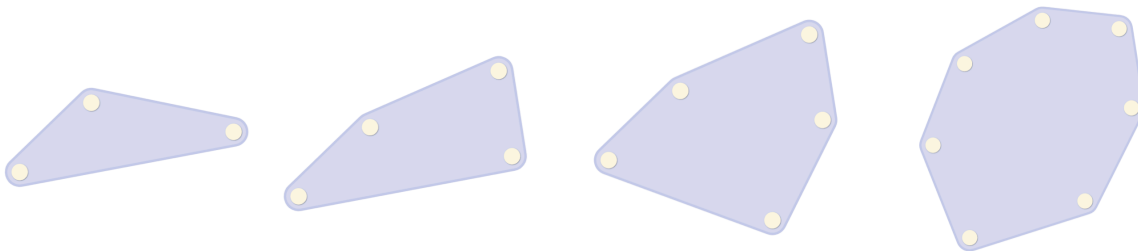
## 5.2.4 Infinitely Customizable shapes

Another major component that was overhauled is the shapes engine. The new system makes this system more flexible in terms of the types of shapes that are possible while reducing the interaction complexity.



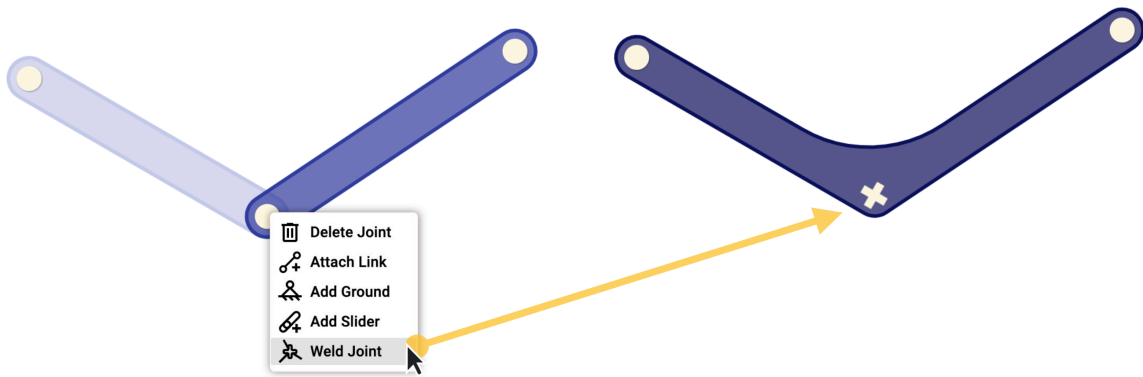
**Figure 74:** A series of images showing the creation of a triangular link from a binary link. From left to right: (a) Right click on the link to bring up the context menu. (b) Click *Add Tracer Point*. (c,d) Drag the tracer point up to create the shape defined by the convex hull of the 3 points.

The first part allows tracer points to define the shape of a link. As shown in Figure 74, this allows the user to add a tracer to a link and drag the joints to create a shape that includes all the external points.



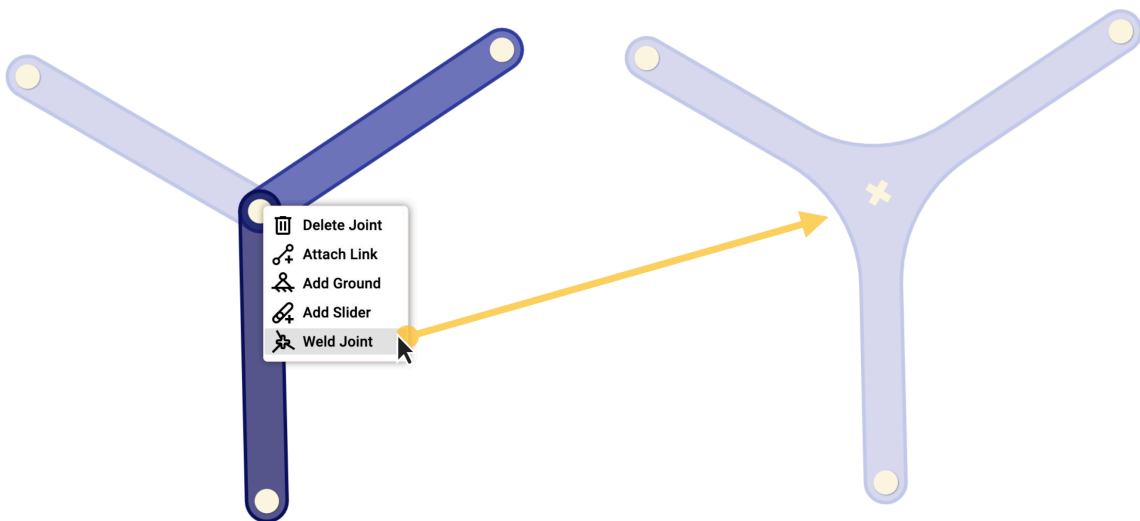
**Figure 75:** Other types of shapes that can be created by adding more tracer points. From left to right, there are 3, 4, 5, and 6 joints that make each link.

As shown in Figure 75, this simple system allows many convex polygons to be created with no limit to the number of joints for each link. This, however, does not allow for concave shapes like a T, C, or V.



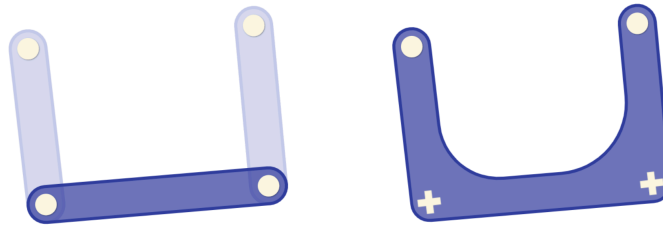
**Figure 76:** Two binary links can be welded together by clicking *Weld Joint* on the joint shared by them.

Welding at a Joint combines all links attached to that joint into one rigid link. The shape of the joint changes from a circular peg to a + shaped peg, indicating that links cannot rotate along that joint anymore. Internal corners are filleted to further emphasize this idea visually, indicating that these two links move as one. This process is shown in Figure 76 and Figure 77.

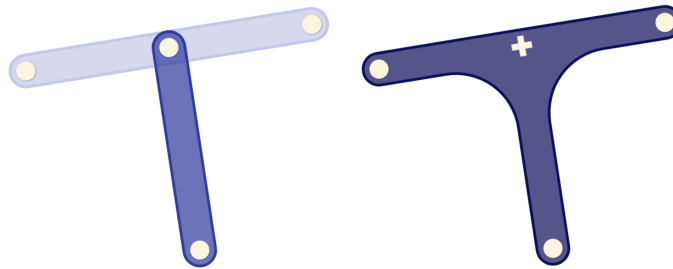


**Figure 77:** Three binary links can be welded together by clicking *Weld Joint* on the joint shared by them.

As shown in Figure 78 and Figure 79. This system allows all shapes that were previously possible in PMKS+ to be created and with more flexibility, as users can define the proportions of each side.

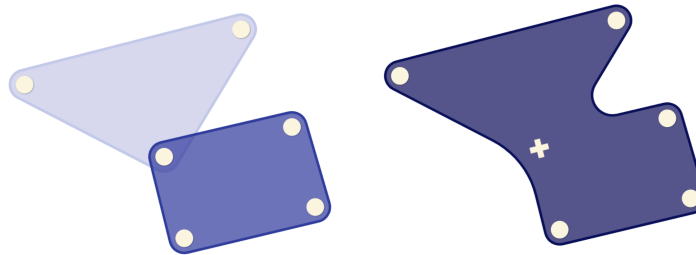


**Figure 78:** A C shape can be created by welding together 3 binary links.

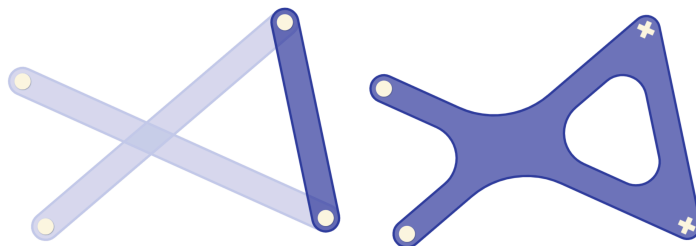


**Figure 79:** A T shape can be created by welding together a binary link with a linear 3 joint link.

Combining both these features allows users to weld together not just binary links but any convex polygon, enabling users to create any non-curved shape imaginable. An example of combining a triangle and a square is shown in Figure 80.



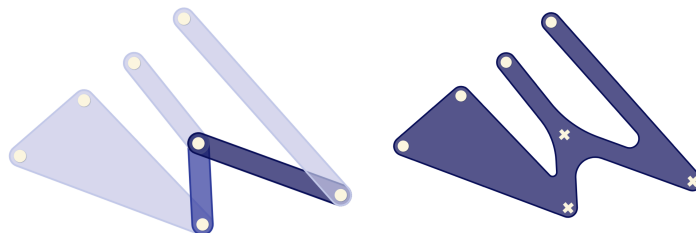
**Figure 80:** Multiple polygons can be welded together to create a compound link.



**Figure 81:** Links that are intersecting can be combined to create complex welded shapes.



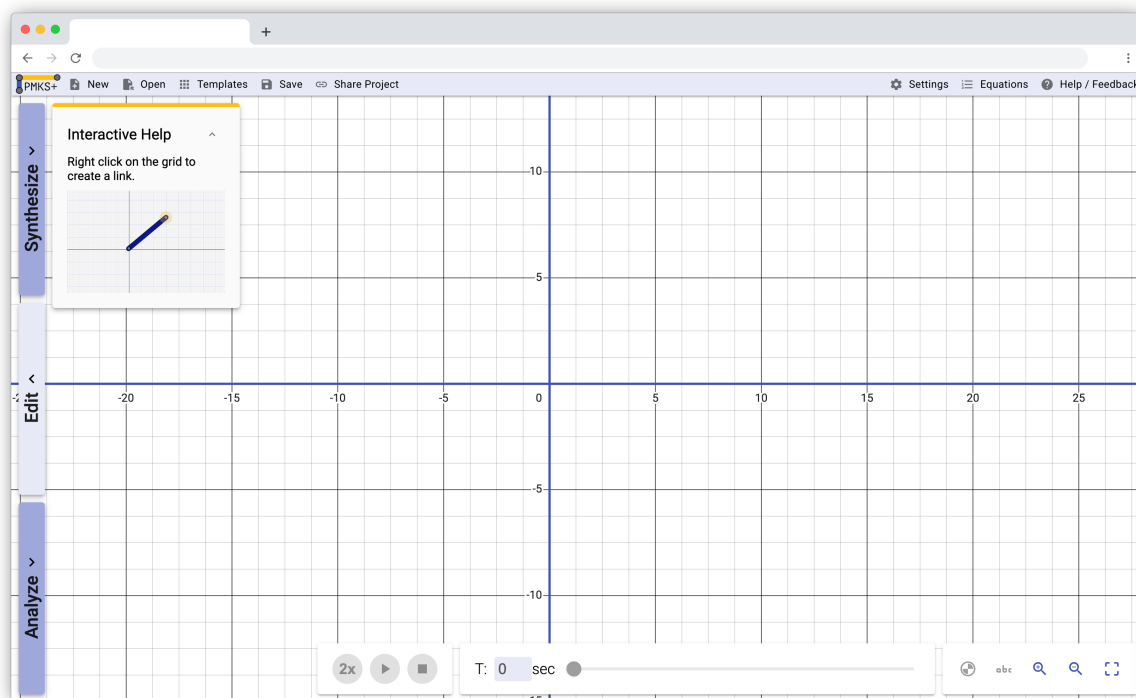
Users can also make shapes that have holes in and an unlimited amount of simple links can be welded together. Examples of this can be seen in Figure 81 and Figure 82. This vast range of customizable shapes significantly expands the possibilities for mechanism design and simulation within PMKS+.



**Figure 82:** Any number of links can be welded together to create compound shapes.

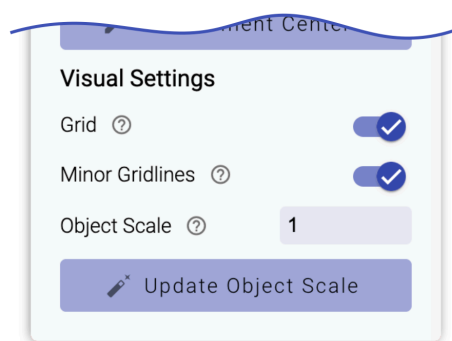
### 5.2.5 Dynamic Grid

The Grid shown behind the linkage has also been greatly upgraded compared to the original application. First, the grid has a much larger zoom range, allowing users to see links as small as 0.5 units all the way to 10,000 units. The limit is completely arbitrary, and there is no reason the zoom range can't be increased if there is a need. This is now possible because the grid lines are rendered dynamically. As users zoom in, only the lines visible in the view are calculated and rendered; also, as they zoom in further, the grid gets subdivided further the closer you zoom in. Similarly, when the user zooms out, the smaller lines dynamically disappear. This ensures that the grid lines are useful for checking lengths of links at a glance. To further aid with this, labels on the x and y-axis have been added at major grid lines, and they appear at easily countable intervals (0.5, 1, 2, 5, 10, 20...). The redesigned grid is show in Figure 83.



**Figure 83:** The new grid has dynamically changing labels along with grid lines the appear to further subdivide the grid when the user zooms in.

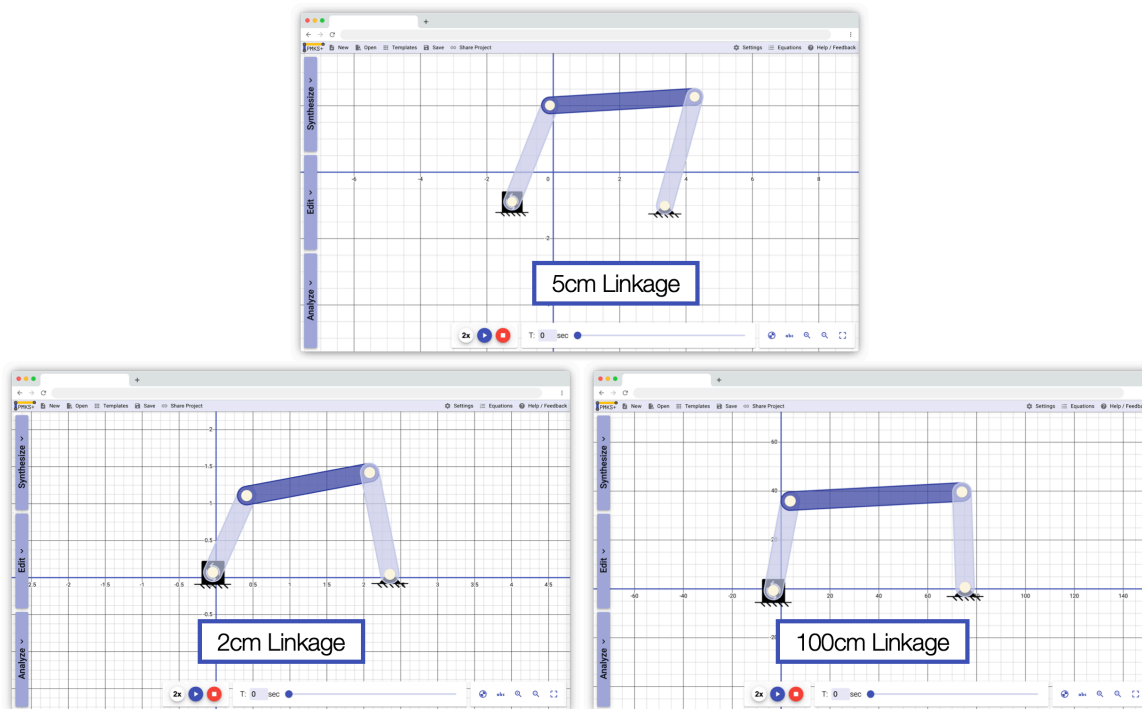
Also, the new grid adds animation options, allowing the recenter option to zoom and pan to the linkage smoothly, which helps the user mentally keep track of where they are in the grid space.



**Figure 84:** The bottom of the *settings panel* has options to toggle the visibility of the grid, the minor grid lines, and the size of the objects on the grid.

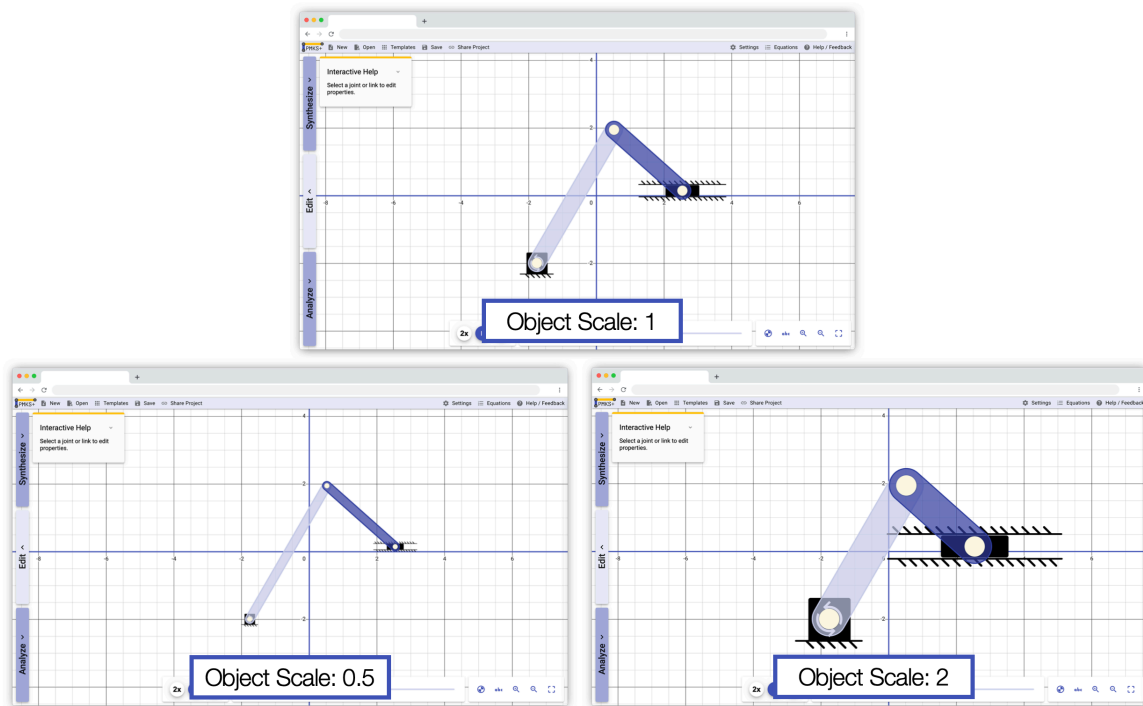
The grid is also now more customizable, allowing the grid to be toggled, as well as minor gridlines that can be hidden to reduce clutter on the screen if needed, all from the settings

panel in Figure 84. Lastly, there is an object scale setting that allows users to manually scale down the visual size of the mechanism, as well as an option to update it automatically based on the zoom level and the size of the linkage.



**Figure 85:** A collection of four-bars created in the new PMKS+ with different sizes. (Left) A four bar with a coupler about 2cm. (Center) A similar four-bar where the coupler is about 5cm. (Right) The coupler is about 100cm.

These updates in aggregate allow users the option to build linkages of any size they like and any complexity. As shown in Figure 85, users can have a similar experience building a 2cm mechanism or something that is 50 times larger. As shown in Figure 86, they can also modify the scale for a simple linkage to make it easier to show to someone on a small screen or reduce the scale for a complicated linkage with many overlapping links.



**Figure 86:** A collection of the same slider crank with different object scales. (Left) Object scale is 0.5. (Center) Object scale is 1, which is the default. (Right) Object scale is 2.

## 5.2.6 Synthesis

The auto-generation of linkages removes one of the most time-consuming parts of a typical user's journey through the app. This feature encourages students to use the application outside of specific required homework assignments by eliminating the largest roadblock and enables the teaching of synthesis methods in courses. This feature has not been completed at the completion of this report. See section subsection 6.1 for more details of upcoming features.

## 5.2.7 Responsiveness and Material Design

Material Design is a design system that was developed by Google in 2014. It is an adaptable system of guidelines, components, and tools that support the best practices of user interface design. It streamlines collaboration between designers and developers, and helps teams quickly build beautiful products.

Material Design is based on the principles of physical reality, such as light, shadow, motion, and material properties. It uses a consistent visual language across different platforms

and devices, such as web, mobile, and desktop. It also provides a set of components, icons, typography, color palettes, and sound effects that can be customized to fit different brands and products.

Material Design is not just a style or a theme, but a way of thinking about design and user experience. It aims to create intuitive, engaging, and delightful interfaces that are responsive, accessible, and inclusive.

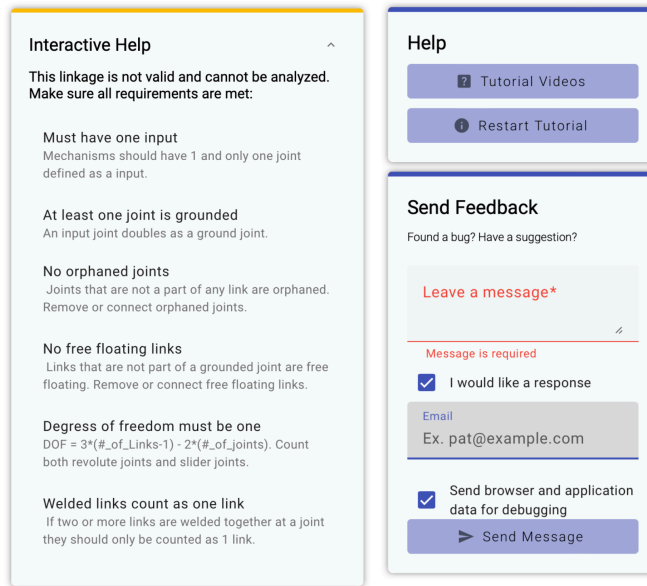
Many components in the application use Angular Material, a Material Design Library for Angular. The pre-made components combined with the use of the Component Development Kit (CDK) for most other custom components ensure that a lot of the app follows Google's modern Material Design guidelines.

In addition, many very small user experience features are consistent with what Material Design suggests. Here are a couple of examples of this in PMKS+:

- Panels on the left and right slide out of the screen instead of fading out
- Draggable elements like joints lift off the screen when they are hovered over and lift off the screen even further when they are being dragged
- Buttons and other inputs change color slightly when hovered over to signify that they are clickable
- Panels have a defined z-height off the grid and never pass through each other
- Expansion panels animate in as the content expands out of the header

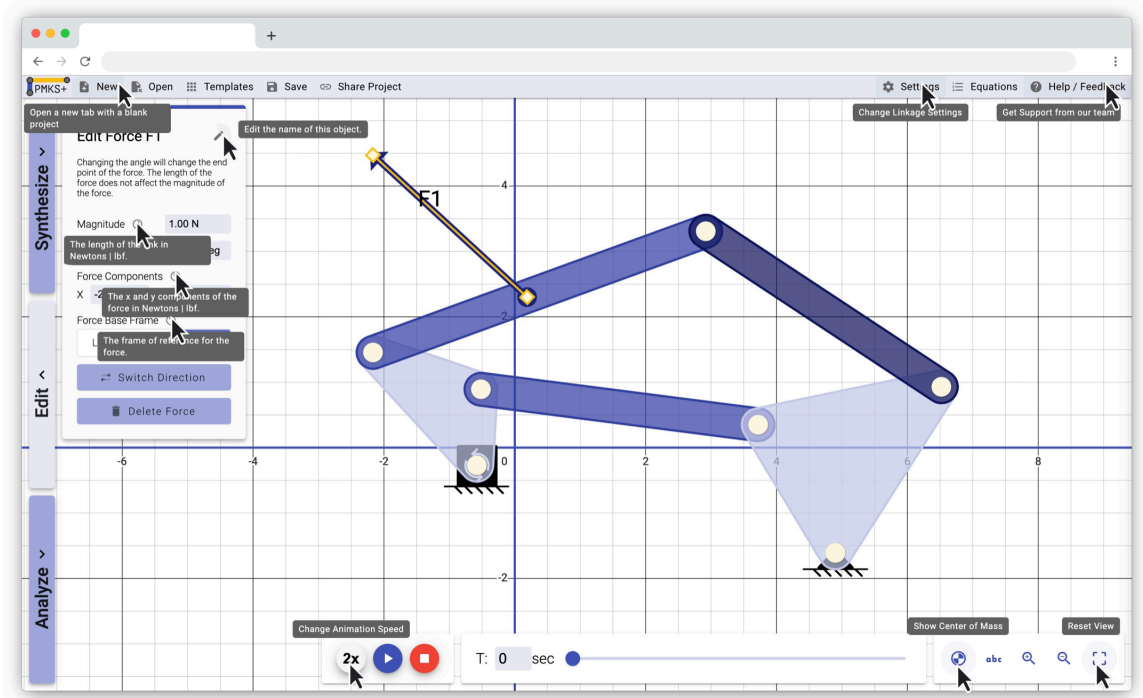
### **5.2.8 Other Quality of Life Improvements**

Several Quality of Life Improvements make the application more delightful to use. Here is a selection of some of the most impactful ones.



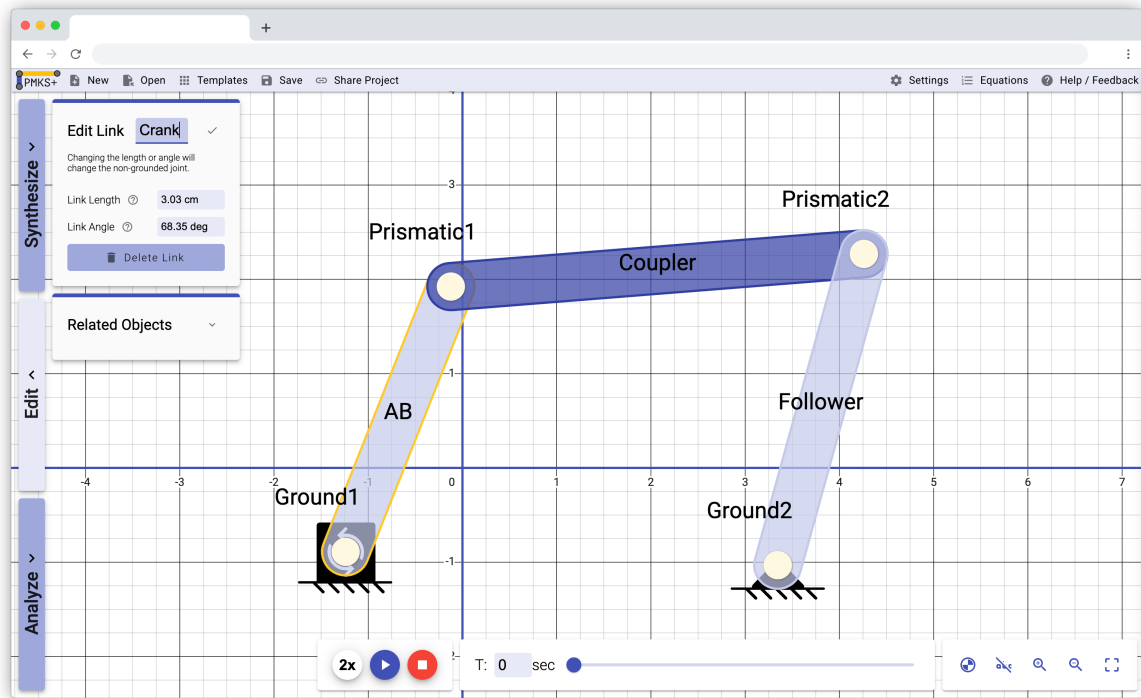
**Figure 87:** (Left) The help panel shown in the analysis tab when the linkage is not valid. (Right) The help panel on the right. There is a are buttons to access tutorials and form to send feedback to the team.

Multiple help options were added, including a guide to help resolve issues with the linkage when it cannot be simulated. Also, a feedback form to get feedback directly from the developers has been added into the help panel. Both of these panels are pictured in Figure 87.



**Figure 88:** An image of the new PMKS showing many possible help tooltips that are shown when the mouse is hovered over a button for a second.

As shown in Figure 88, tooltips have been added to virtually every button that appears when a user leaves a mouse over a button for a second. This helps improve accessibility and helps remove confusion by providing contextual information when needed.



**Figure 89:** An example of a four-bar linkage where all links and joints are given a custom name. Link AB is in the process of being renamed to Crank.

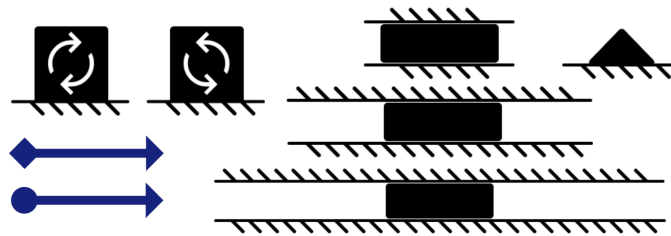
As shown in Figure 89, the option to rename joints, links, and forces have been added, allowing linkage objects to be named more specifically and helping users keep track of important objects more easily.



**Figure 90:** A grid of some of the icons designed specifically for PMKS+ to give a better visual indication of what a button will do.

Iconography has been added to every reasonable button and label. A brand new set of icons (shown in Figure 90) were designed to allow menus to be more descriptive and help users perform tasks quicker by remembering icons instead of reading text.





**Figure 91:** The new graphics uses to represents grounded and input joints, prismatic joints, and forces.

As shown in Figure 91, inputs, grounds, sliders, and forces have been redesigned visually to convey more information at a glance about the expected behavior, balancing visual attractiveness with maintaining similarity with conventional icons for these objects in textbooks and other applications.

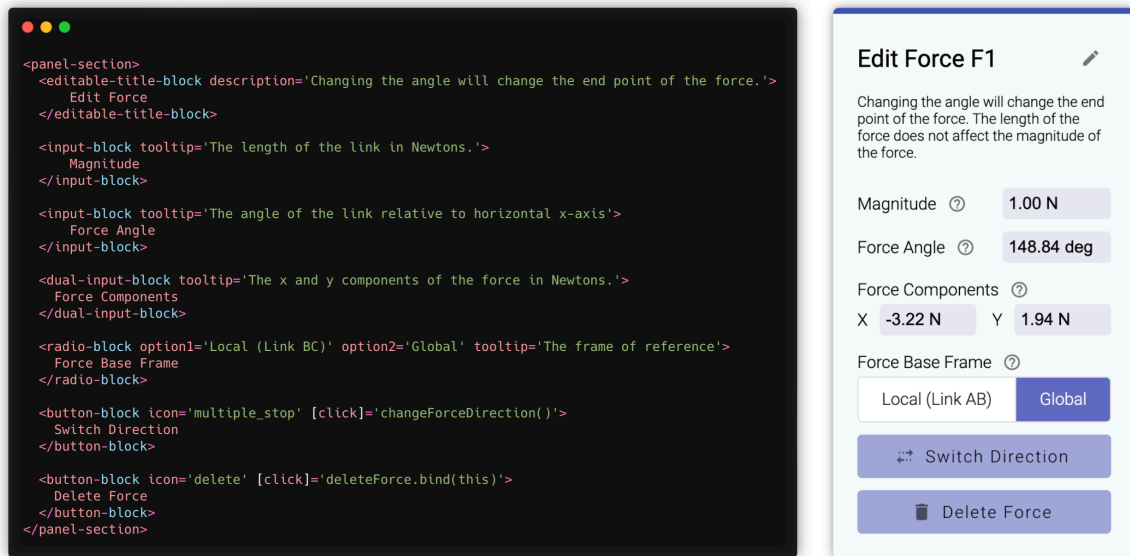
### 5.3 Backend and Software Engineering Optimizations

This report focuses on aspects that impact the user experience, so it will avoid discussing the backend in depth. Instead, it will primarily concentrate on backend improvements that have a direct effect on users or developers who work on PMKS+ in the future.

1. **Improved Content Delivery Network:** In previous versions of PMKS+, users encountered issues with updates not being delivered promptly. As a result, old versions of the app would be served from the cache, causing students to experience different behaviors. By integrating with Netlify and their Content Delivery Network and Continuous Integration / Deployment pipeline, a new commit to the production branch triggers a rebuild, along with an invalidation of worldwide caches, resolving this issue entirely.
2. **No more loading screen:** Loading times had been a problem with previous versions of PMKS+, necessitating the addition of a spinner. A combination of splitting components into smaller classes (which are only loaded when visible) and an improved content delivery network that brings the content closer to end-users has significantly enhanced load times. There is no longer a loading screen, and users are immediately brought to the main interface.
3. **Decomposition:** Since the first migration in 2019, the Object-Oriented Design Structure has not been updated. Instead, developers have continued to add more features to the handful of components initially created, leading to files that exceeded 2500 lines of code and over 100 static variables. Although this transition will be an ongoing task, significant

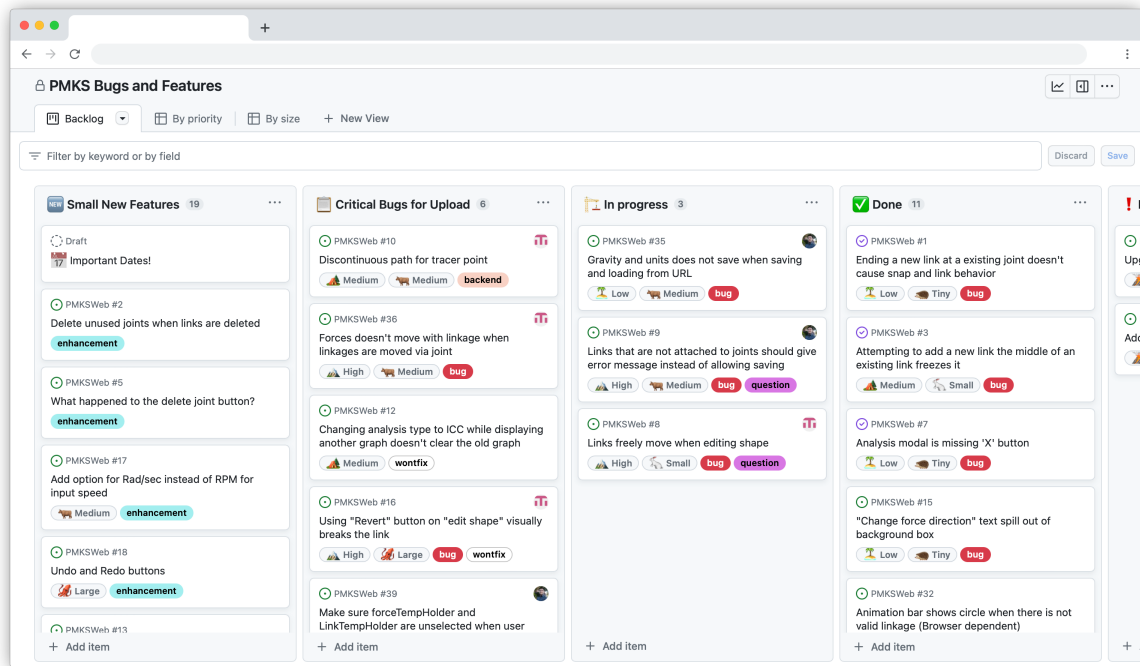
strides have been made in reducing the length and complexity of each component by separating the backend and simulation logic from the user interface code. This was achieved by breaking up large components into smaller, reusable components, allowing developers to understand only the code necessary for the feature they are working on.

4. **Dependency Injection:** Angular enables simple dependency injection by adding an `@Injectable` decorator to classes. By creating multiple service classes like `MechanismService` or `NumberUnitService`, UI components can focus on the logic required to enable the interface, such as the state of a button, instead of embedding backend and data model code in components. For instance, instead of the grid or the animation bar keeping track of the simulation timestep, an injected `Mechanism` service acts as the single source of truth for tracking all information related to the mechanism state. Similarly, instead of each panel containing logic to handle natural language processing, where users can type a unit into a text field and convert it to the correct unit, the textbox simply calls the number unit parsing service, which parses the string into a number and an enum unit. This approach adheres to the single responsibility principle, ensuring that each component is only responsible for one very specific task. Following this principle allows for easier unit testing and prevents unexpected side effects from future changes.
5. **Outsourcing to Libraries:** Wherever possible, complex logic has been outsourced to open-source Angular libraries. For example, when computing the shape of a link based on joints, a convex hull library is used to find the joints that impact the shape of the link and the order to follow to draw the shape. Similarly, a library takes the results of an analysis and plots them. Only minor settings for this library are changed to adapt it to the application. Lastly, Angular Material and its component development kit are used to implement basic material design interactions and simple buttons and inputs. This approach has multiple benefits, including reduced testing and development needed by our team, and increased reliability and security.
6. **Interface Driven Components:** Capitalizing on the declarative nature of HTML, a set of reusable components has been developed to allow future developers to quickly and consistently create interfaces that share the same interaction patterns as the current interface. A prime example of this shown in Figure 92 where the *Edit Force* panel is built based on these blocks.



**Figure 92:** A simplified code snippet illustrating the declarative input block components

- 7. Documentation:** High-quality and accessible documentation is crucial for attracting developers and improving their efficiency. Functional programming has been implemented in more parts of the code, making the code itself more descriptive. The number of comments has been increased, and a high-quality README has been added to help with this. A kanban board (pictured in Figure 93) has been created to keep track of known bugs and issues, improving communication and avoiding duplicate work. Like with some of the other changes, this requires a consistent effort from future developers to maintain and improve.

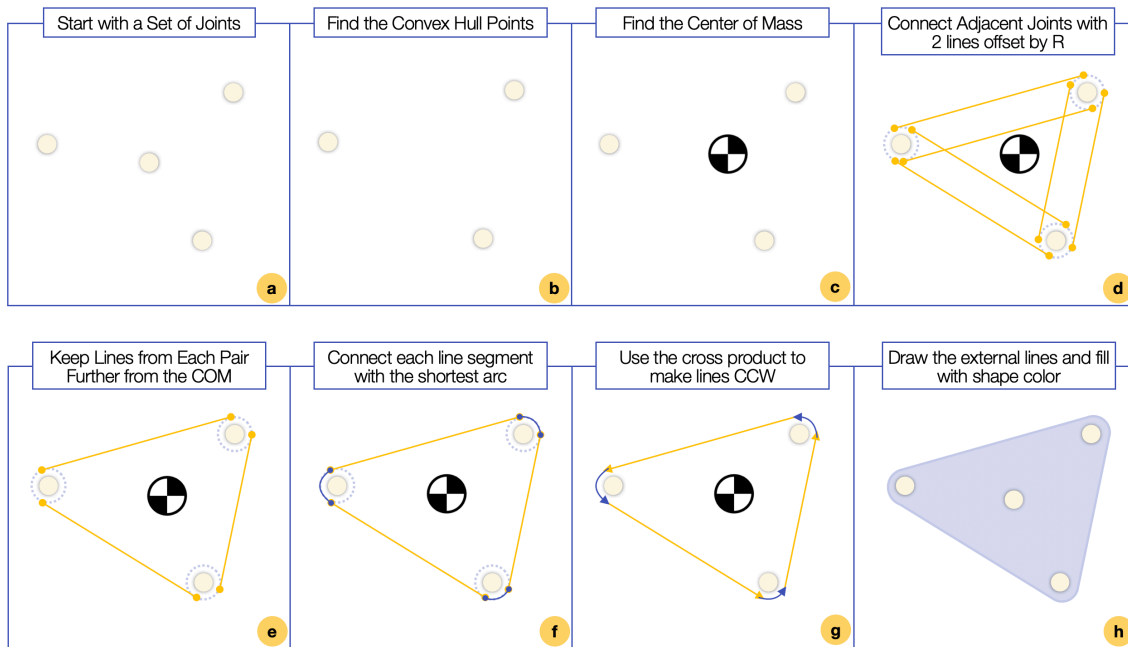


**Figure 93:** An Issue Tracker board showing a list of known issues and tasks assigned to team members

### 5.3.1 Interesting Algorithms

A couple of noteworthy algorithms were developed, which are described here to help future developers better understand them in case modifications are necessary.

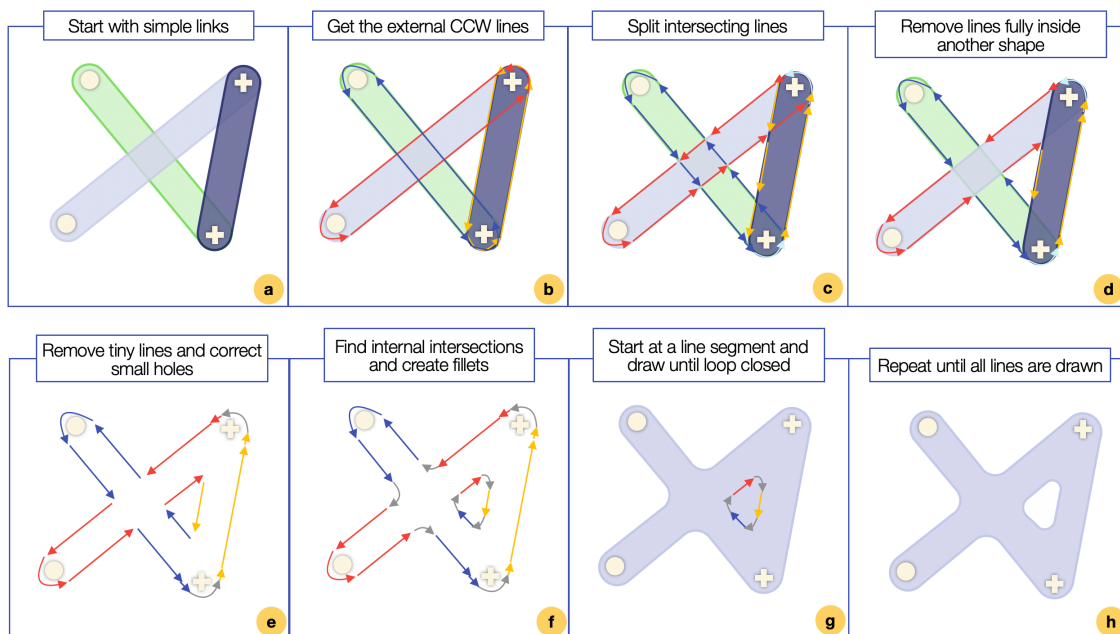
The first algorithm (shown in Figure 94) computes the SVG path needed to draw a simple shape. A simple shape is a convex shape made from any number of joints, such as binary links (bars), triangles, squares, and pentagons.



**Figure 94:** A series of images showing the backend process for computing the SVG path for a simple shape. This process occurs whenever a position of any of the link's joint's position changes.

(a) Initially, the algorithm receives a series of points that will be part of the shape. (b) This series of joints is passed to an open-source library called Hull.js [13], which filters out the internal joints that do not impact the final outer shape. (c) Next, the center of the shape is computed by taking the average x and y values of all joints. (d) Then, a circle is created around each joint, and adjacent joints are connected with two lines that are tangent to the circle created around each line. (Note that the radius R of these circles determines the size of the thickness of the link.) (e) For each set of lines, the ones closest to the COM are removed, leaving only the external lines for each set of joints. (f) Then, the gaps between these external lines are connected by an arc with the same radius as the circles. (g) Lastly, the cross product is used to ensure that the entire loop travels using the right-hand rule (CCW) around the center of mass. (h) This is then converted into an SVG path string and drawn until the path is closed and is filled with the shape's color.

A completely different algorithm (shown in Figure 95) is used to compute compound shape paths.

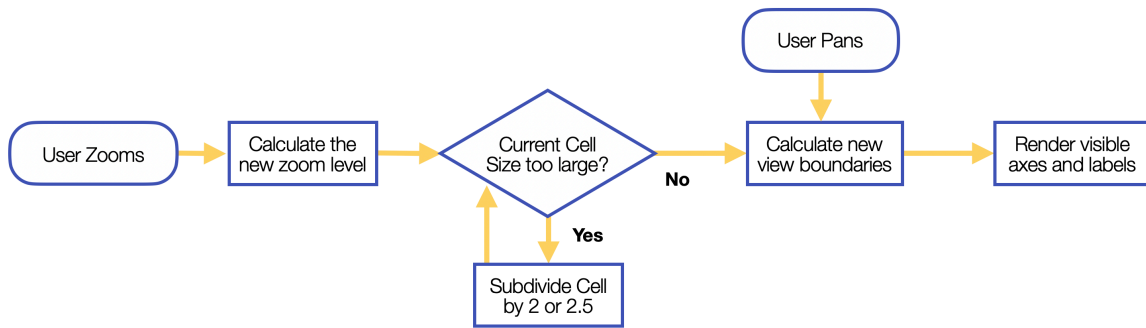


**Figure 95:** A series of images showing the process of computing the SVG path for a compound link. Similar to simple shapes, this computation is done every time any of the joint positions change.

(a,b) First, the algorithm is given a series of simple links, all of which have a series of joints and the external lines already computed. (c) All lines are compared against every other line to find intersections. At each intersection, each line is broken into two at the intersection. (d) Next, any line that is entirely inside another simple link is removed, leaving only the external lines for the compound shape. (e) Now, looking at the intersections, any intersections between two lines (not arcs) are filleted. This is done by shortening the two lines at the intersection and adding an arc that connects the two lines smoothly. The final step of the algorithm is drawing. By starting at one line and finding the line that starts closest to where this one ends, loops can be created that the SVG can follow. (g) If there are lines that remain after the loop is closed, the *evenodd* fill pattern can be used to draw multiple loops, allowing holes to be cut out of the shape.

The new grid has been overhauled to offload most of the zoom and pan behavior to an external library. *Svg-Pan-Zoom* is an open-source library that handles zooming and panning of the SVG content [14]. Compared to a previous team's implementation that implemented the interactions and transformations from scratch, using this library allows for increased robustness with features like animations and touchscreen support.

However, the custom grid spacing and label had to be implemented manually to fit the needs of PMKS+. A high level overview of this is show in Figure 96.



**Figure 96:** A flowchart showing a high-level view of how the dynamic grid lines and labels get updated.

Whenever a user zooms in on the grid, the new zoom level is calculated using the size of the grid on the browser and the zoom level. This allows the grid to calculate the default size of a cell (distance between grid lines). If this cell size is too large, it is subdivided by a sequence of 2, 2, and 2.5. (This means that the cell will decrease in size in this order: 100, 50, 25, 10, 5, 2.5, 1). This is repeated until the cell size reaches a certain threshold where the size is useful for users to use as a ruler on the screen. Next, the new minimum and maximum values of the viewport are calculated, and the visible lines are rendered based on the computed cell size as well as other user-defined grid settings.

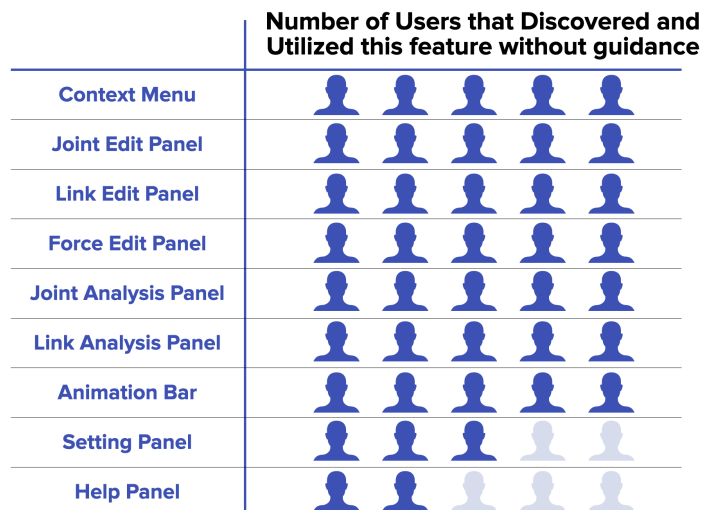
## 5.4 Validation Testing

### 5.4.1 User Testing

The final test was conducted on April 11th, 2023 with a 5-user focus group in person. The test began with a 5-minute demographic survey, followed by a 5-minute task description without showing the participants PMKS+. The participants then spent 20 minutes performing the task, during which time each researcher asked three people to do a cognitive walkthrough with a think-aloud protocol. The session concluded with a 20-minute open discussion, during which the researchers asked the whole group open-ended questions.

One of the key hypotheses for this validation test was that the new design would increase the organic discovery of analysis features. Although none of the users had previously used this redesigned version of PMKS+ and most of them did not have any experience with any version of PMKS, all five participants discovered and utilized the following features during their 20-minute test of the app: opening the right-click context menu to edit the mechanism, editing joint properties with the edit panel, editing link properties with the edit panel, editing force properties with the edit panel, analyzing joints and links with the analysis panel, and using the

animation bar to playback and scrub through the animation. These findings are summarized in Figure 97.



**Figure 97:** Results of the April 2023 study of five students showing that all students discovered and utilized the core functionalities of the new application without any prior experience or instruction.

During the open-ended discussion, several key questions were answered.

- 4 out of 5 participants believed they could learn the program without going through a tutorial or an in-class demo, indicating they felt capable of learning it on their own.
- All students successfully completed the task, creating the linkage, finding the position of the linkage at a specific time, and mapping that time back to the appropriate graph to obtain the necessary data.
- Without any explanation or instruction on how to use the app, students understood terms specific to PMKS+ including the difference between a tracer point and a joint.

#### 5.4.2 Analytics and Feedback

Firestore analytics, which is actively being tested on the development site, plays a crucial role in understanding user behavior by providing diagnostic data from every user session, including which tabs they open and how long they spend on the app. The analytics data supports the claim that users commonly switch to the analysis tab after using the application, with a conversion rate of about 40 percent, significantly higher than the anecdotal value of 10-20 percent with previous versions of the app. This data also provides insights into the average



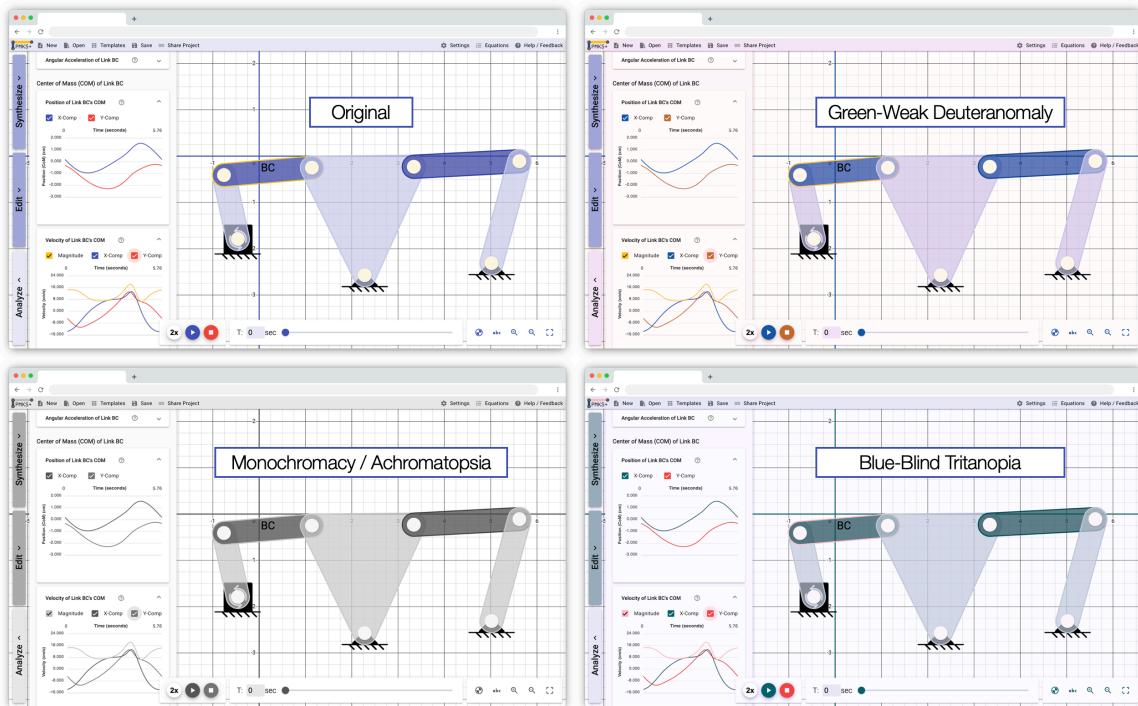
engagement time of users, which can help gauge the effectiveness of the new design. When the redesigned version is launched for use in WPI classes, the integration of analytics will be particularly beneficial in gathering data from a larger group of real-world users completing assignments using the app.

To further enhance user engagement and understanding, a direct feedback mechanism has been introduced, enabling users to submit support requests and feedback through the help tab. Users can optionally add their email to receive a reply and send diagnostic data gathered from their device. This approach allows the development team to stay connected with user needs and address any issues users may encounter, ultimately improving the overall user experience.

### **5.4.3 Accessibility Testing**

Ensuring accessibility is a vital aspect of user experience, a competitive advantage, and a core part of the organization's philosophy. To make the application usable for a diverse range of users, several accessibility tests are conducted.

Firstly, all mockups and UIs undergo testing through a colorblind filter, simulating the view of colorblind users to confirm that they can use the application effectively. Colors that appear similar to colorblind individuals are avoided, and contrast levels are carefully designed to ensure usability for students with low vision. An example of one these tests is shown in Figure 98.



**Figure 98:** A simulated view of PMKS+ through three colorblind filters: Green-Weak Deuteranomaly, Blue-Blind Tritanopia, and Monochromacy / Achromatopsia

Secondly, a screen reader is employed to test the application, verifying that it can be navigated entirely with the keyboard and that all labels and information can be read out loud. All non-visual elements are editable purely through keyboard input.

Finally, BrowserStack is used to test the application on hundreds of different real devices in the cloud, each with varying OSs and browser versions. This extensive testing ensures that potential browser-specific bugs are identified and resolved, delivering a consistent user experience across platforms.

## 6 Future Work and Development Plans

This section will discuss the future work and development plans for PMKS+, including upcoming features, awareness and advertising strategies, and a proposed timeline to ensure the timely implementation of improvements and new functionalities. These efforts will contribute to the continued growth and success of PMKS+ as a comprehensive and accessible tool for designing and analyzing mechanical linkages in engineering education.

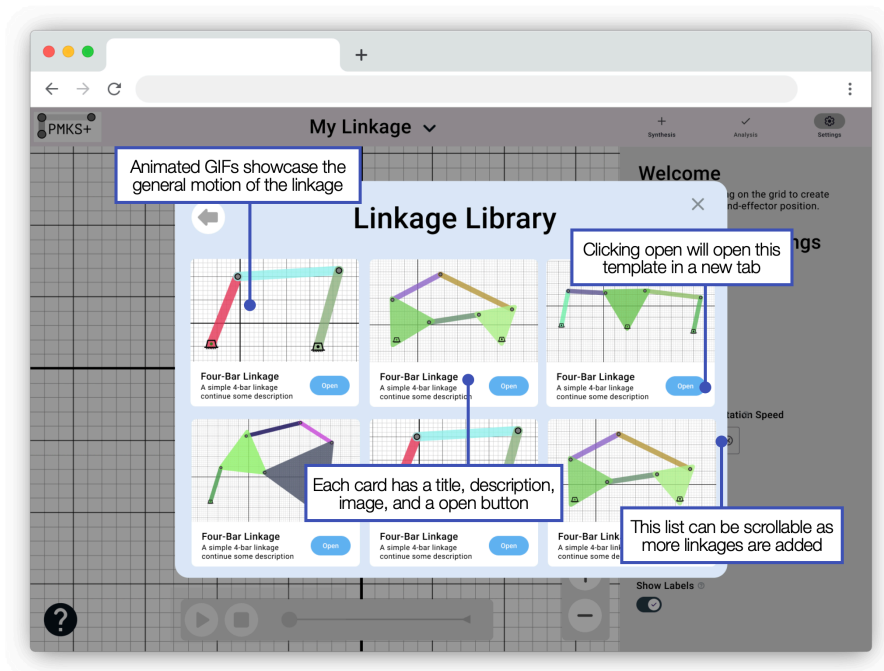
### 6.1 Upcoming Features

Based on the user studies conducted by our team and those from previous years, we have compiled a list of features that we would have liked to implement but did not have the time to. These are ranked in the order we suggest they be implemented, taking into account the amount of work each task requires and the overall benefit to the user in this prioritization.

**Update Save and Open:** Ensure saving and opening functionality supports all new features. Similar to the URL, saving currently does not encode all the information needed to restore the entire linkage state. It is essential to have a seamless save and open experience to prevent user frustration and loss of work. By updating this feature, users can be confident that their work will be preserved and easily accessible when needed.

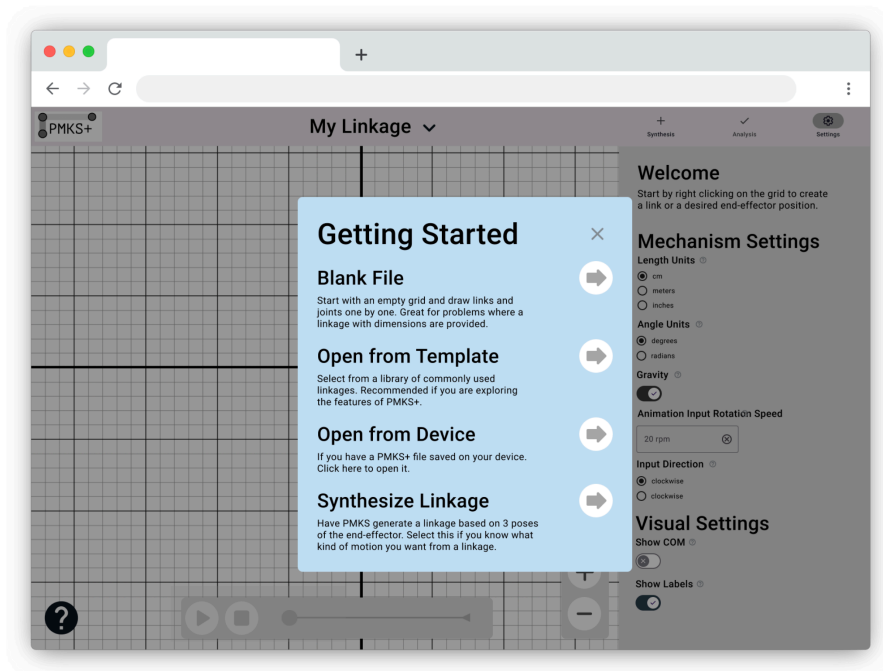
**Stress Analysis for Links:** Introduce fields to edit link minimum cross-sectional areas and stress analysis graphs. Solving for this is relatively simple since the forces between each joint are already being computed. An input field on the link edit panel needs to be added to allow users to input a cross-sectional area. Incorporating stress analysis will enable users to better understand the structural performance of their designs and make informed decisions about material selection and dimensions, leading to more robust and reliable mechanisms.

**Redesigned Templates Menu:** The current templates screen has a list of text buttons that show the available example linkages. Since some users might not be familiar with certain linkage types, such as a Stephenson Linkage, we propose a new version (shown in Figure 99) that incorporates an animated GIF for each linkage. This enhancement will make the templates menu more visually appealing and intuitive, allowing users to quickly identify the desired linkage type based on its appearance and motion characteristics. Additionally, the animated previews will help users better understand how different linkages function and inspire them to explore new design possibilities.



**Figure 99:** A redesigned linkage library popup that has an animated preview for each linkage

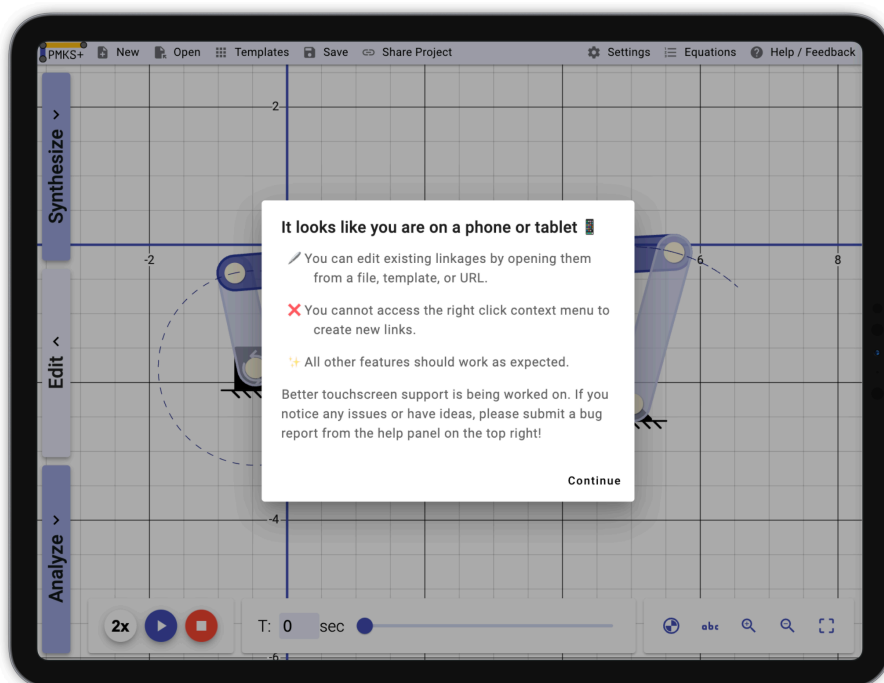
**New User Flow:** Starting with a blank screen can be intimidating for new users. To provide a more welcoming experience, PMKS+ should present four options (shown in Figure 100) for users when visiting a blank URL (i.e., the user is not opening another linkage using an encoded URL). By guiding users on how to get started, we can create a smoother onboarding process and encourage exploration.



**Figure 100:** A welcome popup to guide the user on how to get started

**Undo/Redo:** This feature has been highly requested by users. Implementing undo and redo functionality will allow users to experiment more freely and feel more confident in their actions. We believe the best approach is to store a list of user-performed actions in a linked list and traverse them as necessary.

**Full Touch Screen Support:** While most features currently work on large touchscreen devices, the context menu does not. As a temporary solution, a popup explaining these limitations has been implemented and is shown in Figure 101. However, full touchscreen support is vital for PMKS+ to be accessible to a wider range of devices, and we need to rethink all interaction methods to provide comprehensive support for tablet touchscreens.



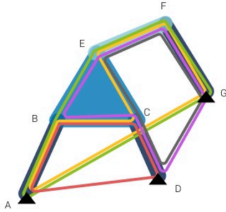
**Figure 101:** The current warning popup when accessing PMKS+ through a touchscreen interface such as a tablet

**Equations and Loops:** Complete the Equations tab based on the mockup. This feature is implemented in PMKS+ 2019 and was redesigned in the mockup (shown in Figure 102) but has not been implemented this year. Providing users with access to the loop equations and equations used to generate graphs will further enhance their understanding of the mechanisms and offer additional analytical capabilities.

### Mecanism Equations

These equations allow you to solve for the position, velocity, acceleration, and forces for the entire linkage simultaneously. You first need to select loop(s) that provide enough equations to solve for each unknown value.

**Selected Loop**



**Select Loop(s) to create equations from**

- Red Loop** ● A → B → C → D → A
- Green Loop** ● A → B → E → F → G → A
- Purple Loop** ● D → C → B → E → F → G → D
- Grey Loop** ● D → C → E → F → G → D
- Yellow Loop** ● A → B → C → E → F → G → A

**Position**

**Velocity**

**Solving for the Angular Velocity of each Joint at Timestep 30:**

$V_{Ax} + V_{CBx} + V_{DCx} + V_{AGx} = 0$  Find the loop equation by adding up velocities along the selected loop.

$\omega_{ABx} \times r_{AB} + \omega_{CBx} \times r_{BC} + \omega_{DCx} \times r_{CD} + \omega_{AGx} \times r_{GA} = 0$  Split Velocites into angular velocity  $\times$  link-length using the equation  $V = \omega \times r$

$20_{rpm} \times (2.8_{cm}, 1.8_{cm}) + \omega_{CBx} \times (5.1_{cm}, 0.2_{cm}) + \omega_{DCx} \times (3.4_{cm}, 2.6_{cm}) + 0_{rpm} \times (2.3_{cm}, 0.8_{cm}) = 0$  Plug in known values. ( $\omega_{A/D} = 0$ ,  $\omega_{B/A} = 20$  (set by user), and  $r$  (position vectors) solved in the [Position section](#))

**Eq1**  $X: 2.9 + 0.272\omega_{CBx} - 3.435\omega_{DCx} = 0$  Take the 3d cross product to find the X and Y components

**Eq2**  $Y: 1.14 + 4.566\omega_{CBx} - 0.942\omega_{DCx} = 0$

$\omega_{ABx} = 20_{rpm}$      $\omega_{CBx} = 8.92_{rpm}$     You now have 2 unknowns and 2 equations. Use the matrix method to solve for the unknowns

$\omega_{DCx} = 12.21_{rpm}$      $\omega_{AGx} = 0_{rpm}$

This is a single scrollable panel. It is split in two for formatting.

### Acceleration

#### Forces (Statics)

**Solving for the Forces of each Joint at Timestep 30:**

Generating Forces require solving multiple equations at the same time by using the [matrix method]. To see how these equations are found open each link to find it's equations.

**Link AB**

**Forces**

$\sum F = 0$  The sum of forces on Link AB is 0

$F_x + F_y + F_{grav} = 0$  There are 3 forces on this link

**Eq1**  $F_{Ax} + F_{Bx} = 0$  Plug in known values. ( $F_{grav} = -9.81$ ) Then split into X and Y components to generate 2 equations

**Eq2**  $F_{Ay} + F_{By} - 9.81 = 0$

**Moments**

$\sum M_A = 0$  The sum of moments on Link AB is 0

$r_{AB} \times F_B + r_{AB} \times F_{grav} + r_{AB} \times F_{grav} + T = 0$  Using [joint A as the moment reference](#), there are 3 forces acting on Joint A.

**Eq3**  $-0.029F_{Bx} + 0.011F_{By} + 5.592 + T = 0$  Plug in known values. ( $F_{grav} = -9.81 / r_{AB} = (2.8cm, 1.8cm)$ )

**Link BC**

**Link CD**

There are 8 unknowns and 9 equations in this system. Solving this set of equations using the matrix method results in the following values:

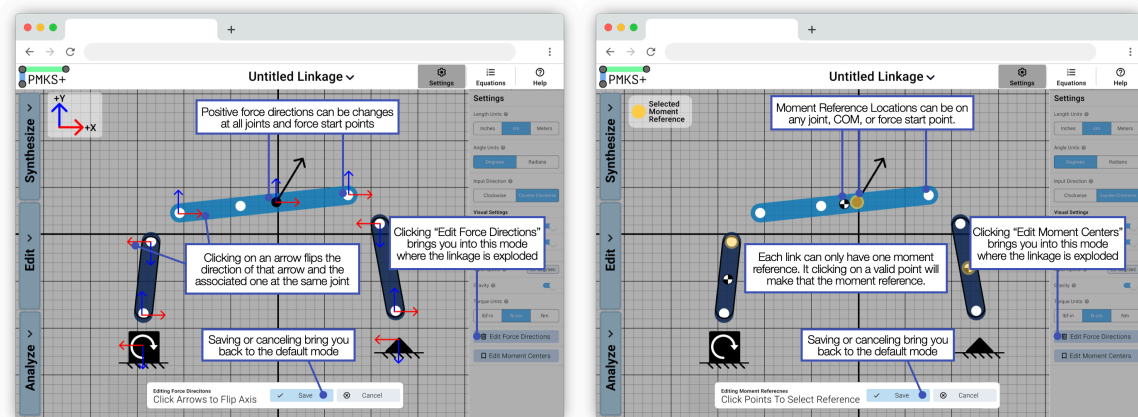
$F_{Ax} = 5.2_N$   $F_{Ay} = 0.2_N$   $F_{Bx} = 0.65_N$   $F_{By} = 1.23_N$   $F_{Cx} = 5.12_N$   $F_{Cy} = 1.42_N$   $F_{Dx} = 8.09_N$   $F_{Dy} = 0.54_N$

#### Forces (Dynamics)

**Figure 102:** A potential interface to show the loop equations and equations used to generate graphs

The first section displays a list of possible analysis loops for the linkage. Users must select two loops to have enough equations to solve simultaneously, allowing them to solve for all unknown variables. The following sections show the steps needed to solve for kinematic properties using the loop equation or forces using the matrix method.

To generate the equations of equilibrium, users need to define a positive x and y direction for each force, along with the moment reference to sum moments about. As this could vary across users depending on their choices, there should be a way to modify the equations PMKS+ generates to match these selections. The proposed solution (shown in Figure 103) involves incorporating two buttons in the settings panel, allowing users to enter a new mode for this purpose.



**Figure 103:** Mockups for editing force moment references and positive force directions

**Improve URL:** Encode all linkage data into the URL when generating a sharing link. Currently, information such as color and welded joints are not encoded. Implement a solution like Firestore and Cloud Functions that redirects shortlinks to the original URL, significantly shortening the URL users need to copy.

**Iterative Link Solving:** Allow users to lock specific lengths, angles, or joints, and enable subsequent changes to solve for new linkages that satisfy all constraints (similar to 2D CAD sketches). Striking a balance between fully recreating Solidworks' sketch tool and the current implementation is crucial. Users should be able to define link lengths first and drag them around to build a linkage.

**Instant Centers Analysis:** Many competitors do not support the instant center of rotation. PMKS+ should include the option to view and possibly edit instant centers for each link COM or joint.

**Additional Synthesis Methods:** MotionGen effectively allows path-based linkage synthesis. PMKS+ should implement more synthesis methods to generate other types of linkages or to follow a freehand path. Some easier options after 3-position synthesis could be 2-position synthesis or a method to generate a slider crank.

**3D CAD Export:** The long-term vision of PMKS+ is to integrate the tool into the workflow of creating any linkage. Users should be able to develop an idea in PMKS+ and seamlessly transition it into a CAD program for further modeling and design. Add the ability to export a 2D PMKS+ linkage to Solidworks, enabling a seamless transition from ideation and basic validation to detailed modeling and design.

By incorporating these features in future updates, PMKS+ can continue to evolve and provide users with a comprehensive, accessible, and efficient tool for designing and analyzing



mechanical linkages.

## **6.2 Awareness and Advertising**

Since PMKS+ is a free, open-source project maintained and updated by a volunteer group of students, it is crucial to work extra hard to keep the project alive and active, aligning incentives accordingly.

Currently, the user base is predominantly at WPI. While there has been success in reaching out to professors who teach linkage analysis courses and getting them to advise the project, further efforts are needed to expand the user base beyond WPI. This can be achieved by actively advertising the project online, at conferences, and at other educational institutions. Creating a better landing page showcasing all PMKS+ features and implementing improved SEO optimization will also help attract more users.

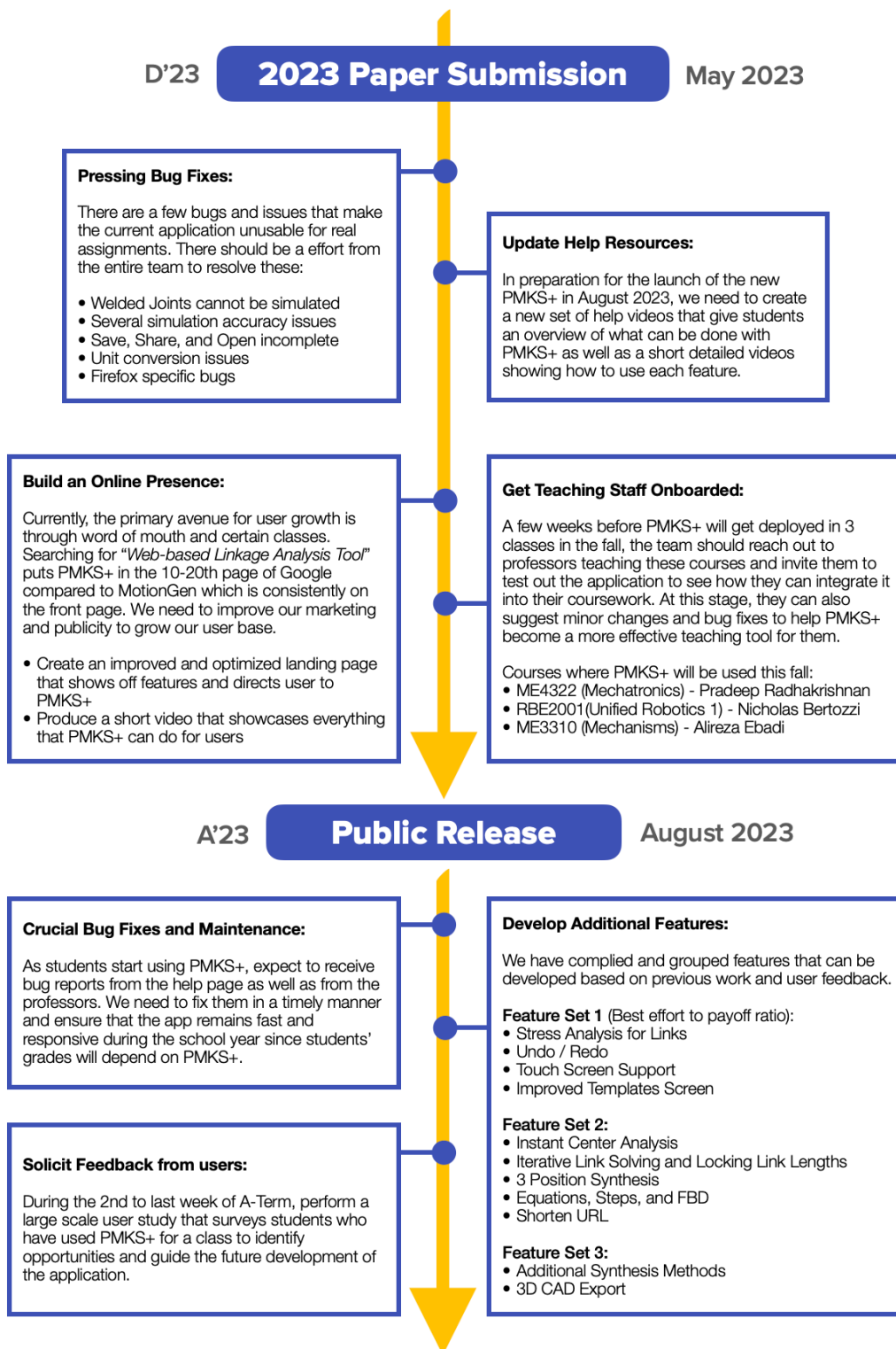
In addition to expanding the user base, it is essential to attract more student developers, especially those who can commit to the project for longer than one year. One challenge with projects like PMKS+ is the limited duration of student involvement, which can lead to knowledge gaps and delays in development. A possible solution could be to have a faculty member or full-time advisor heavily involved in the development, acting as a repository of knowledge.

To entice more students to join, promotional materials such as flyers or websites could be used to showcase the real-world experience they can gain from working on the project, as well as highlighting the number of monthly active users. Offering academic credit for participation may also increase interest. Another strategy to recruit students could involve in-app messaging, encouraging users to apply to work on PMKS+ as their project.

By simultaneously increasing the user base and recruiting long-term project members, a sustainable cycle can be created to ensure the continued growth and success of PMKS+.

## **6.3 Future Development Timeline**

In order for PMKS+ to reach its maximum potential and benefit as many students as possible when learning about linkage design and analysis, we present a suggested development timeline for the next few months in Figure 104. This timeline will help ensure that all essential features and improvements are implemented in a timely and efficient manner, allowing for a public release in August 2023.



**Figure 104:** A suggested plan for the next few months of development for PMKS+. The next key milestone is the public release in August 2023.

This timeline ensures that the development of PMKS+ progresses smoothly, allowing the team to prioritize features and improvements effectively. By following this schedule, PMKS+ will continue to evolve as a comprehensive and accessible tool for designing and analyzing mechanical linkages.

## 7 Conclusion

The development of PMKS+ has been an ongoing collaborative effort, with each team contributing to the project and building upon the work of their predecessors. This year, our focus on reimagining the user experience to ensure that users can discover advanced analysis features is shaping PMKS+ into an invaluable tool for mechanical and robotics engineering students and educators at WPI and beyond.

User studies and demographic research have provided essential insights into the needs and preferences of our target audience. By actively engaging with our users through heuristic evaluations, focus group discussions, and validation testing, we have been able to identify and prioritize features that have the most significant impact on our users. One of the primary successes of our development process is the increased organic discovery of analysis features. User testing has shown that even without prior experience or tutorials, users can discover and utilize essential features of the application. This outcome is a testament to the effectiveness of our redesigned user interface and its ability to promote user engagement and learning.

Informed by user studies, we created high-fidelity mockups, which have allowed us to rapidly visualize and iterate on the UI and user experience. These mockups have guided the design process and enabled us to address key usability concerns. Accessibility has been a cornerstone of our development philosophy, with our commitment being further reinforced by our user studies and mockup iterations. By testing our application through colorblind filters, screen readers, and various devices and browsers, we have ensured that PMKS+ remains an accessible and intuitive platform for users with diverse abilities and backgrounds, while keeping it a free and open-source program for anyone to utilize and learn from.

Our future development plans include a range of exciting new features and improvements. By adding more analysis and synthesis options, quality of life features like Undo and Redo, and integrations with 3D CAD software, we aim to make PMKS+ an even more versatile and powerful tool for mechanical linkage design and analysis. Furthermore, our suggested development timeline ensures that these updates are implemented in a timely and organized manner, allowing for a public release in multiple WPI classes in the fall of 2023.

The potential of PMKS+ to revolutionize the education of mechanical linkage design and analysis is immense. By continuing to actively solicit and incorporate user feedback, we will ensure that our application evolves in response to the ever-changing needs of students and educators. We are immensely proud of the progress we have made thus far, and we are excited to witness PMKS+ becoming a well-known, integral tool in engineering education.

## 8 References

- [1] "PMKS Introduction," Accessed: 2023-05-03, <https://design.engr.oregonstate.edu/pmksintro.html>
- [2] "DesignEngrLab/PMKS: Planar Mechanism Kinematic Simulator," Accessed: 2023-05-03, <https://github.com/DesignEngrLab/PMKS>
- [3] Appikarla, P., Cecil, G. D., Taylor, M. T., and Tsiakmakis, D., 2019, "PMKS+: Recreating a Legacy Application," Worcester Polytechnic Institute.
- [4] Dowd, T. D., Zhang, H., and Dutilleul, R. A., 2020, "PMKS+: Enhancements," Worcester Polytechnic Institute.
- [5] Galvan, A., 2022, "PMKS+: A Tool to Analyze and Synthesize Planar Mechanisms," Worcester Polytechnic Institute.
- [6] Andrews, J., Knox, B., Leach, B., and Rivera, G., 2018, "2018 MQP: Automated Design of Planar Linkages," .
- [7] "Norton Associates Engineering - Linkage Design," Accessed: 2023-05-03, <http://designofmachinery.com/Linkage/>
- [8] "MotionGen Pro," Accessed: 2023-05-03, <https://motiongen.io/>
- [9] "SAM - The Ultimate Mechanism Designer," Accessed: 2023-05-03, <https://www.artas.nl/en/>
- [10] "SOLIDWORKS Community," Accessed: 2023-05-03, <https://my.solidworks.com/>
- [11] "Working Model 2D," Accessed: 2023-05-03, <https://www.design-simulation.com/wm2d/>
- [12] Google, 2022, "Get Started with Material," <https://m3.material.io/get-started>
- [13] Heonia, A., 2021, "AndriiHeonia/hull: A JavaScript library that implements various convex hull algorithms," <https://github.com/AndriiHeonia/hull>.
- [14] Bumbu, A., 2021, "bumbu/svg-pan-zoom: A JavaScript library that enables panning and zooming of an SVG in an HTML document, with mouse events or custom JavaScript hooks," <https://github.com/bumbu/svg-pan-zoom>.

## 9 Appendix

- 2023 Github Repo: <https://github.com/PMKS-Web/PMKSWeb/tree/UI-Redesign>
- 2022 Github Repo: <https://github.com/PMKS-Web/PMKSWeb>
- Final Presentation: <https://tinyurl.com/nusz4tp9>
- Weekly Update Presentation: <https://tinyurl.com/5n7w2b74>
- 2022 A Term Survey: <https://forms.gle/G1u4CwL83MtyMeU48>
- 2022 A Term Survey Results: <https://tinyurl.com/yeynn7nt>
- 2022 B Term Instructions: <https://tinyurl.com/bdrtnxvw>
- 2022 B Term Survey: <https://forms.gle/1qfVnQGSWm7QuCa16>
- 2022 B Term Results: <https://tinyurl.com/4unb9ruc>
- 2023 D Term Survey Instructions: <https://tinyurl.com/bdfndckk>
- 2023 D Term Survey Results: <https://tinyurl.com/2z8xmfuc>
- 2023 D Term Focus Group Notes: <https://tinyurl.com/yv5zcs7n>

## 10 Vita

Kohmei Kadoya is a robotics engineer and user experience designer who completed their M.S. and B.S. in Robotics Engineering with a minor in Computer Science at Worcester Polytechnic Institute (WPI) from September 2019 to May 2023. Two key courses that contributed to their expertise in the development of the Planar Mechanism Kinematic Simulator (PMKS+) are MIS 585 - User Experience Design and CS 3733 - Software Engineering. In the User Experience Design course, Kadoya gained hands-on experience in creating innovative user experiences, focusing on UX/UI, visual, service, experience, and product design. The Software Engineering course provided Kadoya with a solid foundation in modern software development techniques, emphasizing topics such as requirements analysis, design, architecture, implementation, testing, and project management. Kadoya's proficiency extends to various software programming languages, design tools, and fluency in both English and Japanese.

Contact: [kkadoya@wpi.edu](mailto:kkadoya@wpi.edu)

Website: <https://www.kohmeik.com>